

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет
імені Дмитра Моторного
Факультет енергетики і комп'ютерних технологій

ЗАТВЕРДЖУЮ

Зав. каф. "Комп'ютерні науки"

доц. _____ **Сергій ШАРОВ**

«16» червня 2025 р.

Пояснювальна записка

до кваліфікаційної роботи здобувача СВО Бакалавр
(ступінь вищої освіти)

на тему: «Інформаційна система для продажу органічної їжі на основі веб-технологій з використанням Vue.js»

52/4КНД.9685254.000000ПЗ

Виконав: здобувач вищої освіти 4 курсу, групи 41КН
спеціальності 122 Комп'ютерні науки
за ОПП Комп'ютерні науки
(шифр і назва спеціальності та ОПП)

Артем ВАСКОВСЬКИЙ
(підпис)

Керівник ас. _____ **Ганна ГЕШЕВА**
(підпис)

Консультант доц. _____ **Сергій ШАРОВ**
(підпис)

Консультант доц. _____ **Лариса БОЛТЯНСЬКА**
(підпис)

Консультант доц. _____ **Михайло ЗОРЯ**
(підпис)

Нормконтроль, ст.викл. _____ **Ольга ЗІНОВ'ЄВА**
(підпис)

Рецензент, доц. _____ **Юрій СПИЛІЦІН**
(підпис)

Запоріжжя 2025 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет імені Дмитра
Моторного

Факультет: Енергетики і комп'ютерних технологій
Кафедра: Комп'ютерні науки
Ступінь вищої освіти: Магістр
Спеціальність: 122 Комп'ютерні науки
ОПП: Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри_КН
к.пед.н., доц. _____ Сергій ШАРОВ
“10” _жовтня_ 2024__ року

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

Васковський Артем Сергійович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи__ «Інформаційна система для продажу органічної їжі на основі веб-технологій з використанням Vue.js»
керівник проєкту Гешева Ганна Валеріївна, асистент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом університету від “30” вересня_2024_року №_452-С

2. Строк подання здобувачем вищої освіти роботи _12 червня 2025_

3. Вихідні дані до роботи: дані обстеження об'єкту, статистичні дані, технічне завдання на дипломне проєктування, матеріали виробничих практик, нормативні документи, науково-технічна література, електронні ресурси та ін.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)_____

1. Аналіз предметної області та існуючих рішень.
2. Специфікація вимог до інформаційної системи.
3. Опис прийнятих проєктних та технологічних рішень.
4. Дослідна експлуатація інформаційної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників)

Діаграма використання, діаграми IDEF0, розкадровка інтерфейсу, структурна схема, візуальні компоненти інтерфейсу, алгоритм роботи, інтерфейс програми

Презентація – 11 слайдів

1. Титульний слайд
2. Предмет, об'єкт, мета

- 3. Завдання дослідження
- 4. Порівняльна характеристика аналогів програмного продукту
- 5. Діаграма варіантів використання
- 6. Інструментальні засоби
- 7. Інтерфейс і робота веб-застосунку системи
- 8. Інтерфейс і робота веб-застосунку системи
- 9. Тестування
- 10. Висновок

6. Консультанти розділів кваліфікаційної роботи

Розділ/ Підрозділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4.3	Болтянська Л. О.	15.10.2025	15.06.2025
4.4	Зоря М. В.	15.10.2025	15.06.2025

7. Дата видачі завдання 10 жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1.	Аналіз предметної області	12.11.2024	виконано
2.	Специфікація вимог до веб-застосунку	07.12.2024	виконано
3.	Проектування веб-застосунку	24.01.2025	виконано
4.	Обґрунтування інструментальних засобів	25.02.2025	виконано
5.	Розробка веб-застосунку для продажу органічно їжі	04.04.2025	виконано
6.	Тестування та налагодження веб-застосунку	05.06.2025	виконано
7.	Підпис керівником роботи	14.06.2025	виконано
8.	Підпис завідувачем кафедри	16.06.2025	виконано

Здобувач вищої освіти

Керівник
кваліфікаційної
роботи

_____ Артем ВАСКОВСЬКИЙ
(підпис) (власне ім'я та прізвище)

_____ Ганна ГЕШЕВА
(підпис) (власне ім'я та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 91 с., 33 рис., 10 табл., 25 посилань, 6 додатків.

Актуальність: Зростання попиту на продукти здорового харчування та розвиток електронної комерції обумовлюють потребу у створенні спеціалізованого сервісу для ознайомлення та продажу подібної продукції. Створення інтернет-магазину дозволить задовольнити попит цільової аудиторії та надасть легший доступ до органічної продукції. Використання сучасних фреймворків для фронтенду та бекенду є ключовим аспектом для швидкої розробки інтерактивного та легко масштабованого користувацького інтерфейсу.

Об'єкт дослідження: електронна комерція органічних продуктів харчування.

Предмет дослідження: інтернет-магазини органічної продукції, його архітектура та можливості.

Мета роботи: створити інтернет-магазин з продажу органічної продукції використовуючи фреймворки Vue.js та Django.

Задачі дослідження:

- Проаналізувати предметну область та існуючі рішення на ринку органічної їжі.
- Обґрунтувати вибір фреймворку Vue.js як основного інструменту розробки клієнтської частини додатку.
- Сформулювати функціональні та нефункціональні вимоги до інтернет-магазину
- Спроектувати архітектуру та інтерфейс додатку.
- Реалізувати клієнтську частину на Vue.js та серверну частину на Django.
- Провести тестування, підготувати інструкцію користувача та виконати техніко-економічний розрахунок.

Практичне значення дослідження: реалізований прототип може бути впроваджений у малих та середніх підприємствах аграрного сектору для автоматизації продажів.

Ключові слова: ІНТЕРНЕТ-МАГАЗИН, ОРГАНІЧНА ЇЖА, E-COMMERCE, VUE.JS, VUEX, DJANGO.

SUMMARY

Bachelor's thesis: 91 pages, 33 figures, 10 tables, 25 references, 6 appendices.

Relevance: Growing demand for healthy food products and the development of e-commerce necessitate the creation of a specialized service for promoting and selling such products. Creating an online store will satisfy the target audience's demand and provide easier access to organic products. The use of modern frameworks for front-end and back-end is a key aspect of the rapid development of an interactive and easily scalable user interface.

The object of research: e-commerce of organic food products.

The subject of research: online stores of organic products, their architecture, and their capabilities.

The purpose of the work: to create an online store for selling organic products using the Vue.js and Django frameworks.

Research objectives:

- Analyze the subject area and existing solutions in the organic food market.
- Justify the choice of the Vue.js framework as the primary tool for developing the client part of the application.
- Formulate functional and non-functional requirements for the online store.
- Design the architecture and interface of the application.
- Implement the client part on Vue.js and the server part on Django.
- Conduct testing, prepare user instructions, and perform technical and economic calculations.

The practical significance of the research: The implemented prototype can be implemented in small and medium-sized enterprises in the agricultural sector to automate sales.

Keywords: ONLINE STORE, ORGANIC FOOD, E-COMMERCE, VUE.JS, VUEX, DJANGO.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ	8
1.1 Узагальнена характеристика предметної області	10
1.2 Огляд і аналіз існуючих аналогів системи	14
1.3 Розробка технічного завдання	25
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ ...	34
2.1 Глосарій	34
2.2 Специфікація функціональних та нефункціональних вимог	36
2.3 Концептуальна модель використання інформаційної системи	41
2.4 Розробка функціональної моделі	50
РОЗДІЛ 3 ОПИС ПРИЙНЯТИХ ПРОЄКТНИХ ТА ТЕХНОЛОГІЧНИХ РІШЕНЬ	52
3.1 Розробка об'єктної моделі	52
3.2 Проєктування інтерфейсу програмної системи	56
3.3 Обґрунтування вибору мови програмування	61
3.4 Опис програмної реалізації	66
РОЗДІЛ 4 ДОСЛІДНА ЕКСПЛУАТАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ...	73
4.1 Інструкція користувача ІС	73
4.2 Основи тестування програмного забезпечення	75
4.3 Техніко-економічні показники ІС	77
4.4 Охорона праці	84
ВИСНОВКИ	90
СПИСОК ЛІТЕРАТУРИ	92
Додаток А	95
Додаток Б	97
Додаток В	99
Додаток Г	101
Додаток Д	105
Додаток Е	107

ВСТУП

Актуальність. Інформаційно-комунікаційні технології стрімко розвиваються у сучасному світі, проникаючи в усі сфери людської діяльності та змінюючи підходи до ведення бізнесу, торгівлі та повсякденного життя. Одним з найважливіших напрямків застосування ІКТ є електронна комерція. Вона надає підприємствам нові можливості для збільшення продажів, а споживачам – зручний доступ до широкого асортименту товарів та послуг. Особливої актуальності набуває розробка спеціалізованих інтернет-магазинів, орієнтованих на конкретні ніші ринку. Це також стосується ринку здорової органічної їжі, яка в Україні поступово розвивається з кожним роком.

Розробка функціонального та зрозумілого у користуванні інтернет-магазину дозволить виробникам та продавцям ефективно презентувати свої товари, надавати детальну інформацію про їх походження, склад та сертифікацію. Використання сучасного фреймворку для розробки динамічних користувацьких інтерфейсів Vue.js, у поєднанні з серверним фреймворком Django дозволить створити якісний та надійний веб-додаток, який відповідає вимогам ринку.

У зв'язку з необхідністю автоматизації процесу продажу органічної їжі та інформування споживачів про її корисність, тему дослідження було сформульовано як «Інформаційна система для продажу органічної їжі на основі веб-технологій з використанням Vue.js».

Об'єкт: веб-додаток для продажу органічної їжі.

Предмет: інформаційна система для продажу органічної їжі

Мета: створити інформаційну систему для продажу органічної їжі з використанням сучасних веб-технологій.

Сформована мета, об'єкт і предмет дослідження дозволили поставити такі завдання:

1. Провести аналіз предметної області та існуючих рішень у сфері електронної комерції органічної їжі.
2. Розглянути та обрати інструменти для розробки продукту.

3. Створити технічне завдання на розробку веб-додатку.

4. На основі складеного технічного завдання розробити інформаційну систему, скласти інструкцію користувача та провести тестування.

Для реалізації поставлених задач було використано наступні методи дослідження: теоретичні: аналіз наукової та спеціальної літератури з електронної комерції, органічного виробництва, програмування, практичні: програмна реалізація веб-застосунку.

Практичне значення: запропонований прототип може бути впроваджений у підприємствах аграрного сектору для автоматизації процесу прийняття замовлень.

Апробація результатів була здійснена шляхом публікації статті «Розробка веб-системи для продажу органічної їжі з використанням Vue.js» у збірнику матеріалів Всеукраїнської науково-технічної конференції здобувачів вищої освіти за підсумками наукових досліджень 2024 року (м. Запоріжжя, 2025).

Структура і обсяг роботи. Дипломна робота складається з вступу, чотирьох розділів, загальних висновків, списку використаних джерел (25 найменувань). Перший розділ присвячений аналізу предметної області та аналогів розроблюваної інформаційної системи. У другому розділі сформовано специфікацію вимог і побудовано концептуальні та функціональні моделі. Третій розділ присвячений загальному огляду технологічного стеку, обґрунтування вибору мови програмування, розробці об'єктної моделі та опису програмної реалізації. У четвертому розділі наведено інструкцію користувача для розробленої інформаційної системи, результати тестування та техніко-економічні показники. Загальний обсяг роботи складає 79 сторінок.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

1.1 Узагальнена характеристика предметної області

У сучасному світі все більше людей використовують свій мобільний телефон для придбання продуктів харчування та інших речей побуту онлайн. Це дозволяє швидко та зручно придбати усе потрібне без необхідності відвідувати фізичний магазин й витратити декілька годин на пошук потрібних товарів.

Як показує дослідження Forbes, у 2024 році приблизно 20% усіх роздрібних покупок здійснювалося в інтернеті [25]. І як показує тенденція, цей відсоток буде зростати. Глобальний оборот e-commerce зараз становить 6 трильйонів доларів, а до 2027 року має досягнути позначки у 8 трильйонів доларів [18]. Кількість інтернет-покупців сягнула 2,71 млрд (близько 33% населення світу) і продовжує зростати щороку [18]. Це підтверджує що створення інтернет-магазину є перспективним напрямом для розширення цільової аудиторії бізнесу та збільшення продажів.

Електронна комерція – це купівля або продаж товарів і послуг через Інтернет, без необхідності особистої зустрічі обох сторін. Усі дії відбуваються онлайн, і в результаті право власності або право користування переходить від однієї особи до іншої [17].

Зараз майже у кожної людини є телефон з доступом в інтернет, що дозволяє замовляти товари та їжу з доставкою додому за декілька кліків. Наприклад, у США ~73% споживачів роблять покупки за допомогою смартфона, а в Китаї цей показник досягає 92% [18]. Проте не тільки доступність смартфонів впливає на стрімкий розвиток електронної комерції, цьому також сприяє розвиток ІКТ, зменшення вартості послуг та підвищення доступу до мережі Інтернет, збільшення темпів життя покупців, що впливає на

пошук зручніших способів замовлення, тощо [10]. Це говорить про те, що фермерам і підприємствам вкрай важливо користуватися сучасними можливостями і реалізовувати свою продукцію за допомогою онлайн торгівлі.

Іншими перевагами онлайн торгівлі є можливість інтегрувати персональні пропозиції для користувачів, використовувати штучний інтелект або спеціальні алгоритми для підбору цікавих рекомендацій. Український ринок e-commerce також динамічно розвивається, завдяки технологічним інноваціям, адаптації до змін у споживчих звичках та інтеграції з міжнародними ринками, навіть попри виклики останніх років [16]. У 2024 році обсяг онлайн-покупок в Україні зріс на 25% порівняно з попереднім роком, при цьому українці витратили онлайн приблизно 239 млрд грн [1].

Найпопулярнішими категоріями товарів які купляють онлайн є електроніка, одяг і косметика. Але зростає і сегмент продуктів харчування та товарів для здорового способу життя. Війна та пандемія стимулювали перехід бізнесів в онлайн, розширили доставку товарів додому і підвищили довіру споживачів до покупок в інтернеті [16]. В цілому, експерти відзначають, що екологічність дедалі більше впливає на поведінку користувачів інтернету під час вибору товарів. Сучасні споживачі дедалі більше звертають увагу на екологічні ініціативи компаній та приділяють більшу увагу екологічним товарам [5].

Популярність органічної їжі зростає не тільки в Україні, але й в усьому світі. Це пов'язано з бажанням людей більше піклуватися про своє здоров'я та здоров'я своїх рідних, зростанням екологічної свідомості, а також із загальним трендом переходу до сталого споживання. Багато споживачів починають уважніше вивчати склад та походження продуктів на відсутність хімікатів даючи перевагу здоровим, органічним продуктам. Як показують дослідження, жителі розвинених країн готові переплачувати за здорову їжу, а уряди деяких країн стимулюють виробників до виробництва натуральної продукції [4].

Споживачі віддають перевагу органічним продуктам не даремно. В Україні виробництво такої продукції регулюється законодавством. Це

зобов'язує підприємства виробляти продукти рослинного та тваринного походження відповідно до чітких стандартів, щоб вони відповідали вимогам якості. Продукція має вироблятись, сертифікуватись, зберігатись та реалізовуватись за правилами органічного виробництва [7]. Органічна їжа повинна бути вирощена без використання синтетичних хімікатів, таких як штучні пестициди, добрива, регулятори росту рослин, кормові добавки, а також не повинні містити генетично модифікованих організмів [6].

Крім того, важливою ознакою органічної продукції є її сертифікація, яка підтверджує відповідність встановленим стандартам. Станом на 2022 рік, загальна кількість операторів органічного ринку в Україні становила 462, серед яких 380 є безпосередньо сільськогосподарськими виробниками [12]. Їхня діяльність сертифікована провідними іноземними органами відповідно до стандартів ЄС (Регламент Ради ЄС) № 834/2007) та США (NOP) [23].

Станом на 2021 рік в Україні спостерігалася тенденція зростання попиту на органічну продукцію у порівнянні з 2019 та 2020 роками. Проте частка органічної продукції, проданої на внутрішньому ринку України складала лише 3,6% у порівнянні з обсягами експорту. Це говорить про те, що українські споживачі вживають набагато менше органічної продукції, ніж мешканці країн Європейського Союзу. Але з іншої сторони, український внутрішній споживчий ринок органічних продуктів все ж таки розвивається завдяки мережам супермаркетів, які поступово розширюють асортимент такої продукції, тим самим популяризуючи цю продукцію серед населення [23]. Як показують дослідження, рівень поінформованості українців щодо органічної їжі доволі малий [13]. Тож у процесі розробки веб-сайту потрібно до кожного товару додати ознайомчу інформацію про виробника органічної продукції, її склад, сертифікацію та умови зберігання. Додавання динамічного контенту на сайт також допоможе збільшити конвертацію в два рази [14]. Таким контентом можуть бути цікаві інформативні блоги, що розповідають про органічну їжу і чим вона корисна для здоров'я, корисні відео з оглядом продукції, процесом її виготовлення, тощо. Це допоможе ознайомити

споживачів з усією необхідною інформацією та позитивно вплине на довіру до бренду та посприє просуванню сайту у пошуковій видачі.

Інтернет-магазин також буде оптимальним рішенням для малих підприємств, так як маючи невеликий обсяг виробництва, таким бізнесам дуже важко становити конкуренцію на ринку споживачів. Часто, для того щоб не допустити вироблену продукцію на відходи, дрібні підприємства продають її за нижчою ціною [13]. Онлайн магазин здатен вирішити цю проблему, дозволяючи не лише розширити потенційну клієнтуру, а й встановити кращий зв'язок між виробником та кінцевим клієнтом. Такий веб-сайт стане повноцінним інструментом, який допоможе компанії заявити про себе в інтернеті, презентувати свою продукцію, розвивати бренд та вирізнятися серед конкурентів. До того ж, за допомогою сайту можна аналізувати дії користувачів, робити висновки щодо популярності тих чи інших товарів, отримувати зворотній відгук та планувати виробництво згідно з отриманою інформацією. Проте створення інтернет-магазину - це лише початковий етап. Для досягнення стабільного рівня продажів необхідно постійно працювати над SEO-просуванням, вдосконалювати структуру сайту, покращувати контент і юзабіліті. Історія створення інтернет-магазинів свідчить, що маючи якісний веб-ресурс та використовуючи методи та технології інтернет-маркетингу, можна збільшити обсяги реалізації продукції у кілька разів [9]. Крім того, налагоджений інтернет-магазин дозволить підприємству оптимізувати канали розподілу продукції, знизити витрати на реалізацію та підвищити ефективність продажів, що збільшить прибутковість бізнесу [15].

У висновку можна сказати, що створення інтернет-магазину є ефективним рішенням для будь-якого підприємства з продажу органічної їжі. Власний веб-сайт дозволить розширити ринок збуту і підвищити прибутковість, встановити прямий зв'язок із клієнтами, підвищити довіру до бренду та популяризувати здорову, екологічну їжу серед населення.

1.2 Огляд і аналіз існуючих аналогів системи

Дослідження існуючих на ринку аналогів з продажу органічної їжі є ключовим етапом у процесі розробки власного проєкту. Такий аналіз дозволить глибше зрозуміти сучасні тенденції, допоможе визначити сильні та слабкі сторони конкурентів і допоможе уникнути розробки неефективних рішень. Збір та аналіз цієї інформації допоможе зробити платформу більш привабливою для користувачів та більш конкурентоспроможною на ринку.

Метою дослідження є не лише виявлення найкращих практик, але й формування чітких технічних та функціональних вимог до майбутнього інтернет-магазину. Отримані результати під час аналізу стануть фундаментом для прийняття рішення щодо розробки архітектури системи, структури бази даних та методів реалізації окремих компонентів системи.

Для аналізу були використані наступні веб-сайти: «EatOrganic» - ринковий майданчик, який дає можливість виробникам представити свої товари клієнтам. «Смак життя» - інтернет-магазин зі Львова з широким асортиментом органічних, безглютенових і веганських продуктів для здорового харчування. Платформа «Ahimsa», яка спеціалізується на продажі екологічно чистих та натуральних продуктів. Веб-сайт «GreenFood», який орієнтується на продажі органічної їжі, включаючи суперфуди, рослинні напої та продукти без цукру.

Дослідимо веб-сайт «Смак життя» детальніше. Асортимент цього магазину налічує понад 800 товарів з різних категорій, серед яких напої, натуральні солодоші, продукти для імунітету, горіхи та снеки, веган альтернативи, олії тощо. На сайті також наявний зручний швидкий пошук по усьому асортименту, який допомагає знайти необхідні товари за ключовими словами. На головній сторінці увага покупців відразу звертається на товари з маркетингових категорій, зокрема ціна тижня, новинки, хіти продажів, акційні товари тощо. У шапці сайту також знаходяться промо-сторінки, такі як «Ціна тижня» та сезонні знижки, зокрема «Великодня колекція», які заохочують

споживача витратити гроші на товари зі зниженою ціною. Такий комбінований підхід може допомогти підвищити зацікавленість покупців та збільшити продажі. На рисунку 1.2 зображена головна сторінка інтернет-магазину «Смак життя».

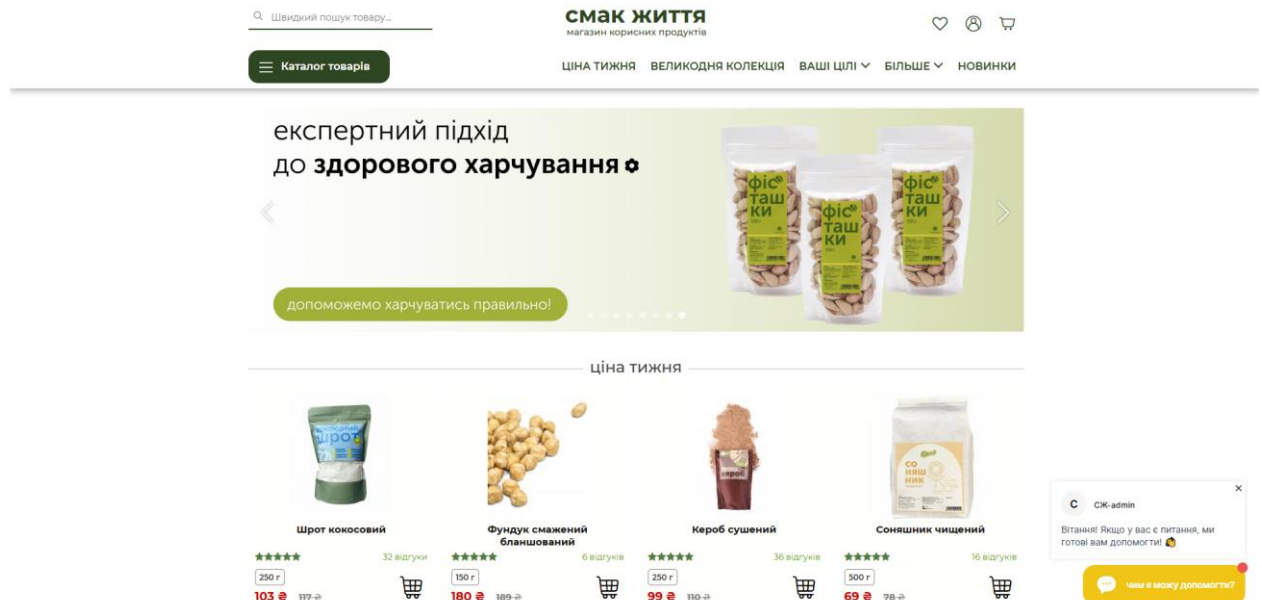


Рисунок 1.2 – Головна сторінка сайту «Смак життя»

Сайт має зручний каталог з великою кількістю категорій та підкатегорій товарів. Пошук потрібного продукту не займає багато часу. Крім того, на сторінці з товарами є можливість відфільтрувати продукти по типу харчування, за наявністю корисних компонентів а також за відсутністю певних інгредієнтів. На кожній позиції товарів є дуже детальний опис і особливості саме цього продукту, умови зберігання, поради у використанні, сертифікація харчова цінність та рекомендований термін придатності. Ця прозорість допомагає ознайомити споживача та викликати довіру до продукту та платформи.

Сайт має простий, зрозумілий та дружній дизайн, дозволяє додавати товари у кошик без необхідності реєструватися. Для оформлення замовлення достатньо вказати телефон, email, прізвище та спосіб доставки.

У особистому кабінеті можна продивитися свою історію замовлень та змінити особисті дані, включаючи адрес доставки. На рисунку 1.2.1 зображено особистий кабінет інтернет-магазину «Смак життя».

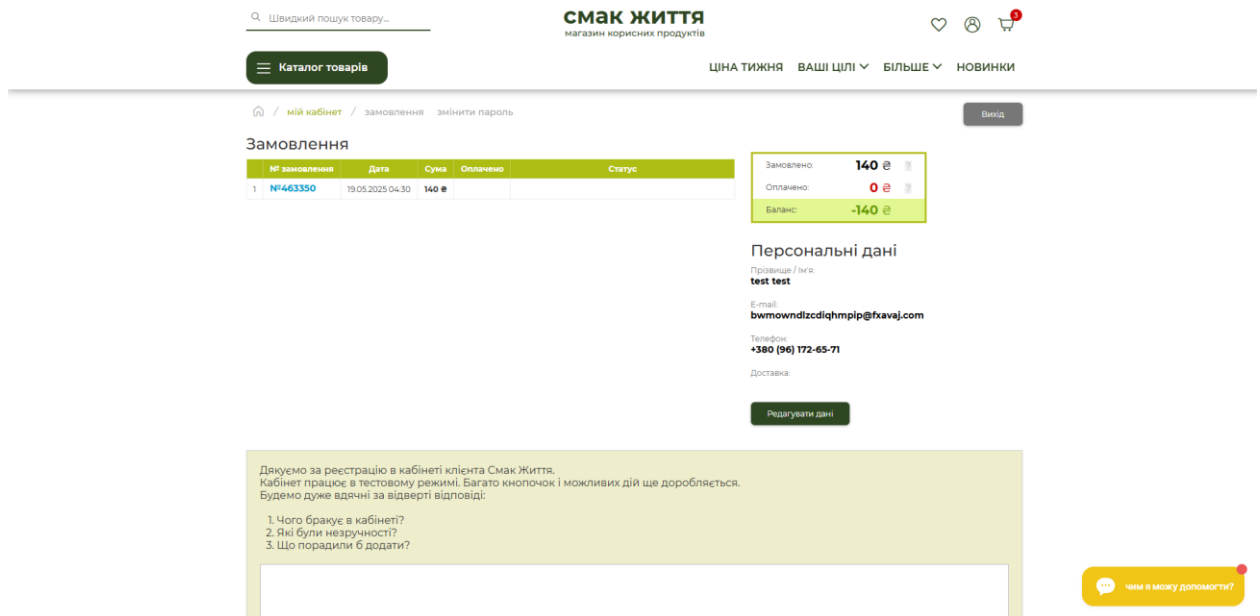


Рисунок 1.2.1 – Особистий кабінет

Також важливим функціоналом на сайті є те, що споживачі можуть продивитися і залишити відгук та рейтинг під кожним товаром, звернутися до підтримки у онлайн чаті, дізнатися цікаву інформацію про історію бренду та ознайомитися з умовами доставки та оплати.

Наступним сайтом можна розглянути українську платформу EatOrganic. Це не типовий інтернет-магазин, у ньому не можна додати товари до кошику чи здійснити оплату. Натомість сайт пропонує ознайомитися з асортиментом продукції, яка є в наявності безпосередньо у виробника. Таким чином, він поєднує фермерів і споживачів, даючи пряму контактну інформацію з виробниками. Сайт позиціонує себе як «Каталог готових українських харчових продуктів. Ринковий майданчик, який дає можливість виробникам органічної продукції представити свої товари клієнтам». На рисунку 1.2.2 зображена головна сторінка «EatOrganic».

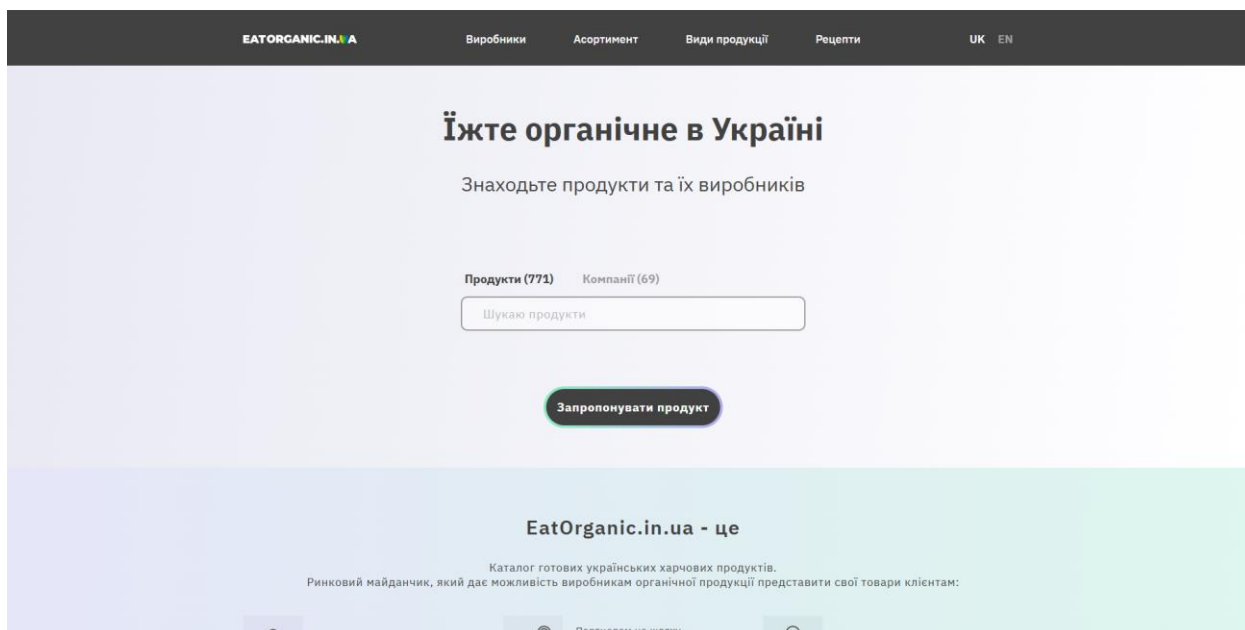


Рисунок 1.2.2 – Головна сторінка «EatOrganic»

Дизайн сайту виглядає чистим, сучасним, не перевантаженим. Пошук товарів здійснюється через зручний пошук по ключовим словам або через сторінки «Асортимент» та «Види продукції». На кожній сторінці товару є посилання на інформацію про виробника, торгова марка та сертифікація. Проте, на відміну від попередньо розглянутого сайту, на багатьох товарах вказаний дуже стислий опис, якого може здатися мало для потенційного споживача.

Асортимент магазину налічує понад 700 позицій товарів від 69 сертифікованих фермерів України. На кожному товарі можна детальніше дізнатися про виробника, розташування підприємства, контактну інформацію, та сертифікат якості. На рисунку 1.2.3 зображена сторінка з наявним асортиментом.

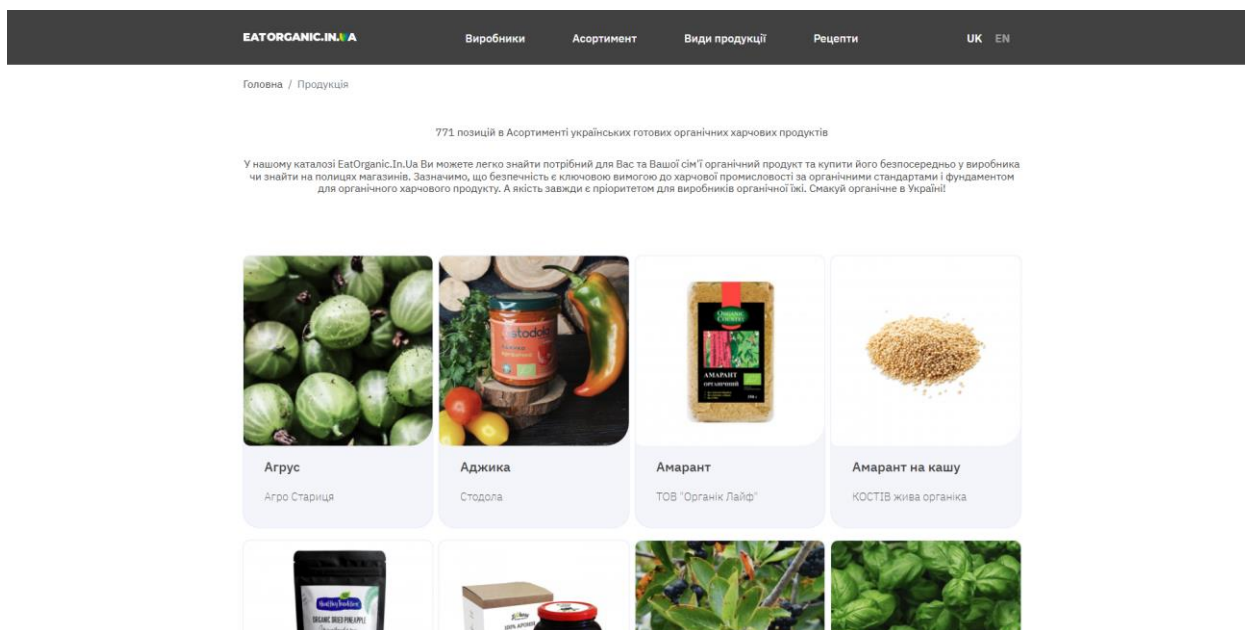


Рисунок 1.2.3 – Сторінка «Асортимент» на платформі «EatOrganic»

Тож для замовлення на цьому сайті потрібно зв'язуватися напряму з фермерами та підприємствами які виробляють продукт. Такий підхід не є ідеальним для звичайного покупця, адже сайт не має кошика, не займається власною торгівлею та не займається логістикою. Усі ці питання потрібно буде вирішувати саме кінцевому споживачу.

Інтернет-магазин Ahimsa – це типовий магазин органічної їжі. Наголошується що поля на яких вирощується продукція знаходяться в екологічно чистій зоні-заповіднику Дніпропетровської області. Асортимент органічної продукції налічує близько 300 товарів, серед яких безглютенова їжа, суперфуди, олії каші, горіхи, сухофрукти тощо. Каталог з вибором категорій виглядає зручним і зрозумілим. Проте дизайн виглядає трохи застарілим і перевантаженим, що трохи ускладнює навігацію та пошук необхідних товарів. До того ж, сайт не має адаптивного дизайну, тож користуватися ним буде дуже важко на мобільних пристроях. На рисунку 1.2.4 зображена головна сторінка інтернет-магазину «Ahimsa»

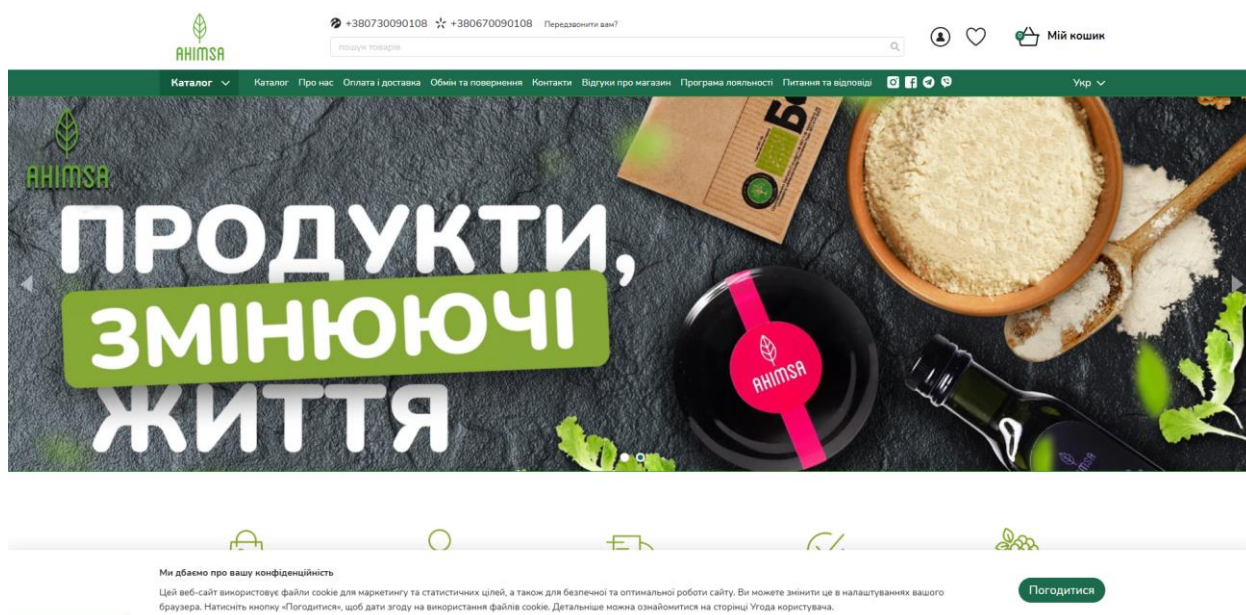


Рисунок 1.2.4 – Головна сторінка «Ahimsa»

Варто зазначити, що на кожному товарі вказана сертифікація продукції, для якої відбувається інспекція виробництва декілька разів на рік. В описі кожного продукту вказана харчова цінність, країна виробництва та дуже детальна інформація про нього, яка допомагає зрозуміти як він виготовлявся та як його використовувати. Як і в магазині «Смак життя» можна подивитися відгуки про кожний товар та залишити свій. Під час додавання товару до кошика автоматично встановлюється обмеження на кількість, яку можна замовити. Це зручно, тому що не потрібно зв'язуватися з менеджером для підтвердження наявності такої кількості товару.

На сайті присутній швидкий пошук серед усієї наявної продукції. Користувач може застосувати фільтри за ціною, брендом, новинкам, хітам та продукції без глютену. Товари можна відсортувати за дешевшими або дорожчими, за релевантністю, популярністю, та за назвою. У розділі «Питання та відповіді» можна отримати відповідь на найчастіші запитання. Програма лояльності спрямована на заохочення постійних клієнтів і збільшення лояльності до магазину.

Останнім сайтом для дослідження та порівняння було обрано інтернет-магазин «GreenFood». Цей сайт спеціалізується на продажу натуральних та

здорових продуктів. На сторінці «Про нас» акцентується увага, що усі продукти в магазині мають сертифікати якості та довгий термін придатності. Сайт має привабливий «екологічний» дизайн у біло-зелених кольорах. В асортименті пропонуються такі товари як бакалія, борошна, напої, крупи та насіння, олії, підсолоджувачі та вітаміни. Присутня також можливість швидкого пошуку необхідного товару за ключовими словами. На рисунку 1.2.5 зображена головна сторінка інтернет-магазину «GreenFood»

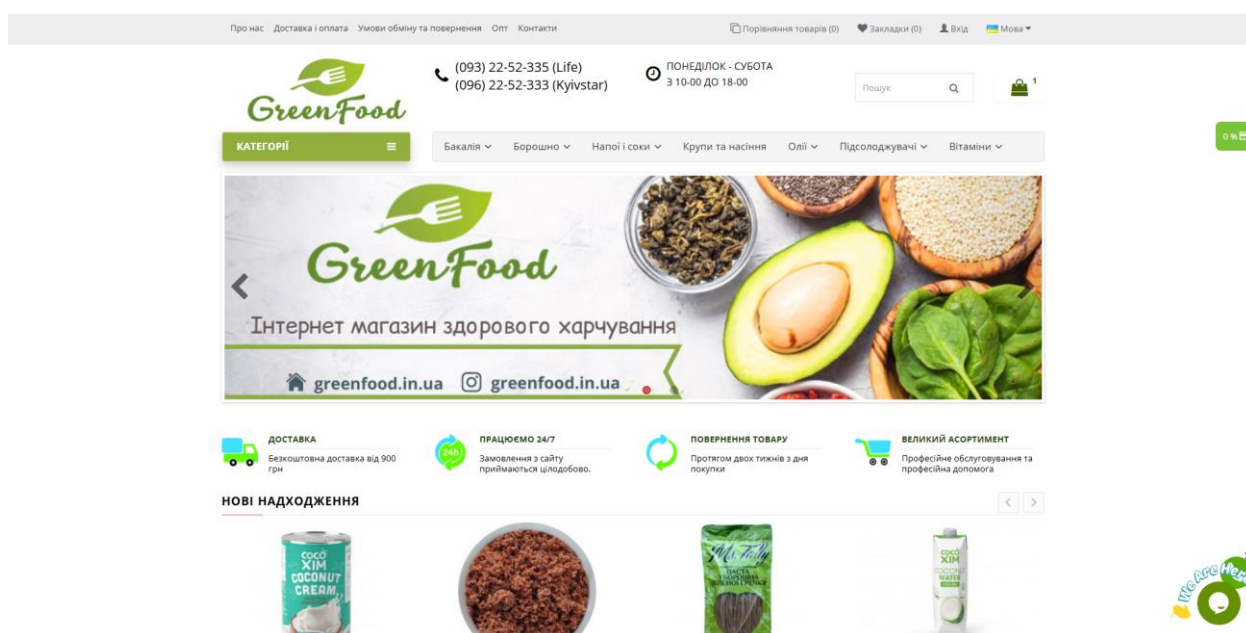


Рисунок 1.2.5 – Головна сторінка «GreenFood»

Товари у каталозі можна сортувати за веганськими, без глютенівими, вегетаріанськими та органічними. У більшості товарів є детальний опис з поясненням у яких стравах використовується цей продукт і рекомендації щодо того, як його правильно споживати. Інформація про виробника та поживну цінність зазначена не на всіх товарах.

На сайті присутня цікава можливість замовити потрібний товар «В один клік» прямо з каталогу, міняючи кошик. Це може бути зручно для деяких покупців, яким потрібно придбати лише щось одне. Серед інших корисних функцій можна виділити можливість додавати товари у закладки для

швидкого доступу до них, а також онлайн чат для вирішення питань пов'язаних з замовленням.

Використовуючи проведені дослідження можна скласти порівняльну таблицю (таблиця 1.2.1) чотирьох веб-сайтів з продажу органічної їжі за функціональними, технічними, комерційними та ергономічними вимогами.

Таблиця 1.2.1 – Порівняльна таблиця інтернет-магазинів

Вимоги	EatOrganic	Смак життя	Ahimsa	GreenFood
1	2	3	4	5
Функціональні вимоги Каталог товарів	Є, налічує понад 700 позицій	Є, налічує понад 800 позицій	Є, налічує приблизно 300 позицій	Є, налічує понад 500 позицій
Кошик	Відсутній	Є, швидке додавання товарів до кошику, автоматична перевірка наявності товарів	Є, автоматична перевірка наявності товарів	Є, потрібно перейти на сторінку товару для додавання товару до кошику
Процес замовлення	Потрібно зв'язуватись з виробником	Замовлення без реєстрації, можливість застосувати промокод, можна додати коментар до замовлення	Замовлення без реєстрації, можна додати коментар до замовлення	Можливість швидкого замовлення «за один клік», можливість замовлення без реєстрації, можна додати коментар до замовлення

Продовження табл. 1.2.1

1	2	3	4	5
Особистий кабінет	Відсутній	Є, вхід через соц. мережі та пошту, перегляд історії замовлень, можливість змінити особисту інформацію	Є, вхід через соц. мережі та пошту, перегляд історії замовлень, можливість змінити особисту інформацію	Є, вхід тільки через пошту, перегляд історії замовлень, запити на повернення, історія транзакцій, можливість змінити особисту інформацію
Інформація про продукти	Компанія виробник, торгова марка, сертифікація, опис товару відсутній або дуже малий	Детальний опис товарів, рекомендації до споживання, харчова цінність, умови зберігання, термін придатності	Детальний опис товарів, гарантія якості, вага, строк та умови зберігання, енергетична цінність, рекомендації до споживання.	Не всі товари мають детальний опис, енергетична цінність, сертифікація, використання, поради
Система пошуку та фільтри	Є пошук серед асортименту, немає фільтрів	Є пошук серед усього асортименту	Є зручний пошук, можливість застосувати фільтри	Є пошук, але неможливо продивитися усі товари у повному розмірі на одній сторінці
Адаптивність під мобільні пристрої	Є	Є	Немає	Є
Технічні вимоги Швидкість завантаження сайту	950 мс	1570 мс	1380 мс	1010 мс

Продовження табл. 1.2.1

1	2	3	4	5
Використані технології	Webpack, jQuery	-	PHP, jQuery, jQuery UI	Bootstrap, PHP, jQuery
Платіжні системи	Оплата на пряму виробнику	За реквізітам и IBAN	Готівка, переказом на картку, Монобанк	Онлайн-оплата карткою, Приватбанк, Монобанк, при отриманні товару
Служби доставки	Доставка узгоджується з виробником	Нова пошта поштою, відділення, кур'єр, самовивіз з магазину	Нова пошта поштою, відділення, кур'єр, самовивіз з магазину, укрпошта	Нова пошта поштою, відділення, кур'єр, самовивіз з магазину, укрпошта
Комерційні вимоги Акції та знижки	Відсутні	Є розділ «Ціна тижня», є знижки на певні категорії товарів	Знижок немає, неможливо відфільтрувати товари за знижкою	Є знижки на дуже малу кількість товарів
Програми лояльності	Відсутні	Є можливість ввести промокод при оформленні замовлення	Є накопичувальна знижка, яка розраховується від загальної суми попередніх замовлень	Є програма лояльності з накопичуваною знижкою, яка розраховується від загальної суми попередніх замовлень
Інформація про постачальників	Є. Вказана детальна інформація про виробника та його контакти	Немає. Вказана тільки країна виробництва товару	Немає. Вказана тільки країна виробництва товару	Є. Вказаний виробник та його короткий опис.

Продовження табл. 1.2.1

1	2	3	4	5
Клієнтська підтримка	Ім'я виробника, email, номер телефону, іноді соц.мережі та сайт компанії	Онлайн чат, телефон, месенджер	Телефон, email, соц.мережі, месенджери	Онлайн чат, телефон, email, адреса
Ергономічні вимоги Інтерфейс користувача	Сучасний, зрозумілий, приємний дизайн, якісні фото товарів	Простий, зрозумілий, зручний дизайн, легка навігація по сайту	Трохи застарілий дизайн, якісні фото товарів	Простий, зрозумілий інтерфейс, легко знайти потрібну інформацію.
Відповіді на типові запитання	Немає	Є інформація про доставку та оплату	Є відповіді на часті питання	Є інформація про доставку та оплату
Читабельність контенту	Висока, контент добре структурований та зрозумілий	Висока, кольоровим і шрифтами виділені основні моменти тексту	Контент добре структурований	Хороша читабельність, приємні кольори, в описі товарів важлива інформація виділена кольоровим та жирним шрифтом
Загальна зручність використання	Зручне використання, сучасний дизайн, легко знайти інформацію про виробників	Зручний та інтуїтивно зрозумілий інтерфейс	Легко знайти товар та оформити замовлення	Загалом зручний, легкий для навігації.

Можна дійти висновку, що при створенні власного інтернет-магазину потрібно орієнтуватися на успішні рішення існуючих аналогів. Зокрема, варто розглянути можливість враховувати персоналізацію до клієнтів, яка за останні роки значно розвилася від простого звернення до клієнта на ім'я, до складних маркетингових стратегій, що базуються на аналізі поведінки, демографії, історії покупок користувачів, тощо [24]. Такий підхід може дозволити налаштовувати пропозиції під кожного клієнта окремо, підвищуючи рівень його задоволеності.

1.3 Розробка технічного завдання

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку веб-додатку для реалізації органічної їжі.

1. Вступ

1.1 Загальні визначення

Повна назва веб-додатку: інтернет-магазин для продажу органічної їжі «EcoFeast».

Замовник – організація, яка замовляє створення веб-додатку.

Виконавець – програміст, який розробляє веб-додаток. В даному випадку це студент 4 курсу Таврійського Державного Агротехнологічного Університету, Васковський Артем Сергійович.

Vue.js – сучасний JavaScript фреймворк для створення користувацьких інтерфейсів.

Django – сучасний Python фреймворк, який дозволяє розробляти веб-застосунки та може використовуватися як серверна частина веб-додатку.

Django REST framework – потужний та гнучкий інструмент для створення Web API на базі Django.

MySQL – реляційна система управління базами даних.

API – програмний інтерфейс, який дозволяє взаємодіяти клієнтській та серверній частинам додатків.

Ендпоінт (англ. Endpoint) – це конкретна URL-адреса у додатку, через який зовнішні системи можуть отримувати доступ до даних, функціональних можливостей або послуг, які пропонує система.

1.2 Призначення документу

У цьому документі приводиться повний перелік вимог до програмної реалізації та впровадження програмного засобу. У цьому документі Замовник і Виконавець підтверджують їх згоду з нижченаведеними фактами і умовами:

1. Виконавець підготував і розробив цей документ, що іменується Технічне Завдання, який містить перелік вимог до виконуваних робіт.
2. Замовник згоден зі всіма положеннями цього Технічного Завдання.
3. Замовник не має права вимагати від Виконавця в рамках поточного Договору виконання робіт або надання послуг, прямо не описаних у справжньому Технічному Завданні.
4. Виконавець зобов'язується виконати роботи в об'ємі, вказаному в справжньому Технічному Завданні.
5. Замовник не має права вимагати від Виконавця дотримання яких-небудь форматів і стандартів, якщо це не вказано в справжньому Технічному Завданні.
6. Усі неоднозначності, виявлені в справжньому Технічному завданні після його підписання, підлягають двосторонньому узгодженню між Сторонами. В процесі узгодження можуть бути розроблені додаткові вимоги, які оформляються додатковою угодою до Договору і відповідним чином оцінюються.

2. Призначення розробки

2.1 Призначення і мета розробки програмного засобу

Предметом розробки є інтернет-магазин «EcoFeast», призначений для

продажу органічних продуктів харчування кінцевим споживачам.

Метою розробки є створення сучасного, зручного та функціонального веб-додатку, який дозволить:

- клієнтам ознайомлюватися з асортиментом товарів, здійснювати пошук та фільтрацію товарів, додавати товари до кошика та оформлювати замовлення;
- адміністратору додавати, редагувати, видаляти товари з асортименту, встановлювати ціну та знижки на товари, додавати та змінювати інформацію про товар, змінювати контент на інформаційних сторінках та обробляти замовлення.

Програмний засіб призначений для зберігання інформації про товари (опис, ціна, наявність, медіа-файли тощо), клієнтів (контактні дані, історія замовлень) і замовлення.

2.2 Характеристика об'єктів автоматизації

Розробка інтернет-магазину допоможе бізнесу автоматизувати процеси продажу товарів, прийняття замовлень, складського обліку та взаємодії з клієнтами в цілому.

Однією з головних проблем підприємств, які орієнтуються на продаж товарів є неефективність ручного прийому замовлень через стандартні способи зв'язку, як наприклад соціальні мережі або телефон. Веб-додаток дозволяє автоматизувати процес вибору товарів та наповнення кошика, обробки замовлень, підтвердження замовлень, тощо. Це також дозволить зменшити кількість потенційних помилок у оформленні замовлення, бо клієнт сам перевіряє правильність введених даних.

Відсутність централізованої системи обліку товарів може призвести до проблем із точним відображенням наявної продукції на сайті, що може збити з пантелику користувачів. Так, наприклад, користувач може обрати товар та

вказати кількість, яка перевищує реальну кількість товарів на складі. Система зможе автоматично вказати ліміт і підказати користувачу скільки насправді товарів цього типу залишилось. При замовленні товару система автоматично спише ту кількість, яка була замовлена та оновить цю інформацію у базі даних.

Не менш важливим є створення розділів на сайті з відповідями на часті питання. Такий підхід допоможе користувачам одразу отримати відповідь на потрібне питання, що у свою чергу зменшить навантаження та кількість типових звернень до клієнтської підтримки.

3. Вимоги до програмного засобу

3.1 Вимоги до оформлення інтерфейсу

Інтерфейс користувача є одним з найважливіших етапів у створенні веб-сайту, тому що від зручності та зрозумілості сайту залежить те, як швидко користувач зможе знайти потрібний товар та оформити замовлення. Саме тому дизайн інтерфейсу повинен бути інтуїтивно зрозумілим, мінімалістичним, сучасним та відповідати естетиці, яка асоціюється з органічними продуктами. Головна сторінка повинна мати навігаційну панель для зручної навігації по сайту, містити тимчасові акційні пропозиції, найпопулярніші категорії товарів та найпопулярніші товари (рис. 1.3.1).



Рисунок 1.3.1 – Інтерфейс головної сторінки

Каталог товарів повинен містити список товарів з яскравими зображеннями продуктів, їх назвою, ціною (зі знижкою або без), рейтингом та кнопками, які дозволять додати цей товар у кошик або замовити цей товар «В один клік». Для швидкого пошуку по каталогу необхідно мати фільтри та сортування за категоріями, ціною, рейтингом та тегами. Фільтри допоможуть користувачу швидко звужити вибір до тих товарів, які відповідають його потребам. На рисунку 1.3.2 наведено приблизну схему графічного інтерфейсу каталогу товарів.



Рисунок 1.3.2 – Каталог товарів

Сторінка товару повинна повністю ознайомити споживача із цим товаром. Така сторінка повинна містити зображення товару, ціну, рейтинги інших користувачів, детальну та коротку інформацію про товар, сертифікацію та посилання на виробника, умови зберігання товару, термін його придатності.

Для того, щоб зацікавити користувача зі схожою продукцією, на сторінці також повинен бути розділ зі схожими товарами та посиланням на них.

Функціонально сторінка з товаром повинна давати можливість додати товар до кошика для подальшого замовлення, збільшити або зменшити кількість товарів які додаються до кошика та продивитись і залишити відгуки з рейтингом.



Рисунок 1.3.3 – Сторінка товару

Варто зауважити, що графічний інтерфейс користувача повинен бути адаптований під усі розміри пристроїв, для того щоб сайт був зручним у користуванні як зі смартфона так і з персонального комп'ютера.

3.2 Вимоги до зберігання даних і мов програмування

Для зберігання даних використовується СУБД MySQL. Графічний інтерфейс реалізується за допомогою JavaScript фреймворку Vue.js у середовищі розробки Microsoft Visual Studio Code, а серверна частина – за допомогою мови програмування Python та фреймворку Django.

3.3 Вимоги до розділення доступу

Повноцінна можливість взаємодії з додатком передбачається як для зареєстрованих так і для не зареєстрованих користувачів. Обидві групи користувачів можуть переглядати каталог товарів, додавати товари у кошик та оформлювати замовлення. Відрізняються ці групи наявністю доступу до особистого кабінету, у якому можна передивитися історію замовлень, вказати особисті дані та дані доставки для більш швидкого здійснення майбутніх покупок. На додачу до цього, зареєстровані користувачі також мають можливість залишати відгуки до товарів.

Адміністратор має доступ до редагування інформації про товари, зміни тексту на інформаційних сторінках та переглядання списку замовлень. Вхід в обліковий запис адміністратора здійснюється за допомогою паролю адміністратора за спеціальним посиланням.

3.4 Вимоги до програмного та технічного забезпечення

При розробці веб-додатку необхідно забезпечити можливість запуску веб-додатку при наступній мінімальній конфігурації комп'ютера:

- процесор з частотою 1 ГГц;
- оперативна пам'ять обсягом щонайменше 1 ГБ;
- монітор із роздільною здатністю щонайменше 800x600 або смартфон з роздільною здатністю екрана щонайменше 320x480;

- відеокарта з обсягом відеопам'яті не менше 128 МБ;
- браузер із підтримкою HTML5, CSS3 та JavaScript;
- доступ до мережі Інтернет.

Створюючи веб-сайт потрібно також орієнтуватися на мобільні пристрої. Мінімальна конфігурація мобільного пристрою повинна бути такою:

- операційна система Android 7.0 або IOS 12.0 і вище;
- процесор з частотою 1 ГГц;
- оперативна пам'ять обсягом не менше 1 ГБ;
- екран з роздільною здатністю щонайменше 320x480;
- сучасний браузер із підтримкою HTML5, CSS3 та JavaScript;
- доступ до інтернету.

4. Стадії і етапи розробки програмного засобу

4.1 Склад і зміст робіт із розробки програмного засобу

Розробка інтернет-магазину з продажу органічної їжі припускає виконання наступних етапів:

1. Аналіз вимог та планування з уточненням функціональних та нефункціональних вимог.
2. Проєктування бази даних, API, користувацького інтерфейсу.
3. Розробка backend логіки, що включає у себе налаштування оточення, створення моделей даних Django, розробка бізнес-логіки, реалізація API ендпоінтів, розробка адміністративної панелі, інтеграція з СУБД MySQL.
4. Розробка frontend, а саме: верстка сторінок на основі макету, розробка компонентів Vue.js, реалізація логіки взаємодії з API, створення адаптивності інтерфейсу під різні пристрої.
5. Перевірка та тестування додатку, виправлення помилок.
6. Розгортання додатку на робочому сервері і фінальне тестування працездатності.

7. виправлення помилок у роботі програми і проведення завершального показу Замовнику.

5. Порядок контролю і приймання

5.1 Перевірка працездатності роботи програми

Перевірка працездатності роботи веб-додатку здійснюється за наступним сценарієм:

1. Реєстрація нового користувача та авторизація.
2. Перегляд каталогу товарів, перевірка роботи фільтрів та пошуку.
3. Додавання товарів до кошику та редагування його вмісту.
4. Оформлення та підтвердження замовлення.
5. Перевірка наявності нового замовлення у історії замовлень в особистому кабінеті, перевірка можливості редагувати особисті дані.
6. Авторизація в обліковий запис адміністратора та перевірка можливості додавати, редагувати, видаляти товари, керувати замовленнями та змінювати інформацію на інформаційних сторінках.
7. Перевірка правильності обміну даними між фронтендом та бекендом.
8. Перевірка коректності збереження інформації у базі даних MySQL.
9. Перевірка коректної роботи у сучасних браузерях та наявність адаптивного дизайну.

5.2 Вимоги до введення програмного засобу в дію

1. Розгортання та налаштування програмного засобу на робочому сервері.
2. Наявність доступу до панелі адміністратора.
3. Початкова наповненість бази даних товарами.
4. Проведення інструктажу Замовнику щодо роботи програмного засобу.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Глосарій

Таблиця 2.1 – Глосарій

Термін	Опис терміну
1	2
1. Основні поняття та категорії предметної області	
Веб-додаток	Програмне забезпечення, доступ до якого здійснюється через веб-браузер.
Органічна їжа	Продукти харчування, вирощені без використання синтетичних добрив, пестицидів, гормонів росту та ГМО.
Інтернет-магазин	Веб-додаток, який дозволяє купувати товари через інтернет.
HTML	Мова гіпертекстової розмітки, яка описує структуру веб-сторінки.
CSS	Мова стилів, що використовується для опису зовнішнього вигляду веб-сторінки, написаних мовою розмітки HTML.
JavaScript	Мова програмування, яка дозволяє створювати інтерактивні елементи на веб-сторінці.
Фреймворк	Набір готових інструментів, бібліотек, шаблонів та правил, які допомагають швидше і зручніше створювати програмне забезпечення.
Vue.js	Фреймворк мови програмування JavaScript, який використовується для створення користувацьких інтерфейсів за допомогою реактивного зв'язування даних та компонентного підходу.
Django	Фреймворк мови програмування Python, який допомагає розробляти серверну частину веб-додатків.

Продовження табл. 2.1.

1	2
Фронтенд	Частина веб-додатку, з якою взаємодіє користувач і яка будується за допомогою мов розмітки та програмування HTML, CSS та JavaScript.
Бекенд	Серверна частина веб-додатку, яка відповідає за обробку даних, взаємодію з базою даних та бізнес-логіку. Реалізацію серверної частини користувач не бачить і може взаємодіяти з нею тільки через клієнтську частину.
База даних	Організоване сховище даних, яке використовується для зберігання.
Юзабіліті (анг. Usability)	Зручність та простота використання веб-сайту для користувачів.
Компонент	Багаторазова, незалежна частина інтерфейсу користувача у якій описана HTML структура, CSS стилі та JavaScript логіка.
Адаптивний дизайн	Підхід до веб-дизайну, спрямований на автоматичне пристосування інтерфейсу користувача до різних розмірів екранів.
Реактивність	Функція, яка дозволяє додатку автоматично оновлювати користувацький інтерфейс, коли змінюються дані.
2. Користувачі системи	
Клієнт	Особа, що має можливість взаємодіяти тільки з клієнтською частиною веб-додатку (перегляд асортименту, пошук, фільтрація, додавання товарів у кошик, оформлення замовлення тощо.)
Адміністратор	Особа, яка має доступ до адміністративної частини веб-додатку для управління товарами, цінами, знижками, замовленнями та наповненням сторінок,
3. Вхідні та вихідні документи	
Технічне завдання	Документ, який визначає цілі, вимоги та функціональність розроблюваного програмного продукту.
Реєстраційна форма користувача	Форма для створення облікового запису клієнта

Продовження табл. 2.1.

1	2
Інформація про товари	Назва, зображення, опис, харчова цінність, рекомендації щодо вживання, умови зберігання, термін придатності, інформація про виробника та його сертифікати.
Особисті дані користувачів	Інформація, що вводиться користувачами при оформленні замовлення або реєстрації у веб-додатку.
Дані про оплату та доставку	Необхідна інформація для здійснення транзакції та логістики.
Підтвердження замовлення	Повідомлення, яке відображається клієнту після успішного замовлення.
Історія замовлень	Запис усіх замовлень користувача з детальною інформацією про склад замовлень.

2.2 Специфікація функціональних та нефункціональних вимог

У цьому розділі буде детально розглянуто функціональні та нефункціональні вимоги до інтернет-магазину з продажу органічної їжі. Це допоможе детальніше зрозуміти вимоги до системи, які функції вона має виконувати та його критерії якості оцінки.

Функціональні вимоги описують конкретні дії, які система має виконувати. Вони описують що саме повинна робити система, включаючи можливість входу в систему, здійснення платежів та інші функції програмного продукту. У таблиці 2.2.1 наведені функціональні вимоги до інтернет-магазину з продажу органічної їжі.

Таблиця 2.2.1 – Специфікація функціональних вимог

Ідентифікатор вимоги	Назва вимоги (варіанта використання)	Атрибути вимог		
		пріоритет	складність	Контакт/виконавець
1	2	3	4	5

Продовження табл. 2.2.1.

1	2	3	4	5
ФВ-001	Управління користувачами			
ФВ-001.1	Реєстрація користувачів	Обов'язкове	Середня	
ФВ-001.2	Авторизація користувачів	Обов'язкове	Середня	
ФВ-001.3	Редагування особистих даних	Обов'язкове	Низька	
ФВ-001.4	Управління адресами доставки	Обов'язкове	Середня	
ФВ-001.5	Зміна пароллю	Рекомендоване	Низька	
ФВ-002	Каталог товарів			
ФВ-002.1	Перегляд асортименту товарів	Обов'язкове	Середня	
ФВ-002.2	Пошук товарів за ключовими словами	Обов'язкове	Середня	
ФВ-002.3	Фільтрація товарів (за ціною, наявністю, виробником, атрибутами)	Обов'язкове	Висока	
ФВ-002.4	Сортування товарів	Обов'язкове	Середня	
ФВ-002.5	Перегляд інформації про товар (назва, зображення, опис)	Обов'язкове	Низька	
ФВ-002.6	Надання інформації про харчову цінність	Обов'язкове	Низька	
ФВ-002.7	Надання рекомендацій щодо вживання	Рекомендоване	Низька	
ФВ-002.8	Відображення умов зберігання та терміну придатності	Обов'язкове	Низька	

Продовження табл. 2.2.1.

1	2	3	4	5
ФВ-002.9	Відображення інформації про виробника та його сертифікати	Обов'язкове	Низька	
ФВ-002.10	Можливість переглядати та залишати відгуки	Обов'язкове	Висока	
ФВ-003	Кошик та оформлення замовлення			
ФВ-003.1	Додавання товарів у кошик	Обов'язкове	Середня	
ФВ-003.2	Видалення товарів з кошика	Обов'язкове	Низька	
ФВ-003.3	Зміна кількості товарів у кошику	Обов'язкове	Низька	
ФВ-003.4	Автоматичне обмеження на максимальну кількість одиниць товару	Рекомендоване	Середня	
ФВ-003.5	Підрахунок загальної вартості замовлення	Обов'язкове	Середня	
ФВ-003.6	Спрощений процес оформлення замовлення без реєстрації	Обов'язкове	Висока	
ФВ-003.7	Введення контактних даних при замовленні	Обов'язкове	Низька	
ФВ-003.8	Вибір способу оплати	Обов'язкове	Середня	
ФВ-003.9	Вибір способу доставки	Обов'язкове	Середня	
ФВ-003.10	Підтвердження замовлення	Обов'язкове	Низька	
ФВ-003.11	Можливість замовити «В один клік»	Рекомендоване	Середня	

Продовження табл. 2.2.1.

ФВ-004	Особистий кабінет			
ФВ-004.1	Перегляд історії замовлень	Обов'язкове	Середня	
ФВ-004.2	Перегляд запитів на повернення товарів	Опційне	Середня	
ФВ-004.3	Зміна паролю	Рекомендоване	Низька	
ФВ-004.4	Зміна адреси доставки	Обов'язкове	Середня	
ФВ-005	Інформаційні розділи			
ФВ-005.1	Розділ «Про нас»	Обов'язкове	Середня	
ФВ-005.2	Розділ «Часті запитання»	Обов'язкове	Середня	
ФВ-005.3	Розділ «Умови доставки та оплати»	Обов'язкове	Середня	
ФВ-005.4	Розділ «Акції та програми лояльності»	Обов'язкове	Середня	
ФВ-006	Адміністративна панель			
ФВ-006.1	Додавання, редагування, видалення товарів	Обов'язкове	Середня	
ФВ-006.2	Управління категоріями та під-категоріями	Обов'язкове	Середня	
ФВ-006.3	Додавання та редагування наповнення товарів	Обов'язкове	Середня	
ФВ-006.4	Перегляд, пошук та фільтрація замовлень	Обов'язкове	Середня	
ФВ-006.5	Зміна статусу замовлення	Обов'язкове	Низька	
ФВ-006.6	Деталі замовлення	Обов'язкове	Середня	

Продовження табл. 2.2.1.

ФВ-006.7	Редагування інформаційних сторінок	Обов'язкове	Середня	
ФВ-006.8	Управління новинами та акціями	Обов'язкове	Середня	
ФВ-006.9	Можливість редагувати ціни товарів	Обов'язкове	Середня	
ФВ-006.10	Можливість встановлювати знижки на товари	Обов'язкове	Середня	

Нефункціональні вимоги – це вимоги до якості роботи системи, наприклад її швидкодії, зручності користування, продуктивності тощо. Для інтернет-магазину з продажу органічної їжі такі вимоги описані у таблиці 2.2.2.

Таблиця 2.2.2 – Специфікація нефункціональних вимог.

№	Назва вимоги	Характеристика
1	2	3
1	Застосовність	
1.1	Час для навчання користувачів	Інтерфейс інтуїтивно зрозумілий. Короткий час для ознайомлення з функціоналом сайту.
1.2	Вимірюваний час відгуку для типових завдань	Час завантаження сторінки до 2 секунд. Час додавання товару до кошика: до 1 секунди. Час оформлення замовлення: до 2 секунд
2	Надійність	
2.1	Середній час безвідмовної роботи	Мінімум 99.5% доступності сервісу на місяць.
2.2	Середнє напруження до ремонту	Прогнозований час до виникнення критичної помилки не менше 500 годин роботи
3	Робочі характеристики	
3.1	Швидкодія для транзакції	Час обробки замовлення: до 5 секунд.

Продовження табл. 2.2.2.

1	2	3
3.2	Продуктивність	Обробка до 500 замовлень на день. Здатність обробляти до 500 запитів до бази даних на хвилину.
4	Проектні обмеження	
4.1	Мова програмування	JavaScript, Python
4.2	Фреймворки та бібліотеки	Vue.js (для фронтенду), Django (для бекенду).
4.3	Бази даних	MySQL
5	Вимоги до документації, призначеної для користувача і до системи	Розділ FAQ, контакти клієнтської підтримки.
6	Інтерфейси	
6.1	Інтерфейси користувача	Зручний інтерфейс для швидкої навігації по сайту, адаптивний дизайн для різних пристроїв.
6.2	Програмні інтерфейси	API для взаємодії між фронтендом та бекендом.
7	Вимоги до ліцензування	Використання відкритого програмного забезпечення та дотримання умов у використанні матеріалів з авторським правом.
8	Вживані стандарти	Відповідність стандартам HTML5, CSS3, ES6+.

2.3 Концептуальна модель використання інформаційної системи

Цей підрозділ присвячений створенню концептуальної моделі використання інформаційної системи для інтернет магазину з продажу органічної продукції. Для моделювання діаграми варіантів використання, специфікації варіантів використання та розкадровки варіантів використання буде використано уніфіковану мову моделювання UML.

Діаграма варіантів використання допоможе візуалізувати функціональні вимоги до системи з точки зору акторів та визначити основні взаємодії між акторами та системою (рисунок 2.3.1).

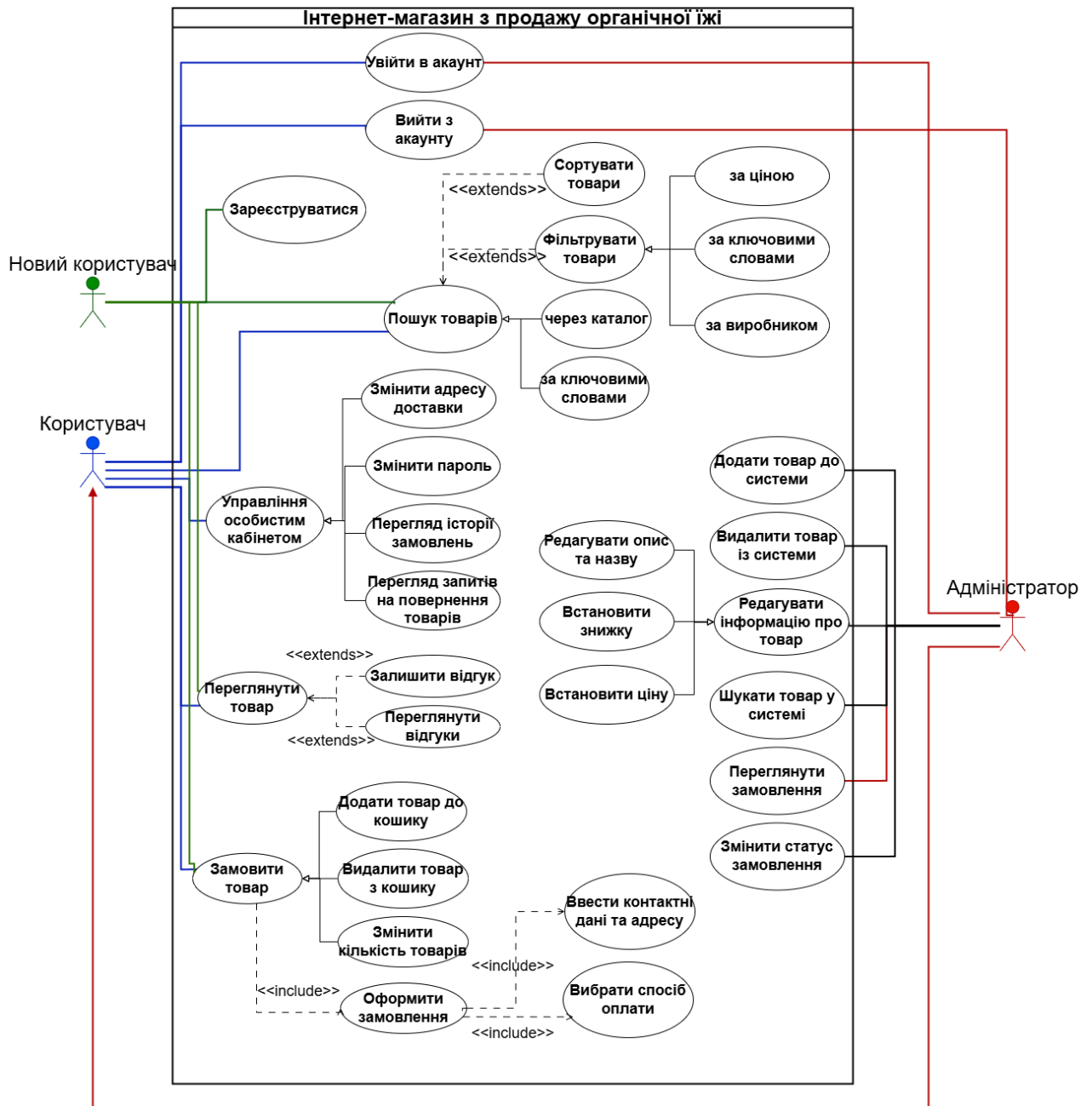


Рисунок 2.3.1 – Діаграма варіантів використання для інтернет-магазину з продажу органічної їжі.

Для деталізації ключових варіантів використання, які реалізують основну функціональність складаються їх специфікації у вигляді таблиці. У таблицях 2.3.1 – 2.3.3 наведені специфікації для важливих варіантів використання.

Таблиця 2.3.1 – Специфікація варіанту використання «Реєстрація користувача»

Реєстрація користувача	
Контекст використання	Дозволяє новому користувачу створити обліковий запис у системі для подальшого здійснення покупок та використання функціоналу особистого кабінету.
Дійові особи	Новий користувач, система (зберігає дані користувача)
Передумова	Користувач ще не зареєстрований та не авторизований у системі.
Тригер	Користувач натискає на кнопку «Реєстрація»
Сценарій	1. Користувач переходить на сторінку реєстрації. 2. Система відображає форму реєстрації. 3. Користувач вводить необхідні дані. 4. Користувач натискає кнопку «Зареєструватися» 5. Система перевіряє правильність даних (формат email, чи email вже існує у системі та чи збігаються паролі). 6. У разі успішної валідації, система створює новий обліковий запис. 7. Автоматична авторизація користувача.
Постумова	У системі створено новий обліковий запис користувача. Користувач авторизований. У разі невдалої реєстрації, дані залишаються незмінними.

Таблиця 2.3.2 – Специфікація варіанту використання «Оформлення замовлення»

Оформлення замовлення	
Контекст використання	Дозволяє користувачу здійснити покупку товарів доданих до кошика. У результаті формується нове замовлення, яке відображається у системі та надсилається на обробку адміністратору.
Дійові особи	Користувач, Система (оброблює замовлення, підраховує вартість, додає у систему)
Передумова	Користувач має товари у кошику
Тригер	Користувач натиснув кнопку «Оформити замовлення» у кошику, або натиснув кнопку «Купити в один клік».

Продовження табл. 2.3.2.

Сценарій	1. Користувач переходить на сторінку оформлення замовлення. 2. Система відображає список товарів у кошику, їх кількість та вартість. 3. Незареєстрований користувач вводить контактні дані. 4. Вибір способу доставки 5. Вибір способу оплати 6. Користувач перевіряє введені дані та натискає на кнопку «Підтвердити замовлення» 7. Система створює нове замовлення. 8. Система оновлює інформацію про наявність товарів на складі. 9. Система надсилає підтвердження замовлення користувачу та адміністратору. 10. У разі вибору онлайн-оплати, система перенаправляє користувача на сторінку платіжного сервісу.
Постумова	Створено нове замовлення, користувач отримав підтвердження про створення замовлення, оновлено інформацію про наявність товарів.

Таблиця 2.3.3 – Специфікація варіантів використання «Додавання, редагування, видалення товарів»

Додавання, редагування, видалення товарів	
Контекст використання	Дозволяє адміністратору керувати асортиментом товарів. Успішне завершення – додавання нового товару, оновлення інформації існуючого товару або видалення товару з каталогу.
Дійові особи	Адміністратор Система (зберігає дані, відображає товари у каталозі)
Передумова	Адміністратор авторизований у системі.
Тригер	Перехід до розділу «Управління товарами» в панелі адміністратора та вибір дії.

Продовження табл. 2.3.3.

Сценарій	Додавання товару. 1. Адміністратор натискає кнопку «Додати товар». 2. Система відображає форму додавання товару. 3. Адміністратор заповнює необхідні дані про товар. 4. Адміністратор натискає на кнопку «Зберегти». 5. Система валідує дані та додає новий товар до каталогу. Редагування товару. 1. Адміністратор вибирає товар зі списку. 2. Система показує форму редагування даних вибраного товару. 3. Адміністратор вносить зміни до даних. 4. Адміністратор натискає кнопку «Зберегти». 5. Система валідує дані та оновлює інформацію про товар. Видалення товару. 1. Адміністратор обирає товар для видалення. 2. Система запитує підтвердження видалення. 3. Адміністратор підтверджує видалення. 4. Система видаляє товар з каталогу.
Постумова	Інформація про товар або каталог оновлена.

Розкадровка варіантів використання дозволяє візуалізувати поведінку користувачів у різних сценаріях взаємодії із системою. Це досягається за допомогою ілюстрацій, на яких зображені ключові етапи виконання певного сценарію використання.

Розглянемо варіант використання «Реєстрація користувача». На рисунку 2.3.2 зображена форма реєстрації нового користувача з такими полями для заповнення: пошта, пароль та підтвердження паролю.

Реєстрація

Пошта

Пароль

Підтвердження паролю

Кнопка "Зареєструватися"

Вже маєте обліковий запис?
Увійти

Рисунок 2.3.2 – Форма реєстрації нового облікового запису.

Альтернативним сценарієм успішної реєстрації може бути помилка валідації паролів. Така помилка може статися коли користувач помилився у повторному вводиті паролю (рисунок 2.3.3).

Реєстрація

Пошта

Пароль

Підтвердження паролю

Кнопка "Зареєструватися"

Вже маєте обліковий запис?
Увійти

Паролі не співпадають

Рисунок 2.3.3 – Помилка валідації паролю при реєстрації нового облікового запису

При успішному вводиті користувачем адреси електронної пошти та паролів та натисканні на кнопку «Зареєструватися», користувача автоматично авторизує та перенаправляє до особистого кабінету (рисунок 2.3.4). На цьому етапі процес реєстрації успішно завершений.



Рисунок 2.3.4 – Особистий кабінет

Розглянемо варіант використання «Оформлення замовлення» користувачем.

Після успішного додавання товарів до кошику, користувач переходить до етапу оформлення замовлення (рисунок 2.3.5). На цьому етапі незареєстрований користувач повинен ввести особисті дані, адресу доставки, країну та електронну адресу для повідомлень. За бажанням користувач може дописати побажання до замовлення. Після перевірки правильності введених даних користувач натискає на кнопку «Розмістити замовлення».

Інформація для виставлення рахунку	Підсумки замовлення
Особисті дані	Список товарів
Адреса	Підсумкова сума
Країна	Ціна за доставку
Електронна адреса	Всього
Додаткові побажання до замовлення	Метод оплати
	Кнопка "Розмістити замовлення"

Рисунок 2.3.5 – Форма оформлення замовлення

Якщо користувач не ввів дані необхідні для оформлення замовлення, система не завершить таке замовлення і повторно запитає вказати цю інформацію (рисунки 2.3.6).

Інформація для виставлення рахунку	Підсумки замовлення
Особисті дані Дані не вказані	Список товарів
Адреса Дані не вказані	Підсумкова сума
Країна Дані не вказані	Ціна за доставку
Електронна адреса	Всього
Додаткові побажання до замовлення	Метод оплати
	Кнопка "Розмістити замовлення"

Рисунок 2.3.6 – Оформлення замовлення не вдалося через відсутність інформації про користувача.

Якщо користувач вже був зареєстрований у системі та авторизований своєму обліковому записі, особисті дані, адреса доставки та країна автоматично будуть вказані у поля оформлення замовлення. Після успішного вводу інформації та підтвердження замовлення, система сповіщає користувача про успішне оформлення замовлення.

Оформлення замовлення

Замовлення успішно виконане!

Рисунок 2.3.7 – Сповіщення користувача про успішно виконане замовлення.

Варіант використання «Додавання та редагування товарів».

Для додавання та редагування товарів, особі необхідно авторизуватися за логіном та паролем адміністратора та потрапити у панель адміністратора. Адміністратор може змінити інформацію про вже існуючий товар або додати новий (інтерфейси ідентичні). У товарі вказується його назва, опис, ціна, категорія до якої належить цей товар, його зображення та бренд виробника (рисунок 2.3.8).

Панель адміністратора

Категорії	+	Додати / змінити продукт
Замовлення	+	
Платежі	+	
Продукти	+	
Профілі	+	
Відгуки	+	
Бренд	+	

Назва	
Опис	
Ціна	
Категорія	
Зображення	
Бренд	

Зберегти

Рисунок 2.3.8 – Форма для додавання та зміни інформації про продукти у панелі адміністратора

Варіант використання «Видалення товарів».

Для видалення товарів з асортименту, адміністратору достатньо обрати необхідні товари зі списку та обрати дію «Видалити товари» (рисунок 2.3.9)

The image shows a web interface titled "Панель адміністратора" (Administrator Panel). On the left is a sidebar menu with the following items: Категорії (+), Замовлення (+), Платежі (+), **Продукти (+)** (highlighted), Профілі (+), Відгуки (+), and Бренд (+). The main content area is titled "Видалити товар" (Delete product). Below this title is a button labeled "Видалити товари" (Delete products). In the center of the main area is a large rectangular box containing the text "Список товарів" (List of products).

Рисунок 2.3.9 – Форма для видалення продуктів у панелі адміністратора

2.4 Розробка функціональної моделі

Функціональна модель дозволяє описати основні процеси, які відбуваються в межах програмного продукту. Головною метою побудови такої моделі є визначення основних функцій системи, їх взаємозв'язки і вхідні та вихідні дані. Для побудови такої моделі зазвичай використовується методологія IDEF0.

IDEF0 – це методологія функціонального моделювання, яка дозволяє графічно описати процеси системи у вигляді ієрархії функціональних блоків. Кожен блок являє собою «чорну скриньку» з чітко визначеними входами та виходами, елементами управління та механізмами.

Розробка функціональної моделі дозволить візуально представити структуру системи та побачити як функції взаємопов'язані між собою, що у свою чергу дозволить краще зрозуміти логіку роботи системи.

Для визначення функціональних вимог до інформаційної системи «Інтернет-магазин з продажу органічних продуктів», була розроблена

функціональна модель за стандартом IDEF0. На рисунку 2.4.1 зображена діаграма A0.

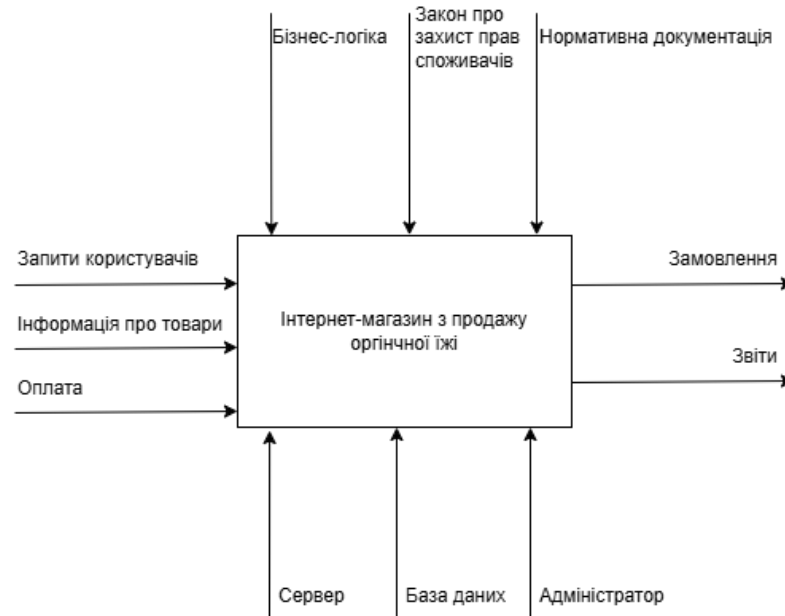


Рисунок 2.4.1 – Контекстна діаграма інтернет-магазину з продажу органічної їжі в нотації IDEF0

На цій діаграмі центральний блок представляє собою всю розроблювану систему зі стрілками, які входять та виходять із неї. Стрілки, які входять у систему з лівої сторони називаються вхідними. Це ті матеріали чи дані, які система оброблює. У випадку інтернет-магазину, у таку систему надходять запити користувачів, оплата та інформація про товари. Стрілки що виходять із системи праворуч є результатом обробки вхідних даних, у цьому випадку це замовлення та звіти. Стрілки зверху – це стрілки управління, вони вказують обмеження, правила та політики які безпосередньо керують виконанням функції. Знизу у систему надходять механізми. Механізмами можуть бути засоби, ресурси або виконавці, які реалізують функцію.

Розроблена діаграма дозволяє поверхнево отримати уявлення про основні процеси та взаємозв'язки у майбутньому інтернет-магазині і виступає основою для подальшої деталізації системи.

РОЗДІЛ 3

ОПИС ПРИЙНЯТИХ ПРОЄКТНИХ ТА ТЕХНОЛОГІЧНИХ РІШЕНЬ

3.1 Розробка об'єктної моделі

Об'єктна модель дозволяє абстрактно представити структуру, властивості, поведінку та зв'язки основних сутностей системи. Вона допомагає зрозуміти внутрішню структуру інтернет-магазину та визначити логіку його роботи. Для побудови об'єктної моделі були використані діаграми класів, станів і ER-діаграма. Ці діаграми дозволять зрозуміти логіку взаємодії сутностей між об'єктами та приблизну структуру бази даних.

Діаграма станів використовується для опису поведінки об'єкта у різних ситуаціях. На рисунку 3.1.1 зображена діаграма для об'єкту «Замовлення». Вона показує як змінюється стан замовлення від моменту його створення до успішного завершення або повернення відправнику.

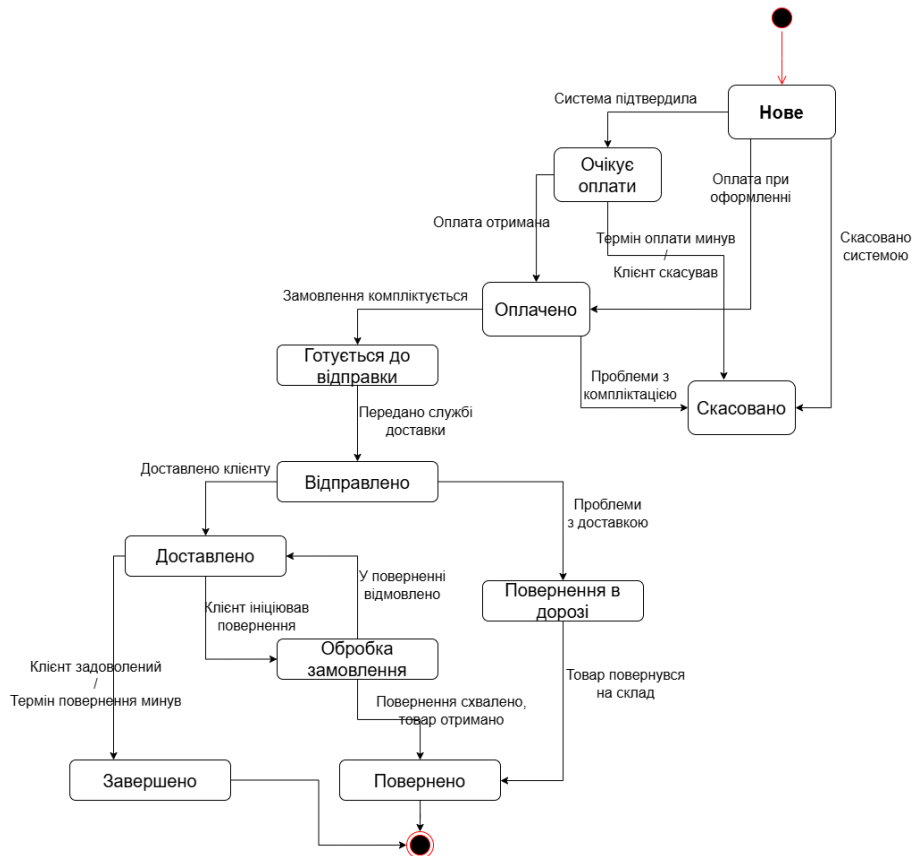


Рисунок 3.1.1 – Діаграма станів об'єкту «Замовлення».

На цій діаграмі зображені зв'язки усіх елементів системи. Наприклад, вказано, що у системі існує 3 типи користувачів: звичайний користувач, у якого є можливість зареєструватися, продивитися товари, додати їх у кошик та розмістити замовлення, зареєстрований користувач, який наслідуює можливості звичайного користувача та додатково має можливості керувати своїм профілем, переглядати історію замовлень та залишати відгуки, та адміністратор, який може виконувати усі функції зареєстрованого користувача та звичайного користувача та додатково управляти наповненням сторінок на сайті, додавати, видаляти та редагувати інформацію про товари, встановлювати знижки та керувати замовленнями. У діаграмі також вказано які поля та функції має товар, описані його зв'язки між учасниками системи та іншими об'єктами.

ER-діаграма описує логічну структуру бази даних інтернет-магазину та описує основні сутності системи, їхні атрибути та зв'язки між ними. Така діаграма дозволяє розробникам чітко зрозуміти, яку саме інформацію необхідно зберігати, як різні частини даних пов'язані між собою, та як забезпечити цілісність і несуперечливість цих даних. Використання такої діаграми дасть змогу оптимізувати запити до бази даних, зменшити дублювання даних, підвищити узгодженість збереженої інформації та дозволить масштабувати систему у майбутньому. У процесі проєктування інтернет-магазину було розроблено ER діаграму, яка зображена на рисунку 3.1.3.

3.2 Проектування інтерфейсу програмної системи

Для розробки графічного інтерфейсу використовується безкоштовний макет Figma, який підходить під усі зазначені функціональні та нефункціональні вимоги системи.

Так як графічний інтерфейс повинен бути зрозумілим та зручним у використанні, головна сторінка не повинна мати зайвих компонентів. У макеті пропонується у шапці головної сторінки розмістити пошукову стрічку для швидкого пошуку потрібного товару. На навігаційній панелі будуть розміщені посилання на інформаційні сторінки «Про нас», «Відповіді на часті питання», «Акції», «Блог», «Умови доставки» та «Зв'язатися з нами». На бічній панелі розташований перелік з усіма категоріями товарів для швидкої навігації (рисунк 3.3.1).

Головна сторінки наповнена інформацією про знижки та бонусні програми. Наприклад у центрі екрану розміщений слайдер, який ознайомлює користувача з поточними акційними пропозиціями та програмою лояльності.

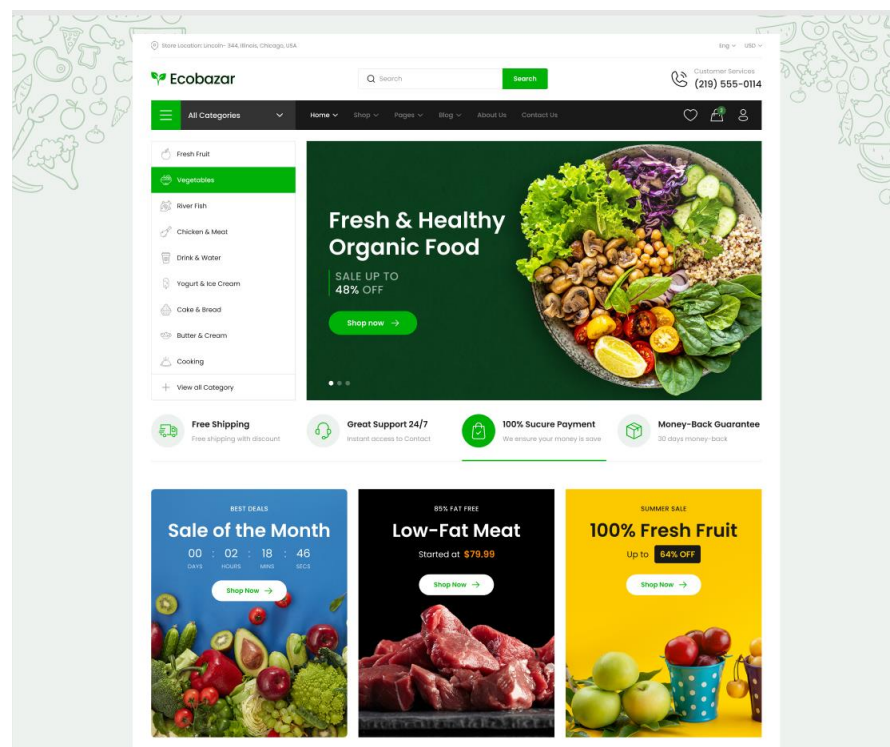


Рисунок 3.3.1 – Головна сторінка

Нижче вказаний перелік популярних категорій та популярних продуктів серед споживачів (рисунок 3.3.2). Це допоможе зацікавити потенційного клієнта придбати додаткові товари.

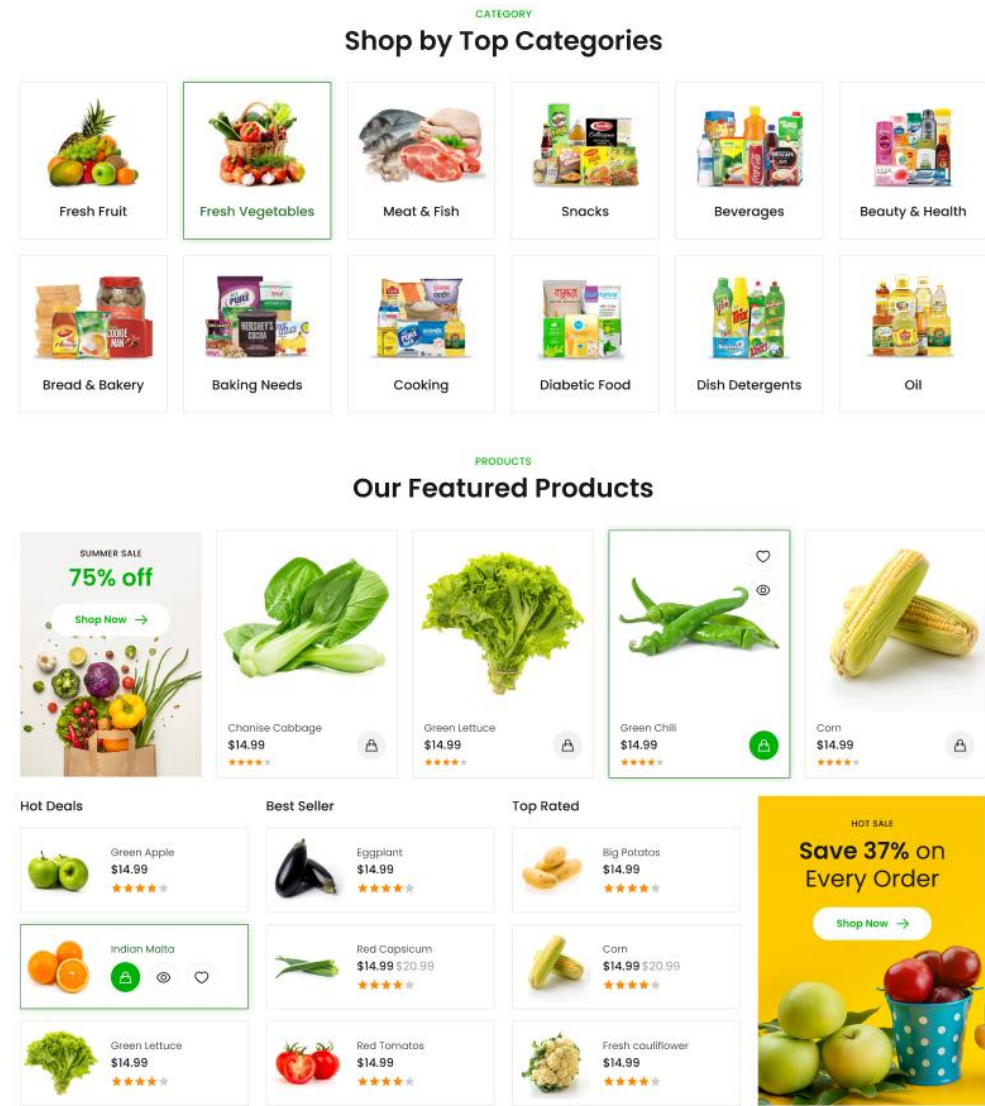


Рисунок 3.3.2 – Популярні продукти на головній сторінці

Коли користувач натискає на будь-яку категорію зі списку, він потрапляє на сторінку з товарами (рисунок 3.3.3), яка налічує усі товари з обраної категорії. Клієнт може скористуватися можливістю відфільтрувати товари за ціною, рейтингом, тегами та відсортувати від меншої до більшої ціни та навпаки. Під кожним товаром написана його назва, ціна та рейтинг.

Наводячи на товар, з'являється можливість додати його до кошику для подальшого оформлення замовлення.

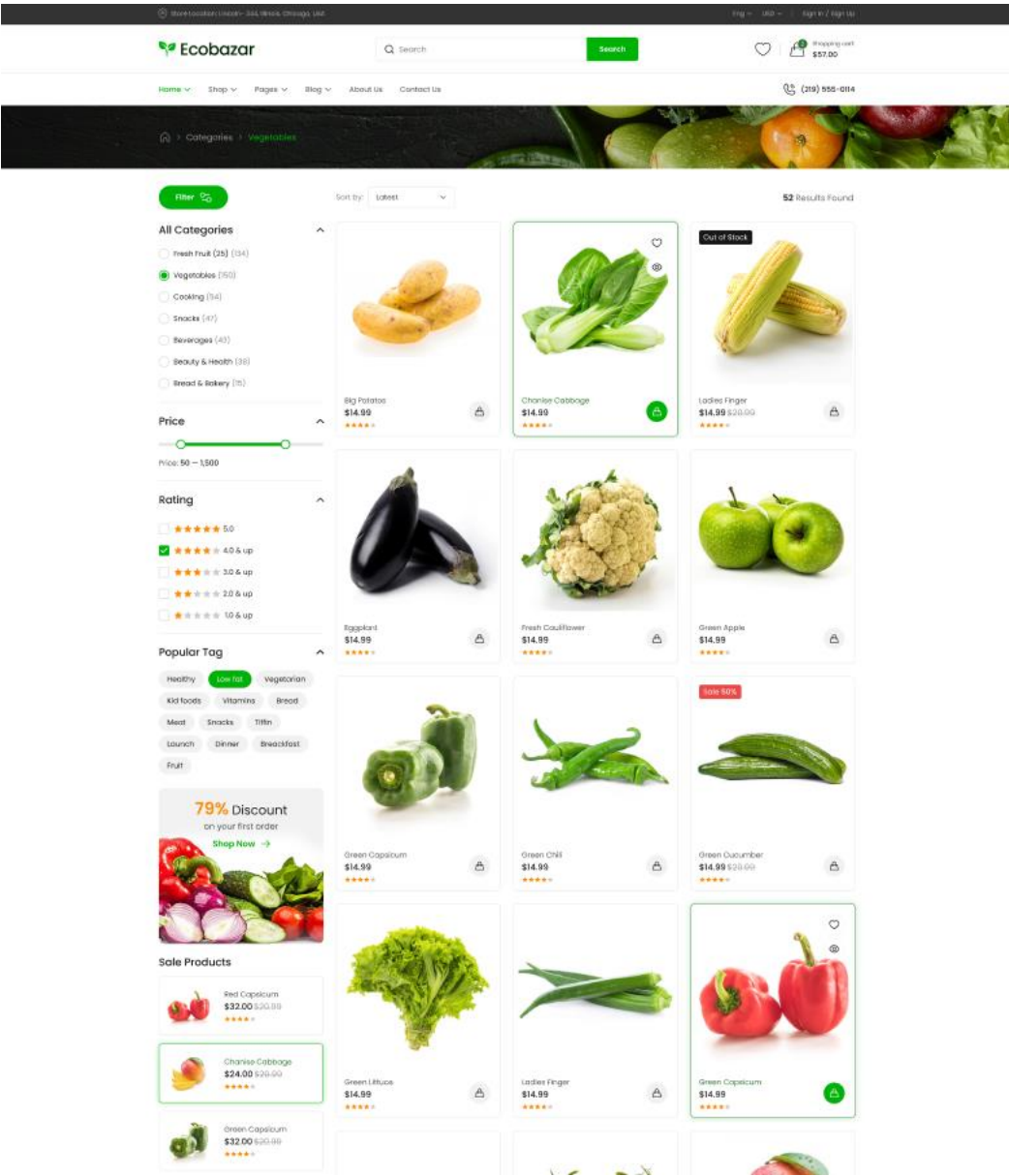


Рисунок 3.3.3 – Сторінка з товарами

Натиснувши на зображення товару, клієнт потрапляє на сторінку товару (рисунок 3.3.4). На цій сторінці споживач може додати товар до кошику, ознайомитися з відгуками та детальною інформацією про обраний товар, дізнатися термін придатності, рекомендації до споживання, харчову цінність

та хто його виробник. Під інформацією розташований блок зі схожими товарами, які можуть його зацікавити.

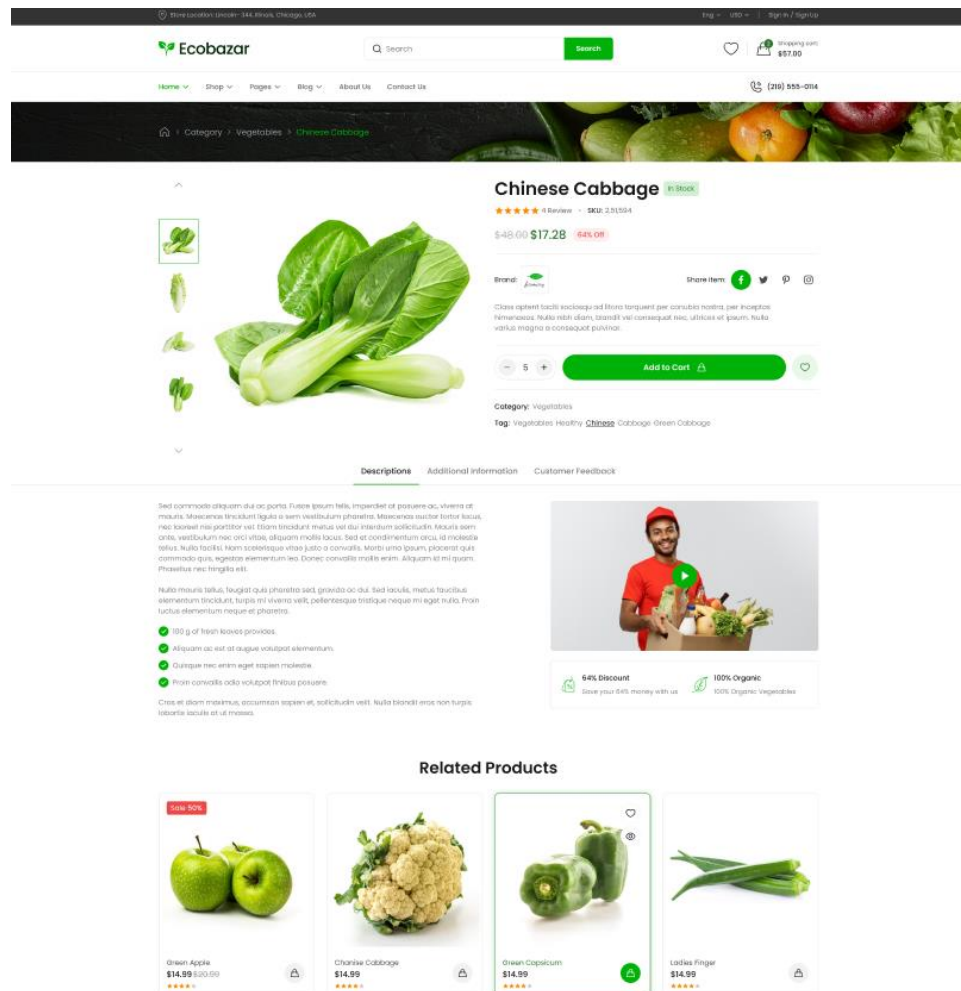


Рисунок 3.3.4 – Сторінка товару

Кошик товарів має простий та інтуїтивно зрозумілий дизайн (рисунок 3.3.5). Він налічує список обраних товарів із зображенням, назвою і ціною. Користувач може збільшити кількість товарів у кошику натиснувши на відповідні кнопки «+» та «-». Варто зауважити, що якщо товарів недостатньо на складі, то система це розпізнає та не дасть збільшити кількість.

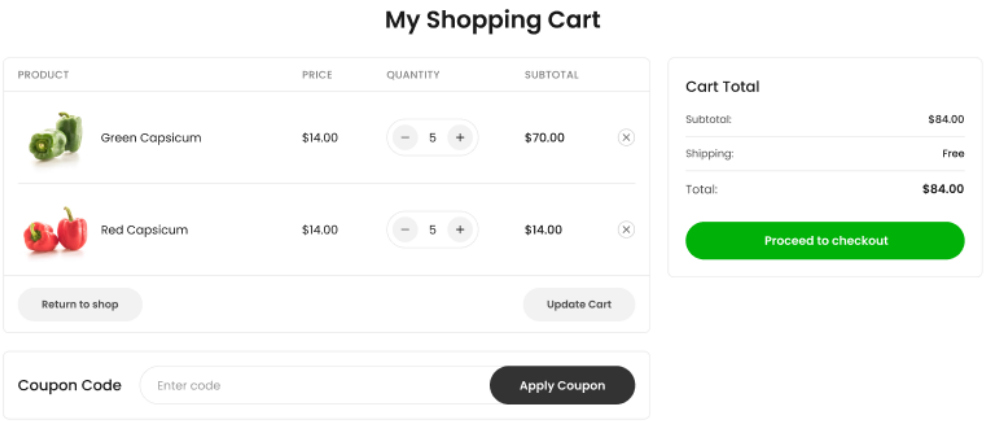


Рисунок 3.3.5 –Кошик товарів

Сторінка оформлення товару пропонує користувачу ввести особисті дані для доставки, такі як ім'я та фамілія, країна та місто, вулиця, поштовий індекс, електронна адреса та телефон. За потреби є можливість вказати побажання до замовлення в спеціальному полі.

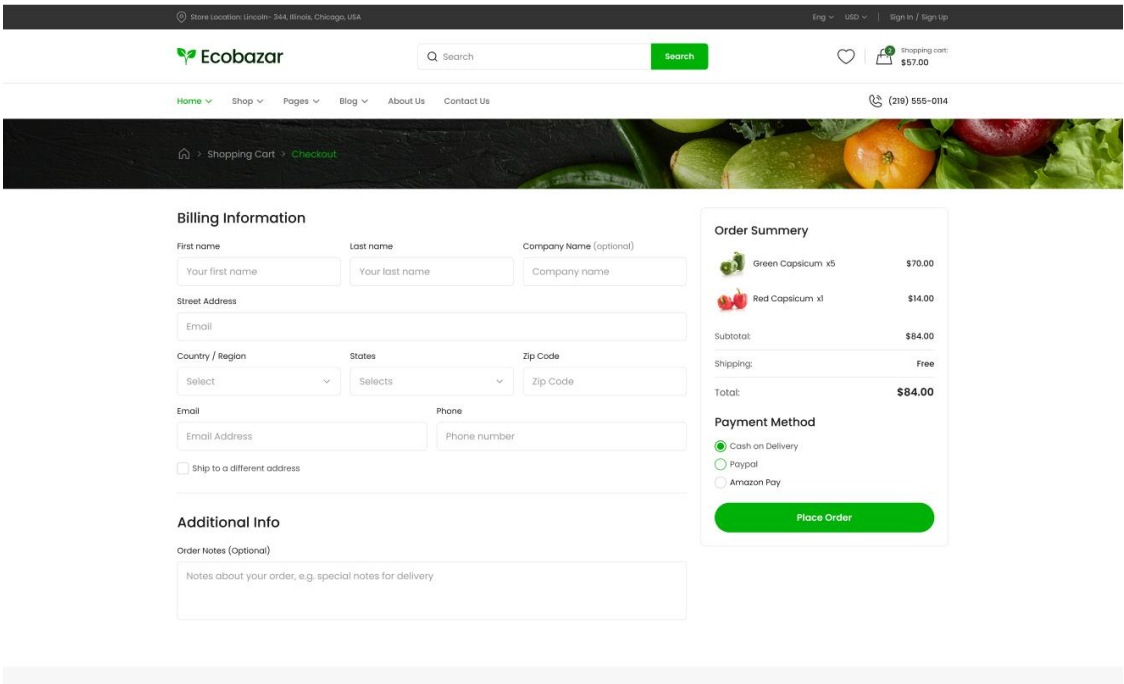


Рисунок 3.3.6 – Сторінка оформлення замовлення.

Успішно зробивши замовлення, споживач може перейти до особистого кабінету, натиснувши на іконку людини у шапці сайту. В особистому кабінеті

є можливість ознайомитися із історією замовлень та змінити контактну інформацію та адресу доставки. Дизайн сторінки особистого кабінету зображений на рисунку 3.3.7.

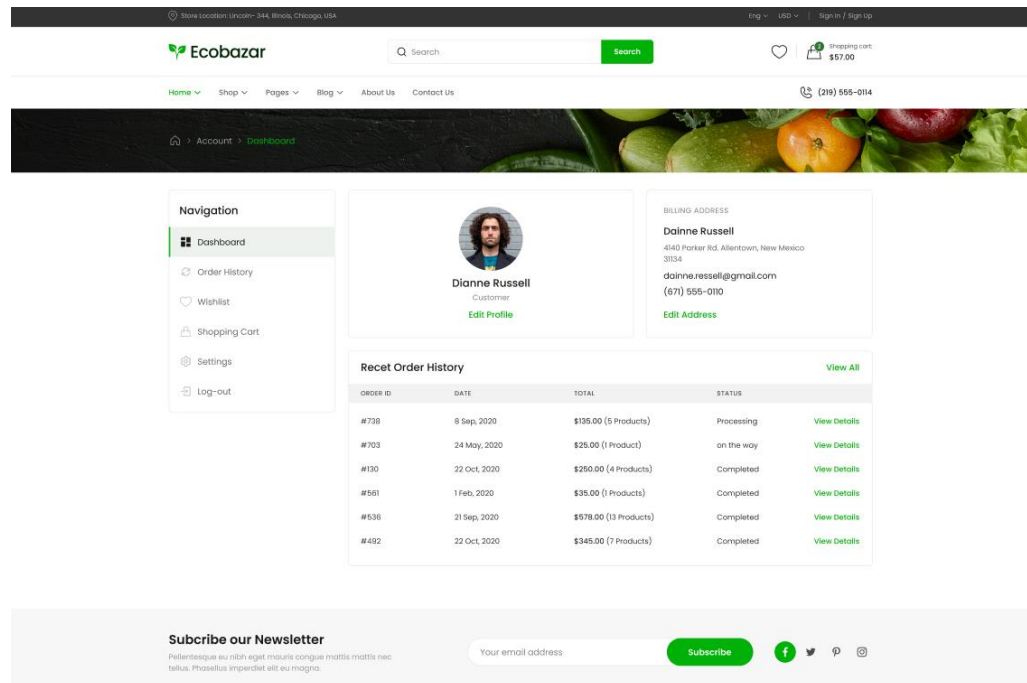


Рисунок 3.3.7 – Сторінка управління особистим кабінетом

Кольорова палітра сайту створює асоціації з натуральністю, свіжістю, чистотою та довірою. Палітра побудована на принципах мінімалізму та акцентного використання кольору. Білий фон дозволяє акцентувати увагу споживача на яскравих зображеннях товарів та підвищує читабельність шрифтів. Зелений колір використовується для виділення ключових інтерактивних елементів сторінки та напряму асоціюється із свіжою органічною продукцією.

3.3 Обґрунтування вибору мови програмування

У цьому розділі буде детально розглянуто обрані мови програмування, фреймворки та додаткові технології, які були використані для створення інтернет-магазину з продажу органічної їжі.

На сьогоднішній день розробка додатків майже не обходиться без використання фреймворків. Фреймворком називають набір бібліотек, інструментів та методологій, які спрощують процес розробки програмного продукту [22]. Він дозволяє швидше створювати додатки, уникати типових помилок та збільшує його стабільність. Для розробки веб-додатку було обрано фреймворк Django мови програмування Python для серверної частини, фреймворк Vue.js мови JavaScript для клієнтської частини і в якості бази даних була обрана СУБД MySQL. Серверна частина виконує бізнес-логіку додатку, взаємодіє з базою даних, управляє даними та надає API для клієнтської частини. Клієнтська частина відповідає за візуальну складову додатку, взаємодію з клієнтом та відправляє запити на сервер. СУБД використовується для зберігання, організації та управління всіма даними інтернет-магазину.

JavaScript був розроблений ще у середині 90-х років з метою зробити сторінки інтерактивними та «живими» і з того часу він продовжив розвиватися по теперішній час [3]. Наразі JavaScript вважається стандартом фронтенд розробки та підтримується усіма сучасними браузерами, саме тому вибір безальтернативно падає на нього.

На сьогоднішній день JavaScript налічує декілька популярних фреймворків для розробки веб-додатків, серед яких React, Angular та Vue.js.

React – це найпопулярніша JavaScript бібліотека для побудови користувацьких інтерфейсів, яка підтримується компанією Facebook. Вона має велику спільноту та широку екосистему зі сторонніми бібліотеками та інструментами. Завдяки своїй популярності, багато великих компаній активно застосовують цю бібліотеку для розробки власних продуктів. Але є і недоліки: React складніший для освоєння ніж Vue, ядро фреймворку надає лише базовий функціонал і для повноцінної реалізації деяких функцій потребує підключення сторонніх бібліотек [2].

Angular – це повноцінний фронтенд фреймворк, розроблений і підтримуваний Google для створення односторінкових застосунків, що надає можливість писати чистий та читабельний код за допомогою строго-

типізованої мови TypeScript [21]. Проте, через сурову типізацію, Angular має більш круту криву навчання [21]. Ще одним його недоліком є обмеженість у сферах його застосування, так як він здатен використовувати надмірний рівень ресурсів, уповільнюючи швидкість завантаження сторінок та довше обробляти запити, тому він зазвичай використовується у великих проєктах [8].

Vue.js – це прогресивний фреймворк для розробки односторінкових додатків, який дозволяє будувати користувацькі інтерфейси за допомогою компонентів та двостороннього зв'язування даних [20]. Низький поріг входу, легкість у вивченні та його гнучкість робить цей фреймворк популярним вибором для невеликих проєктів [21]. Великими перевагами Vue є те, що він ефективно застосовує можливості деяких інших успішних фреймворків та велика кількість потрібного функціоналу у ньому вже йде «з коробки» [8,2]. Недоліки у контексті даного проєкту у Vue є незначними: порівняно з іншими фреймворками він має меншу кількість готових UI-компонентів, його надмірна гнучкість може призвести до різноманітності коду, якщо команда програмістів у майбутньому буде розширюватись.

Згідно проведеному порівнянню фреймворків для клієнтської частини додатку, можна дійти висновку що Vue.js є оптимальним вибором завдяки своєму низькому порогу входу, легкості у вивченні та гнучкості розробки.

Крім того, Vue.js налічує такі додаткові бібліотеки як Vuex, Vue Router та Axios. Vuex використовується для керування станом додатку. Всі важливі дані (стан кошика, дані користувача, фільтри товарів) зберігаються у Vuex store, що забезпечує передбачуваність та легкість відстеження змін. Vue Router допомагає створити навігацію між сторінками без їх перезавантаження, а бібліотека Axios використовується для виконання HTTP-запитів до бекенду, що полегшує взаємодію з RESTful API.

Для серверної частини розглядалися такі мови програмування як Python, Node.js та PHP.

Python відомий своїм простим синтаксисом та високою читабельністю коду, великою вбудованою бібліотекою та величезною кількістю сторонніх

бібліотек від спільноти [3]. Перевагами цієї мови є наявність потужних фреймворків, таких як Django та Flask, легкість інтеграції з базами даних та великий потенціал для розширення функціоналу сайту. Django забезпечує швидку розробку додатків завдяки можливості швидко створювати моделі даних та дуже зручними інструментам для роботи з базою даних [19]. Із недоліків можна виділити загальну швидкість виконання великих за обсягом завдань, бо Python є інтерпретованою мовою, через що код виконується рядок за рядком за допомогою інтерпретатора, на відміну від компільованої мови.

Node.js – це середовище виконання JavaScript на стороні сервера за допомоги движка V8 від Google. Тобто Node.js допомагає запускати JavaScript код на стороні серверу. Node.js має швидкий та мінімалістичний фреймворк Express.js для створення веб-додатків та API. Перевагами використання Node.js є однакова мова програмування як для фронтенду так і для бекенду та висока продуктивність I/O операцій, що може допомогти з великою кількістю одночасних підключень. Із недоліків можна відмітити складнощі з реалізації асинхронних шматків коду, меншим вбудованим функціоналом Express.js та однопоточність.

PHP – це популярна серверна скриптова мова програмування, яка широко використовується для веб-розробки. Налічує такі фреймворки як Laravel та Symfony. Перевагами є величезна спільнота, потужні фреймворки, які надають багато інструментів для швидкої розробки та низький поріг входження. Із недоліків можна відмітити відсутність суворої типізації, через що можуть виникнути помилки та неузгодженість і проблеми з масштабованістю та підтримкою коду для великих проєктів [3].

Порівнявши серверні мови програмування було прийнято рішення, що Python з фреймворком Django виглядає найбільш збалансованим та доцільним вибором через легку та зрозумілу мову програмування, велику спільноту, велику кількість вбудованого функціоналу, можливості швидко розробляти додаток та можливості Django ефективно спілкуватися з базою даних.

База даних є важливим складовим у розробці інтернет-магазину, оскільки вона відповідає за надійне зберігання, організацію та швидкий доступ до всієї важливої інформації. Для порівняння було обрано такі СУБД як MySQL, PostgreSQL та MongoDB.

MySQL – це система управління реляційними базами даних, яка заснована на мові SQL. Вона використовується у веб-розробці та виконує функції зберігання, управління та видалення даних. Основними перевагами цієї бази даних є легкість використання, структурованість даних та наявність ACID транзакцій, які є критично важливими для операцій, пов'язаних з замовленнями та фінансами. Недоліком є складність у горизонтальному масштабуванні, однак для невеликого інтернет-магазину, це не буде великою проблемою.

PostgreSQL – це ще одна потужна об'єктно-реляційна СУБД з відкритим вихідним кодом, яка часто розглядається як альтернатива MySQL. Вона широко використовується у веб-розробці, має високу продуктивність, багато можливостей для розширення функціональності та дозволяє користувачам визначати власні типи даних, функції та оператори [11]. Недоліком є складність в адмініструванні та налаштуванні порівняно з MySQL.

MongoDB – це популярна гнучка NoSQL база даних, що зберігає дані у вигляді JSON-подібних документів. Пропонує можливості для зберігання та обробки даних великих обсягів, а також дозволяє легко змінювати структуру документів [11]. Така база даних не гарантує цілісність даних через відсутність ACID транзакцій, і відповідальність за підтримку зв'язків часто лягає на рівень додатку, що може призвести до дублювання даних або їх неузгодженості.

Після аналізу різних типів СУБД, для розробки інтернет-магазину було обрано СУБД MySQL. Не зважаючи на переваги PostgreSQL у деяких аспектах та гнучкість MongoDB, для цього проєкту важливими факторами стали чіткі зв'язки між товарами, категоріями, користувачами та замовленнями, які гарантує реляційна модель. ACID-транзакції є важливими для обробки замовлень та фінансових операцій. В цілому, можливостей MySQL буде

цілком достатньо для реалізації всіх запланованих функцій для розробки інтернет-магазину.

Після вибору основних мов програмування та фреймворків для розробки, наступним кроком необхідно обрати зручну IDE (інтегроване середовище розробки) для роботи. Розробка фронтенд частини зазвичай здійснюється у таких середовищах як: Microsoft Visual Studio Code, Sublime Text, IntelliJ IDEA, Brackets та багато інших.

У контексті розробки цього проєкту було обрано Visual Studio Code, бо він дозволяє розробляти на десятках мов програмування, включаючи JavaScript з фреймворком Vue.js та Python. Він також має безліч аддонів які полегшують та прискорюють розробку, дозволяє додавати власні сніппети коду, має систему контролю версій Git, та багато інших налаштувань для зручної розробки.

Вибір цих засобів розробки засновується не лише на теоретичних перевагах, але й на успішному застосуванні у реальних проєктах. Наприклад, такі сайти як Alibaba, Xiaomi та Adobe частково або повністю інтегрують Vue.js у свій стек технологій. Django у свою чергу впроваджений у таких сервісах як Instagram, Spotify та YouTube. MySQL використовується у Facebook, Sony та WePay.

3.4 Опис програмної реалізації

Розроблений інтернет-магазин створено за класичною клієнт-серверною архітектурою. Клієнтська частина представляє із себе односторінковий додаток, розроблений з використанням фреймворку Vue.js, який відповідає за відображення інтерфейсу, взаємодію з сервером та динамічно оновлює дані на сторінці. Серверна частина реалізована за допомогою фреймворку Django, який відповідає за обробку бізнес-логіки, взаємодію з базою даних, управління користувачами, обробку замовлень та надання API для клієнтської частини. Взаємодія між клієнтською та серверною частинами відбувається за

допомогою RESTful API. На клієнтській частині відправляються HTTP запити до певних URL-адрес на сервері, а сервер повертає відповіді у форматі JSON об'єкту. База даних MySQL використовується для постійного зберігання всієї інформації о продуктах, категоріях, користувачах, замовленнях, тощо.

Розглянемо детальніше клієнтську частину додатку. Vue.js пропонує компонентний підхід до розробки, що дозволяє розробити певний компонент один раз та мати можливість використовувати його повторно безліч разів. Кожен візуальний та функціональний блок інтерфейсу реалізовано як окремий компонент, який містить три секції:

`<template>` - це HTML структура компонента, яка окрім звичайного HTML може відображати змінні, які за рахунок реактивності та двостороннього зв'язування динамічно оновлюють інформацію на сторінці. Також `template` дозволяє використовувати у собі спеціальні директиви, такі як `v-if`, `v-for`, `v-bind` та `v-on`. `V-if` призначена для умовного рендерингу DOM елементів, яка дозволяє повністю видаляти або додавати окремі елементи, якщо виконується певна умова. `V-for` дозволяє проходитися по масиву та створювати повторювані елементи. В цілому це аналог циклів, як у інших мовах програмування. `V-bind` відповідає за динамічне зв'язування HTML атрибутів з властивостями змінної чи результатом виконання функції у компоненті. Директива `v-on` відповідає за обробку подій, таких як кліки, наведення курсору, введення тексту тощо [20].

`<script>` - це JavaScript код компонента, який містить усю його логіку, та включає у себе такі елементи як `data`, `methods`, `computed`, `props` та `lifecycle hooks` [20]. Елемент `data` дозволяє зберігати реактивні дані компонента, які можна використовувати всередині цього ж компонента. `Methods` містить звичайні JavaScript, які можуть бути використані безпосередньо у компоненті. `Computed` містить функції, які повертають обчислювані значення. Ці значення переобчислюються лише тоді, коли змінюються їх реактивні залежності. У кожного компонента є свій життєвий цикл, під час якого викликаються певні

функції (created, mounted, updated, unmounted). Це дозволяє виконувати потрібний код у певний момент життєвого циклу компоненту.

`<style>` потрібен для визначення CSS правил, які визначають зовнішній вигляд компонента. Він дозволяє інкапсулювати стилі, щоб вони не впливали на інші компоненти або глобальні стилі, за допомогою атрибуту `scoped`. Цей блок також підтримує використання препроцесору SCSS, який значно полегшує написання CSS стилів через наявність можливості оголошувати власні змінні, які можуть використовуватися для зберігання кольорів, відступів, шрифтів тощо. SCSS препроцесор дозволяє вкладати CSS селектори один в одного, відтворюючи структуру HTML документа для більшої читабельності коду та його логічного структурування.

У ході розробки інтернет-магазину було створено велику кількість окремих компонентів, зокрема для шапки сайту (HeaderItem, HeaderLogo, HeaderSearchBar), для відстеження вводу від користувача (Button, CircleButton, CheckboxOptions, RadioOptions, Slider та Tags), та інші. Для виведення важливої інформації були розроблені такі компоненти як: DropDownMenu, InfoCard, CategoryCard, ListItem, ProductCard (Додаток А), тощо.

Усі компоненти не існують по-одиночці, а збираються у віджети, які у свою чергу відображаються на сторінці. Тобто віджети зберігають та структурують у собі один та більше компонентів для реалізації функціонально цілісної та логічно завершеної частини користувацького інтерфейсу. Серед віджетів були розроблені CartPopup, Footer, HeaderBottom (Додаток Б), HeaderMiddle, HeaderTop, ImageSlider та інші.

Для централізованого управління спільним станом додатку використовується бібліотека Vuex. Вона допомагає створити єдине «джерело правди» – глобальне сховище, в якому зберігається весь стан додатку, наприклад вміст кошика, інформація про користувача, список товарів, активні фільтри, тощо.

Vuex сховище складається зі стану (State), геттерів (Getters), мутацій (Mutations), дій (Actions) та модулів (Modules).

State – об’єкт в якому зберігаються дані додатку для певного користувача.

Getters – функції для отримання актуальної інформації зі State, дозволяють фільтрувати або трансформувати дані з State перед їх використанням. Getters використовуються наступним чином, на прикладі отримання списку продуктів із стану: `getProducts: (state) => state.products`

Mutations – це синхронні функції, які є єдиним способом змінити State. Призначені для передбачуваної зміни стану і викликаються за допомогою `store.commit('mutationName', payload)`. Використання мутацій на прикладі зміни стану `isMobile`: `setMobile(state, value) {state.isMobile = value}`, де «`state.isMobile = value`» – зміна стану змінної `isMobile` на передане у `value` значення.

Actions – це функції, які можуть містити асинхронні операції. Зазвичай Actions використовується для створення запитів до бази даних або серверу. Такі функції не змінюють стан напямую, а делегують цю задачу мутаціям. Викликаються через `store.dispatch('actionName', payload)`.

Modules дозволяють розбити сховище на окремі модулі для кращої організації, кожен зі своїм state, getters, mutations та actions.

Для використання Vuex сховища у Vue компонентах необхідно імпортувати потрібні елементи за допомогою `import { mapGetters, mapMutations, mapActions } from 'vuex'`. Це дозволить у methods оголосити `...mapMutations({setFilters: "userData/setFilters"})`, та `...mapActions({fetchProducts: 'products/fetchProducts'})`, що дозволить використовувати мутацію `setFilters` та асинхронну функцію `fetchProducts` у самому компоненті. Таким же самим чином у Computed можна оголосити `mapGetters`.

Для зберігання інформації про продукти на клієнтській частині застосунку було створено Vuex сховище `products.js`. Воно описує основні дії з продуктами: отримати продукти з серверу, отримати категорії, сортувати продукти, тощо (Додаток В).

Для навігації по сторінкам сайту було використано можливості Vue Router, який дозволяє динамічно змінювати контент на сторінці без необхідності її перезавантажувати. Для цього у головному файлі Vue Router в масив `routes` вказуються шляхи до сторінок з відповідним Vue компонентом. Для перенаправлення користувача на обрану сторінку, використовується `<router-link to="назваСторінки">` або у методах компоненту `this.$router.push('/назваСторінки')`.

У процесі розробки графічного інтерфейсу було використано принцип верстання «Desktop-first» через відсутність мобільної версії макету. Проте це не завадило зробити інтуїтивно зрозумілий адаптивний дизайн для усіх розмірів екрану. Ключовими методами розробки адаптивного інтерфейсу стало використання `grid` та `flex` відображення та використання математичної функції `clamp()` у CSS для забезпечення еластичності окремих елементів, для визначення їх мінімальних та максимальних розмірів. Медіа запити дозволили налаштувати деякі частини веб-сайту під конкретні розміри пристроїв. Наприклад, медіа запити використовуються для приховування деяких елементів у шапці сайту, що забезпечило її коректне відображення на мобільних девайсах.

Серверна частини веб-додатку реалізована за допомогою можливостей Django на мові програмування Python. Фреймворк Django значно спростив розробку API, запитів до бази даних та цілком створення серверу за рахунок наявності у собі усіх необхідних компонентів для зручної розробки.

Вся основна функціональність серверної частини була реалізована в рамках єдиного Django-додатку під назвою `shop`. Головний проект Django, названий `online-shop-server`, містить файли конфігурації `settings.py` та `urls.py`. Ці файли містять усю конфігурацію застосунку, включаючи параметри підключення до бази даних, його компоненти та інші налаштування [19].

Взаємодію з базою даних MySQL допомагає здійснювати Django ORM (Object-Relational Mapper), який дозволяє створювати, отримувати, оновлювати та видаляти об'єкти з бази даних за допомогою Python класів [19].

Спроектовані на етапі аналізу класи та сутності були реалізовані як Python класи у файлі `models.py` (Додаток Г). Було створено такі ключові моделі, як:

- `Category` з категоріями товарів;
- `Product`, що містить назву, опис, ціну, бренд, відгуки та інші поля товару;
- `Order`, який зберігає інформацію про замовлення, включаючи інформацію про замовника, його контактну інформацію, методи оплати та доставки та іншу інформацію;
- `Review` описує вигляд структури відгуків про товари;
- `OrderItem` пов'язує замовлення з конкретними товарами, їх кількістю та ціною на момент покупки;
- `PaymentMethod` описує структуру таблиці з інформацією про методи оплати.

Використання полів типу `ForeignKey` забезпечує реляційну цілісність даних, як наприклад у `Product` для зв'язку з `Category`. Django дозволяє автоматично синхронізувати внесені зміни у класах з самою базою даних MySQL за допомогою міграцій.

Бізнес-логіка інтернет-магазину реалізована у представленнях у файлі `shop/views.py`. Ці представлення оброблюють HTTP запити від клієнтської частини, взаємодіють з моделями даних та формують відповідь у форматі JSON. Для реалізації цих представлень використовуються класові представлення, такі як `ModelViewSet` для комплексного управління сутностями `ProductViewSet`, `CategoryViewSet`, `ReviewViewSet` та `OrderCreateSet`. `ReadOnlyModelViewSet` використовується для надання доступу тільки для читання, як наприклад у `PaymentMethodViewSet` та `UserOrderViewSet`.

Кожен `ViewSet` пов'язаний з відповідною моделлю даних (`Product`, `Category`, `Order`) та серіалізатором (`ProductSerializer`, `OrderSerializer`), який відповідає за перетворення даних між об'єктами Django та форматом JSON.

Управління доступом до різних операцій реалізовано за допомогою системи дозволів DRF. `UserAddressListCreateView` та `UserOrderViewSet` вимагають автентифікації користувача, тоді як `OrderCreateSet` дозволяє створювати замовлення усім користувачам, при цьому автоматично прив'язуючи замовлення до автентифікованого користувача, якщо такий є.

Файл `shop/urls.py` визначає основні ендпоінти, які пов'язують URL адреси з відповідними функціями або класами представлень у `views.py` [19]. Для організації маршрутів до REST API ендпоінтів використовується `DefaultRouter` з бібліотеки Django REST Framework. Цей маршрутизатор автоматично генерує набір URL-шляхів для стандартних CRUD-операцій на основі зареєстрованих `ViewSet`-ів.

Окрім маршрутів згенерованих `DefaultRouter`, деякі шляхи були створені вручну за допомогою функції `path()`. Наприклад, при зверненні клієнта за посиланням `users/me/orders/`, система автоматично спрямовує запит на обробник `UserOrderViewSet`, що розміщений у моделі `views`. Виглядає це наступним чином: `path('users/me/orders/', UserOrderViewSet.as_view({'get': 'list'}), name='user-orders-list')`,

Для зручного управління даними інтернет-магазину було використано адміністративну панель Django (Додаток Д). Вона дозволяє керувати інформацією у базі даних без необхідності прямого втручання в неї. У файлі `shop/admin.py` були зареєстровані усі основні моделі для відображення та управління в адмін-панелі.

Таким чином було успішно поєднано клієнтську частину додатку, розроблену на `Vue.js` та серверну, яка розроблена на Django.

РОЗДІЛ 4

ДОСЛІДНА ЕКСПЛУАТАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Інструкція користувача ІС

1. Загальний огляд інтерфейсу та навігація.

На головній сторінці розташований слайдер зображень з акційними пропозиціями, натиснувши на який можна перейти до необхідного товару чи інформаційної сторінки. Нижче на ній розміщені популярні категорії товарів, популярні продукти та інформація про команду сайту з посиланнями на сторінки в соціальних мережах. У шапці сайту розташовано логотип, головне меню з навігацією по сайту, з можливістю перейти до сторінки категорій, та на інформаційні сторінки «Про нас», «Акції», «Блог», «Часті запитання», «Умови доставки» та «Зв'язатися з нами». У верхній частині сайту є можливість увійти до облікового запису, знайти товари через рядок пошуку та перейти до кошику.

2. Реєстрація та вхід

Для створення облікового запису необхідно натиснути на іконку людини у шапці сайту. Після цього з'явиться сторінка з можливістю авторизуватися чи зареєструватися. Для реєстрації необхідно ввести адресу електронної пошти, та двічі підтвердити свій пароль. Якщо у користувача вже є обліковий запис достатньо просто ввести адресу електронної пошти та пароль у вікно авторизації. Після цієї дії користувача буде перенаправлено на сторінку особистого кабінету.

3. Пошук та вибір товарів.

Пошук товарів може здійснюватися декількома методами. Користувач може шукати товари через пошукову стрічку у шапці сайту, обирати товари на головній сторінці з запропонованих варіантів, або переглядати каталог товарів. Пошук дозволяє швидко знаходити потрібний товар за ключовим

словом. Каталог товарів дає можливість ефективно шукати необхідні товари використовуючи функцію фільтрації за категоріями, ціною, рейтингом та тегами.

Натиснувши на товар, користувач потрапляє на сторінку цього товару, на якій можна ознайомитись з фотографіями, детальним описом, харчовою цінністю, наявністю та відгуками від інших покупців.

4. Робота з кошиком.

Додати товари до кошику також можна декількома способами. Для цього достатньо у каталозі на потрібному товарі натиснути на іконку кошику, або на сторінці товару натиснути кнопку «Додати у кошик». Функціонал кошику дозволяє передивитися додані у нього товари, збільшити або зменшити кількість на обраних позиціях та дізнатися загальну ціну за усі товари.

5. Оформлення замовлення.

Для оформлення замовлення необхідно на сторінці кошику натиснути на кнопку «Перейти до оформлення». Ця кнопка перенаправить користувача на сторінку оформлення, де потрібно заповнити контактну інформацію, обрати зручний спосіб доставки, вибрати спосіб оплати та за необхідністю залишити побажання до замовлення у спеціальному полі. Якщо замовлення було успішно створене, система сповістить про це користувача.

6. Особистий кабінет.

Після входу у систему користувач отримує доступ до особистого кабінету. Для того щоб потрапити у нього, достатньо у шапці сайту натиснути на іконку людини. У ньому є можливість продивитися свою історію замовлень, змінити особисті дані та адреси доставки.

7. Відгук про товар

Під кожним товаром у зареєстрованих користувачів є можливість залишити відгук. Для цього потрібно лише зайти на сторінку потрібного товару, перейти у підрозділ «Відгуки клієнтів» та натиснути кнопку «Залишити відгук».

4.2 Основи тестування програмного забезпечення

Тестування є завершальним етапом розробки веб-додатку. Метою є перевірка реалізованого функціоналу поставленим вимогам а також виявлення та усунення потенційних помилок.

У ході тестування було проведено перевірку роботи основних функцій інтернет-магазину, працездатність клієнтської та адміністративної частини додатку, збереження інформації до бази даних та коректне відображення усіх сторінок веб-додатку. Опис тестових ситуацій та результат тестування наведений у таблиці 4.2.1.

Особливу увагу приділено тестуванню основних бізнес-процесів: реєстрації користувача, авторизації, перегляду каталогу, додавання товарів до кошика, оформлення замовлення, перегляду історії замовлень та роботи адміністративної панелі. Також було виконано перевірку адаптивності графічного інтерфейсу та часу завантаження сторінок.

Таблиця 4.2.1 – Тестові ситуації

№	Очікуваний результат	Статус
1	2	3
1	Реєстрація нового користувача створює новий профіль	Пройдено
2	Авторизація користувача дає доступ до особистого кабінету	Пройдено
3	Каталог товарів коректно відображає усі товари	Пройдено
4	Пошук товарів за ключовими словами видає необхідні результати	Пройдено
5	Товари успішно фільтруються за вказаними параметрами	Пройдено
6	Товари успішно додаються у кошик	Пройдено
7	Кількість товарів у кошику успішно змінюється	Пройдено

Продовження табл. 4.2.1.

1	2	3
8	Товари успішно видаляються з кошика при натисканні на відповідну кнопку	Пройдено
9	Оформлення замовлення відбувається коректно і підтверджується	Пройдено
10	Збереження інформації про замовлення в особистому кабінеті	Пройдено
11	Можливість залишити відгук до товару	Пройдено
12	Вхід до облікового запису адміністратора здійснюється успішно	Пройдено
13	Адміністративна панель дозволяє додавати і редагувати товари	Пройдено
14	Адміністративна панель дозволяє змінювати статус замовлення	Пройдено
15	Інтерфейс адаптовано під мобільні пристрої	Пройдено
16	Час завантаження сторінок < 2 секунд	Пройдено

В результаті тестування усі тестові ситуації були успішно пройдені. Ключові сценарії взаємодії користувача із системою, такі як реєстрація, пошук товарів, робота з кошиком та оформлення замовлення успішно функціонують. Створений веб-додаток має зрозумілий і зручний інтерфейс, сторінки швидко завантажуються, інтерфейс адаптований під розміри усіх пристроїв. Таким чином, можна зробити висновок, що інтернет-магазин готовий до використання та демонстрації.

4.3 Техніко-економічні показники ІС

Витрати на оплату праці з нарахуваннями – складаються з витрат на основну заробітну плату, додаткову заробітну плату та нарахувань згідно чинного законодавства.

Витрати на основну заробітну плату – визначаються виходячи з місячного посадового окладу або вартості 1 години роботи розробника та витрат часу, понесених на розробку даного продукту. Розрахунок здійснюється за формулою:

$$ЗПосн = \frac{ЗПміс}{ФРЧ} * Тр = \frac{10000}{176} * 240 = 13636 \text{ грн} \quad (4.3.1)$$

де ЗПосн – основна заробітна плата, грн

ЗПміс – місячний посадовий оклад розробника, грн/міс

ФРЧ – місячний фонд робочого часу, год/міс

Тр – трудомісткість (затрати часу) виготовлення програмного продукту, год.

Місячний фонд робочого часу (ФРЧ) розробника визначається виходячи з кількості робочих днів на місяць та тривалості робочого дня:

$$ФРЧ = Крдм * Трз = 22 * 8 = 176 \text{ год/міс} \quad (4.3.2)$$

де Крд – кількість робочих днів на місяць, дн.

Трз – тривалість робочого дня (зміни), год.

У випадку, якщо при розробці продукту задіяні декілька працівників, розмір основної заробітної плати розраховується за кожним а дані зводяться до таблиці 4.3.1.

Таблиця 4.3.1 – Розрахунок основної заробітної плати розробників

Найменування посади	Місячний посадовий оклад, грн.	Трудовісткість участі у виготовленні продукту, год.	Витрати на основну заробітну плату, грн.
Програміст	10000	240	13636
Всього	10000	240	13636

Додаткова заробітна плата – передбачає різноманітні види доплат за рівень кваліфікації, стаж роботи тощо. Розмір нарахувань здійснюється відповідно до норм прийнятих на підприємстві-замовника.

Сума додаткової заробітної плати розраховується від розміру основної:

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} * \frac{\%_{\text{дод}}}{100} = 13636 * \frac{22}{100} = 2999 \text{ грн} \quad (4.3.3)$$

де $\%_{\text{дод}}$ – відсоток нарахувань додаткової заробітної плати, %

Нарахування на заробітну плату здійснюють від суми основної та додаткової зарплати. Розмір нарахувань становить 22%.

$$Нзп = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) * \frac{\%_{\text{нар}}}{100} = (13636 + 2999) * \frac{22}{100} = 3659 \text{ грн} \quad (4.3.4)$$

Де $\%_{\text{нар}}$ – відсоток нарахувань заробітної плати, %

Розрахунок загальної суми витрат на оплату праці з нарахуваннями наведемо у вигляді таблиці 4.3.2.

Таблиця 4.3.2 – Розрахунок витрат на оплату праці з нарахуваннями розробників програмного продукту

Найменування посади	Основна заробітна плата, грн.	Додаткова заробітна плата, год.	Розмір нарахувань на оплату праці, грн.	Всього витрат на оплату праці з нарахуваннями, грн.
Програміст	13636	2999	3659	20294
Всього	13636	2999	3659	20294

Розрахунок суми амортизаційних відрахувань обладнання, яке використовується при розробці продукту

Нарахування суми амортизаційних відрахувань здійснюється найбільш поширеним прямолінійним методом.

Відповідно до податкового кодексу України електронно-обчислювальні машини (ПК) відносять до 4 групи, з терміном корисного використання 2 роки. Відповідно річна норма амортизації становить 50%.

Балансова вартість ПК (або ноутбуку) на момент придбання (розрахунку) становить 8000 грн.

Розрахунок річної суми амортизаційних відрахувань:

$$A_{річ} = B_{Вобл} * \frac{\%_{аморт}}{100} = 8000 * \frac{50}{100} = 4000 \text{ грн} \quad (4.3.5)$$

де $B_{Вобл}$. – ціна придбання, або балансова вартість комп'ютера на момент придбання, грн.

$\%_{аморт}$ – річний відсоток нарахувань додаткової заробітної плати, %

Суму амортизаційних відрахувань, що увійде до собівартості програмного продукту розраховується з врахуванням часу використання обладнання при розробці продукту.

$$A = 4500 * \frac{30}{365} = 369,8 \text{ грн} \quad (4.3.6)$$

Де Твик – тривалість використання комп'ютера для розробки продукту.

$$T_{\text{вик}} = \frac{T_p}{T_{pz}} = \frac{240}{8} = 30 \text{ днів} \quad (4.3.7)$$

Витрат на матеріали у процесі розробки цього проекту не було.

Витрати на електроенергію – розраховуються з потужності обладнання, що використовується при розробці продукту, коефіцієнту використання потужності, фактичного часу (трудомісткості) на розробку та ціни електроенергії, що склалась на момент розрахунку:

$$\text{Вел.} = P * K_{\text{вп}} * T_p * C_{1\text{кВт}} = 0,4 * 0,95 * 240 * 4,32 = 394 \text{ грн} \quad (4.3.8)$$

Де P – встановлена потужність комп'ютера та ін. пристроїв, кВт

K_{вп} – коефіцієнт використання потужності, приймаємо K_{вп} = 0,95

C_{1кВт} – фактична ціна 1 кВт., яка на момент розрахунку становить 4,32 грн/кВт · год.

Інші (накладні) витрати – до даної статі включають витрати, які не відносяться до вищеперелічених статей. Приймаються у відсотках, прийнятих на підприємстві, можуть коливатись в межах від 25 до 50%. Приймаємо в розмірі 25%.

$$\begin{aligned} \text{Внакл.} &= (\text{ЗПосн} + \text{ЗПдод} + \text{Нзп}) * \frac{\%_{\text{накл}}}{100} = (13636 + 2999 + 3659) * \\ 25\% &= 5073 \text{ грн} \end{aligned} \quad (4.3.9)$$

Де $\%_{\text{накл.}}$ – відсоток інших накладних витрат, % (приймаємо 25%)

Витрати на розробку програмного продукту визначається підсумуванням всіх статей, які включаються до витрат розробки:

$$\begin{aligned} \text{Врозр} &= \text{ЗПосн} + \text{ЗПдод} + \text{Нзп} + \text{А} + \text{Вм} + \text{Вел} + \text{Внак} = 13636 + \\ 2999 + 3659 + 369 + 0 + 394 + 5073 &= 26130 \text{ грн} \end{aligned} \quad (4.3.10)$$

Розрахунок собівартості одиниці копії програмного продукту

Собівартість одиниці продукції являє собою сукупні витрати на її виробництво і збут. До її складу включають вартість всіх ресурсів, що були використані для розробки вхідного продукту, витрати на розповсюдження реклами, участь у виставках. Також включають вартість розміщення на сервері сайту, який містить інформацію про розроблене інноваційне рішення.

Собівартість одиниці копії (Св) визначають з врахуванням обсягу попиту, що очікується на даний виріб (обсягу продажу):

$$\text{Св.} = \frac{\text{Врозр} + \text{Врозм.с} + \text{Врек} + \text{Вінш}}{\text{Опрод.}} = \frac{26130 + 330 + 500 + 500}{1} = 27460 \text{ грн} \quad (4.3.11)$$

Де Врозр. – витрати на розробку програмного продукту, грн

Врозм.с. – витрати, пов'язані з розміщенням інформації на сервері та вартість хостингу сайту, грн

Врек – витрати на рекламу, грн

Вінш – інші витрати, пов'язані з продажем продукту

Опрод – одиниць продажу копій програмного забезпечення. Приймаємо що Опрод – 1 шт.

Розрахунок ціни реалізації проектного рішення

Оскільки, програмний продукт приймається та впроваджується за завданням замовника, а також може реалізовуватись іншим покупцем відповідно доцільно встановлювати верхню та нижню межі ціни.

Нижня межа ціни – це рівень мінімально допустимої ціни на продукцію. Встановлення ціни на товар нижче зазначеного рівня робить подальше його виробництво та продаж не доцільним, але бувають випадки коли необхідно штучно занижувати ціну для початку аби захопити певну цільову групу споживачів.

Нижню межу ціни визначаємо з врахуванням нормативу рентабельності, який дозволить завоювати цільову аудиторію. Його рівень може коливатись в межах 15-25%. Приймаємо $P = 18\%$.

Верхня межа визначається з врахування нормативу рентабельності, який було узгоджено з замовником або встановлений державою.

Ціна реалізації розраховується за формулою:

$$\text{Цр.} = \text{Св} * \left(1 + \frac{P}{100}\right) * \left(1 + \frac{\text{ПДВ}}{100}\right) = 27460 * \left(1 + \frac{18}{100}\right) * \left(1 + \frac{20}{100}\right) = 38883 \text{ грн} \quad (4.3.12)$$

де P – прийнятий норматив рентабельності, %; Приймаємо $P = 18\%$.

ПДВ – ставка податку на додану вартість, приймається на рівні 20%.

Нижня (Цр.н.м) і верхня (Цр.в.м) межі ціни реалізації розраховуються за формулою 4.3.11 у випадку, якщо приймаються 2 нормативи рентабельності (норматив нижньої та верхньої межі).

Критичний обсяг продажу характеризує той обсяг реалізації копій продукту, який забезпечить беззбиткове проектне рішення. Для розрахунку критичного обсягу використовують показники умовно-постійні та умовно-змінні витрати.

Будемо вважати, що умовно-змінні витрати у складі собівартості продукту становлять – 70%, а умовно-постійні – 30%.

Визначення критичного обсягу (Опрод.кр., копій) продажу здійснюється

$$\text{за формулою: Опрод. кр} = \frac{Св * \text{Опрод.} * \frac{\%ПВ}{100}}{Цр - (Св * \frac{\%ЗВ}{100})} = \frac{27460 * 1 * \frac{30}{100}}{38883 - (27460 * \frac{70}{100})} = 0,419 \quad (4.3.13)$$

Приймаємо 1 шт.

де %ПВ – відсоток умовно-постійних витрат в собівартості, %;

%ЗВ – відсоток умовно-змінних витрат в собівартості;

Розрахунок чистого прибутку

Розрахунок чистого прибутку від реалізації запланованого обсягу програмного продукту здійснюється з врахуванням річного прибутку від реалізації та податку на прибуток, що існує на момент розрахунку. Прибуток від реалізації розраховується за формулою:

$$П = (Цр - Св) * \text{Опрод} = (38883 - 27460) * 1 = 11423 \text{ грн} \quad (4.3.14)$$

Відповідно чистий прибуток:

$$\text{ЧП} = П - \left(П * \frac{\%ПП}{100} \right) = 11423 - \left(11423 * \frac{18}{100} \right) = 9367 \text{ грн} \quad (4.3.15)$$

де %ПП – відсоток на прибуток, приймається на рівні 18%.

Термін окупності характеризує період часу протягом якого окупляється витрати на розробку програмного продукту замовником.

Розрахунок періоду окупності здійснюється за формулою:

$$\text{Ток} = \frac{Врозр}{\text{ЧП}} = \frac{26130}{9367} = 2,79 \text{ років} \quad (4.3.16)$$

4.4 Охорона праці

4.4.1 Розрахунок освітлення в приміщенні

Коротка характеристика приміщення і виконуваних робіт:

Приміщення призначене для роботи одного програміста. Основне обладнання робочого місця включає персональний комп'ютер (системний блок, два монітори, клавіатура, миша), розміщений вздовж короткої стіни. Робота пов'язана з тривалим перебуванням за комп'ютером. Приміщення має один вхід, обладнання та меблі розміщені таким чином, щоб забезпечити вільний доступ до робочого місця та шляхів евакуації.

Планування і розміщення робочих місць:

Робоче місце організовано з урахуванням ергономічних вимог для забезпечення комфортних та безпечних умов праці, а також для запобігання втомі та професійних захворюваннях. Робочий стіл має такі характеристики:

Висота робочої поверхні столу: 75.5см

Ширина робочої поверхні: 132см

Глибина робочої поверхні: 77см

Простір для ніг: ширина – 65 см, висота – 63.5 см.

Розрахунок освітлення:

Робоча кімната має розміри 4,0м x 2,5м. Таким чином площа складає 10 м². З висотою $H = 2.8$ м площа приміщення складе 28 м³. У приміщенні знаходиться 1 вікно висотою 1,12м і шириною 0,75м і двері з двома скляними частинами висотою 0,85м і шириною 0,3м на протилежній стороні. Згідно БНП П-4-79, коефіцієнт природної освітленості (КПО) у цьому випадку при бічному освітленні повинен складати $e_n = 1,5\%$.

Враховуючи коефіцієнт світлового клімату $m = 0,8$ і коефіцієнт сонячності клімату при тому, що вікна приміщення виходять на східний бік, $C = 0,7$, визначаємо нормоване значення КПО для даного приміщення

$$e = e_n * m * C = 1,5 * 0,5 * 0,7 = 0,525\%$$

При бічному освітленні КПО можна оцінити за формулою:

$$e = \frac{\tau_0 * r_1 * S_0}{K_3 * n_0 * K_{зд} * S_n} * 100\%$$

де S_n – площа підлоги приміщення, m^2

S_0 – площа світлових прорізів, m^2

K_3 – коефіцієнт запасу (приймається в межах від 1,2 до 2,0 залежно від можливого забруднення світлових прорізів)

n_0 – світлова характеристика вікон

$K_{зд}$ – коефіцієнт, що враховує підвищення коефіцієнта завдяки світлу, відбитому від поверхні приміщення та підстиляючому шару, який прилягає до будівлі

τ_0 – загальний коефіцієнт світлопропускання, що визначається за формулою:

$$\tau_0 = \tau_1 + \tau_2 + \tau_3 + \tau_4$$

Де τ_1 – коефіцієнт світлопропускання матеріалу

τ_2 – коефіцієнт, що враховує втрати світла в межах вікна

τ_3 – коефіцієнт, що враховує втрати світла в несучих конструкціях

τ_4 – коефіцієнт, що враховує втрати світла в сонцезахисних пристроях

Площа світлових прорізів становить $S_0 = 1,12 * 0,75 + 0,85 * 0,3 + 0,85 * 0,3 = 1,35 \text{ м}^2$. Площа підлоги приміщення становить $S_{\text{п}} = 10 \text{ м}^2$. Коефіцієнт запасу приймається на рівні $K_3 = 1,2$.

Виходячи зі співвідношення розмірів приміщення і вікон, світлова характеристика вікон $n_0 = 0,135$. Коефіцієнт, що враховує КПО за рахунок відбиття від поверхонь приміщення і підстиляючого шару, для даного приміщення становить $r_1 = 1,3$.

Для вікон з подвійними рамами коефіцієнт світлопропускання $\tau_1 = 0,8$

Оскільки палітурка вікна подвійна металопластикові, то $\tau_2 = 0,8$. При бічному освітленні $\tau_3 = 1$. Як сонцезахисних пристроїв використовуються регульовані внутрішні штори, тому $\tau_4 = 0,9$.

Таким чином, отримуємо

$$\tau_1 = 0,8 * 0,8 * 1 * 0,9 = 0,576$$

$$e = \frac{0,576 * 1,3 * 1,35}{1,2 * 0,135 * 1 * 10} = 0,623\%$$

Тож природнього освітлення недостатньо, необхідно передбачити додаткове освітлення від люмінесцентних світильників та світлодіодними лампами біля робочої зони.

4.4.2 Обґрунтування робочого місця оператора

У процесі розробки робоче місце повинне відповідати певним вимогам. Монітор повинен розташовуватися на відстані 45-70 сантиметрів від очей, висота столу та стільця підбирається так, щоб передпліччя були паралельні підлозі, а кути в ліктьових, тазостегнових та колінних суглобах становили приблизно 90 градусів. Спина має підтримуватися спинкою стільця, особливо в поперековій ділянці. Стопи повинні повністю спиратися на підлогу або

спеціальну підставку, а клавіатура та миша повинні знаходитися в зоні легкої досяжності. На рисунку 4.4.1 зображена ергономіка робочого місця.

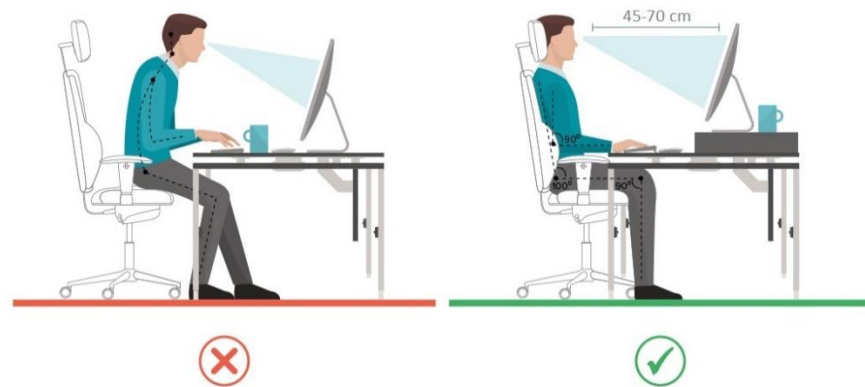


Рисунок 4.4.1 – Правильне положення за комп'ютером

Організація робочого місця повинна відповідати ергономічним вимогам для забезпечення високої працездатності. Виконаємо обґрунтування робочого місця.

Робочий стіл повинен бути достатньо просторим для розміщення монітора, клавіатури та миші. Висота столу повинна забезпечувати правильне положення рук та ліктів. Клавіатура розташовується на поверхні столу на відстані 100-300 мм від краю, повернутого до користувача, з кутом нахилу від 5 до 15 градусів, а висота клавіатури не повинна перевищувати 30 мм. Під столом повинно бути достатньо простору для вільного розміщення ніг. Рисунок 4.4.2 зображує ергономіку робочого місця для комфортної роботи за комп'ютером.

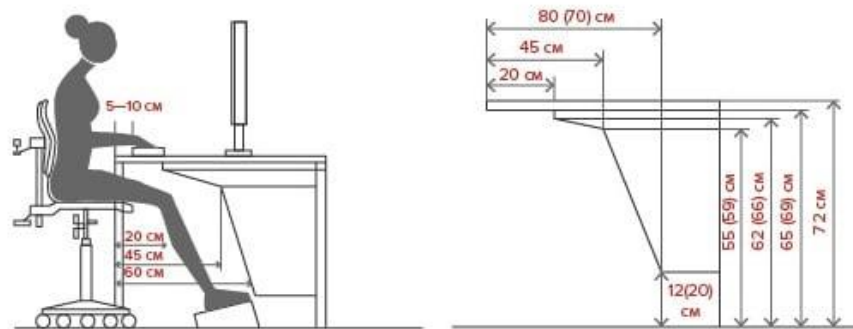


Рисунок 4.4.2 – Ергономіка робочого місця

Підберемо відповідний стіл та стілець для роботи за комп'ютером. На рисунку 4.4.3 зображено модель робочого столу і на рисунку 4.4.4 зображено модель робочого стільця програміста.



Рисунок 4.4.3 – Модель робочого столу програміста



Рисунок 4.4.4 – Модель робочого стільця програміста

ВИСНОВКИ

У кваліфікаційній роботі була встановлена мета, яка полягала у створенні функціонального, зручного та сучасного веб-додатку для реалізації органічної продукції у мережі Інтернет, і була успішно досягнута.

Для досягнення поставленої мети було розв'язано низку задач: проведено аналіз предметної області електронної комерції та ринку органічної продукції, що підтвердило актуальність розробки; здійснено огляд та порівняння існуючих програмних рішень, що дозволило виявити кращі практики та сформулювати вимоги до власного продукту; розроблено детальне технічне завдання та визначені функціональні і нефункціональні вимоги; спроектовано архітектуру системи за допомогою функціональних та концептуальних моделей; обрано та обґрунтовано технологічний стек; безпосередньо розроблено програмне забезпечення; проведено його тестування та розраховано техніко-економічні показники.

У результаті реалізації проєкту було розроблено працюючий прототип інтернет-магазину для продажу органічної їжі за допомогою можливостей фреймворку Vue.js. Розроблений додаток відповідає сформульованим вимогам і технічному завданню. Він охоплює усі ключові аспекти веб-розробки – від проєктування архітектури до практичного впровадження та тестування. Практична частина роботи підкреслила здатність застосовувати здобуті знання й навички для створення придатного до використання продукту.

В результаті виконання роботи було поглиблено знання щодо сучасних підходів до розробки веб-додатків, зокрема використання фреймворків Vue.js та Django. Було проаналізовано особливості проєктування та реалізації систем електронної комерції на прикладі продажу органічної продукції.

Розроблений інтернет-магазин реалізує основний функціонал для перегляду каталогу, додавання товарів у кошик, оформлення замовлення,

управління наповненням сайту через адмін-панель та має адаптивний користувацький інтерфейс. Тож створений продукт може слугувати основою для подальшого розвитку та впровадження у реальну комерційну діяльність після доопрацювання певного функціоналу та інтеграції з платіжними системами і службами доставки.

СПИСОК ЛІТЕРАТУРИ

1. E-commerce: як змінився ринок за рік. URL: <https://speka.media/e-commerce-yak-zminivsia-rinok-za-rik-9qyrwk>.
2. Малохвій Е. Е., Бугай В. С., Молчанов Г. І., Черних О. П. Аналітичний огляд та порівняння сучасних JavaScript рішень для розробки веб-додатків. *Системи управління, навігації та зв'язку*: зб. наук. пр. 2021. № 4(66). С. 55–58.
3. Жилияєв В. Мови програмування для веб-розробки та супутні технології. URL: <https://drukarnia.com.ua/articles/movi-programuvannya-dlya-vebrozrobki-ta-suputni-tekhnologiyi-wzeAr>.
4. Амонс С. Е. Стан та перспективи розвитку виробництва органічної продукції в Україні. *Сільське господарство та лісівництво*. 2021. № 22. С. 221 – 236.
5. Багорка М. О., Якубенко Ю. Л. Розвиток ринку органічної продукції в Україні: проблеми, перспективи та шляхи їх вирішення. *Агросвіт*. 2025. № 4. С. 16 – 23.
6. Бурковська А. І., Бурковська А. В. Перспективи розвитку ринку органічної продукції на прикладі Франції. *Сучасний менеджмент економічних систем в координатах парадигми сталого розвитку: матеріали III міжнар. наук.-практ. конф.* 2021. С. 93 – 95.
7. Про основні принципи та вимоги до органічного виробництва, обігу та маркування органічної продукції : Закон України від 10.07.2018 р. № 2496-VIII. URL: <https://zakon.rada.gov.ua/laws/show/2496-19#Text>.
8. Іванов Д. А., Вакалюк Т. А. Огляд JavaScript-фреймворків для створення користувацьких інтерфейсів. *Тези доповідей III Всеукраїнської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих учених «Інформаційно-комп'ютерні технології: стан, досягнення та перспективи розвитку»* (Житомир, 26–27 листопада 2020 р.). Житомир: Житомирська політехніка, 2020. С. 9 – 10.

9. Ілляшенко С. М., Іванова Т. Є. Перспективи та основні проблеми розвитку Internet-торгівлі в Україні. *Механізм регулювання економіки*. 2014. № 3. С. 72 – 81.
10. Крилов Д. В. Розвиток електронної комерції в Україні в сучасних умовах. Проблеми сучасних трансформацій. *Економіка та управління*. 2024. № 12. DOI: 10.54929/2786-5738-2024-12-03-02.
11. Лебединський А., Черкасов М. Аналіз сучасних систем управління базами даних. *Комп'ютерні технології і мехатроніка*. 2023. С. 32 – 36.
12. Органічний ринок в Україні. Вікіпедія: вільна енциклопедія. URL: <https://uk.wikipedia.org/w/index.php?curid=5215924>.
13. Новицька І. В. Управління просуванням органічної продукції через канали розподілу. *Проблеми системного підходу в економіці*. 2019. № 5(2). С. 156 – 161.
14. Пахуча Е. В., Чміль Г. Л., Олініченко К. С. Маркетингові технології просування органічної продукції в умовах цифровізації. *Економіка і регіон*. 2022. № 3(86). С. 72 – 79. DOI: 10.26906/EiR.2022.3(86).2650.
15. Писаренко В. В. Реалізація органічної сільськогосподарської продукції через мережу Інтернет. *Ефективна економіка*. 2015. № 3. С. 1 – 3.
16. Риждова Г. А. Особливості розвитку електронної торгівлі в Україні; Features of the development of e-commerce in Ukraine. *Економічний вісник Дніпровської політехніки*. 2024. С. 120 – 129.
17. Романчук О.І., Біляк Т.О. Перспективи розвитку інтернет-торгівлі в Україні. 2017. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2017/06/131-1.pdf>.
18. Світові тренди e-commerce 2024 в цифрах. URL: <https://www.proresultat.com/svitovi-trendi-e-commerce2024-v-cifrah>.
19. Django: офіц. технічна документація. URL: <https://docs.djangoproject.com/en/5.2/>.
20. Vue.js: офіц. документація API. URL: <https://vuejs.org/api>.

- 21.Ткачов В. Аналіз JavaScript-фреймворків для розроблення вебзастосунків. *Integration of science as a mechanism of effective development: матеріали XI Міжнар. Наук.-практ. конф. (м. Гельсінкі, 28 листоп. – 1 груд. 2023 р.).* Helsinki: International Science Group. 2023. С. 466 – 468.
- 22.Тузенко О. О., Володін С. І. Розробка веб-додатку для електронної комерції. *Вісник Приазовського Державного Технічного Університету. Серія: Технічні науки.* 2023. № 47. С. 58 – 66.
- 23.Шпикуляк О. Г., Беженар І. М., Білокінна І. Д. Сучасні тенденції розвитку органічного виробництва в аграрному секторі і фермерських господарствах. *Проблеми сучасних трансформацій. Серія: економіка та управління.* 2023. № 8. DOI: 10.54929/2786-5738-2023-8-04-04.
- 24.Щитов Д. М., Жадько К. С., Мормуль М. Ф. Структура товарної української електронної комерції. *Ефективна економіка.* 2024. № 9. DOI: 10.32702/2307-2105.2024.9.7.
- 25.Як відкрити інтернет-магазин і як у цьому допомагають цифрові інструменти? URL: <https://vchasno.ua/yak-vidkryty-internet-mahazyn/>.

Додаток А

Код компоненту картки товару ProductCard.vue

```

<template>
  <div class="product-card">
    <span v-if="stock < 1" class="out-of-stock-label">Товар
закінчився</span>
    
    <div class="product-card__bottom">
      <div class="product-card__bottom-left">
        <span class="product-card__name">{{ name }}</span>
        <span class="product-card__price">{{ price }} €</span>
        <div class="product-card__rating">
          
          </div>
        </div>
        <div class="product-card__bottom-right">
          <CircleButton v-if="stock > 0" @click="addToCart"
:btnStyle="{ background: '#F2F2F2', hoverBackground: '#00B207'
}" />
        </div>
      </div>
    </div>
  </div>
</template>

<script>
import CircleButton from '@components/CircleButton.vue';
export default {
  name: 'ProductCard',
  components: {
    CircleButton
  },
  props: {
    id: {
      type: [Number],
      default: null
    },
    imgUrl: {
      type: String,
      default: 'eggPlant'
    },
    stock: {
      type: Number,

```

```

        default: 0
      },
      rating: {
        type: Number,
        default: 4
      },
      name: {
        type: String,
        default: "Product"
      },
      price: {
        type: Number,
        default: 0.00
      }
    },
    data() {
      return {

      }
    },
    computed: {},
    methods: {
      addToCart() {
        var product = {
          id: this.id,
          imgUrl: this.imgUrl,
          name: this.name,
          price: this.price,
          quantity: 1,
          stock: this.stock,
        };
        this.$store.commit('userData/addToCart', product);
        this.$store.commit("userData/setShowCart", true)
      }
    }
  }
</script>

<style scoped lang="scss">
@import "@/assets/styles/components/product-card.scss";
</style>;

```


Додаток Б

Код компоненту шапки сайту HeaderBottom.vue

```

<template>
  <div class="header-bottom">
    <div class="header-bottom__nav">
      <div class="header-bottom__nav-categories"
@mouseenter="toggleWrapper" @mouseleave="toggleWrapper">
        <div class="header-bottom__nav-categories-hamburger"
@click="$emit('toggleMenu')"></div>
        <span class="header-bottom__nav-categories-text"
@click="$emit('toggleMenu')"
:style="{ backgroundColor: isCategoriesActive ?
'#333333' : '#1A1A1A' }">Yci kateropii</span>
        <div class="vertical-list__wrapper" v-if="!isMobile &&
isWrapperActive">
          <transition enter-active-class="animate__animated
animate__fadeInDown animate__faster"
leave-active-class="animate__animated
animate__fadeOutUp animate__faster">
            <VerticalList v-if="isCategoriesActive"
:itemsArray="categories" :isActive="isCategoriesActive">
              </VerticalList>
            </transition>
          </div>
        </div>
      <ul class="header-bottom__nav-list">
        <HeaderItem v-for="item, index in nav" :key="index"
:pageName="item.name"></HeaderItem>
      </ul>
    </div>
    <div class="header-bottom__icons">
      <div class="header__wishlist">
        
      </div>
      <div class="header__cart"
@click="toggleCart"
:style="{ '--cartVisibility': getCartLength > 0 ? 'flex'
: 'none', '--cartCount': ` '${getCartLength}' ` }">
        
      </div>
      <div class="header__user">
        <router-link :to="/account/dashboard">
          
        </router-link>
      </div>
    </div>
  </div>

```

```

        </div>
      </div>
      <CartPopup v-if="getShowCart" @close="setShowCart(false)" />
    </div>
  </template>

  <script>
import HeaderItem from '@components/Header/HeaderItem.vue';
import VerticalList from '@widgets/VerticalList.vue';
import HeaderLogo from '@components/Header/HeaderLogo.vue';
import { mapGetters, mapMutations } from "vuex";
import CartPopup from '@widgets/CartPopup.vue';
export default {
  name: 'HeaderBottom',
  components: {
    HeaderItem,
    VerticalList,
    HeaderLogo,
    CartPopup,
  },
  props: {
    isCategoriesActive: {
      type: Boolean,
      default: false
    }
  },
  data() {
    return {
      isWrapperActive: false,
    }
  },
  methods: {
    ...mapMutations("userData", ["toggleCart", "setShowCart"]),
    toggleWrapper() {
      this.isWrapperActive = !this.isWrapperActive;
    },
  },
  computed: {
    ...mapGetters({
      isMobile: 'isMobile',
      nav: 'getNav',
      categories: 'products/getCategories',
      getShowCart: 'userData/getShowCart',
      getCartLength: 'userData/getCartLength'
    })
  },
}
  </script>
  <style scoped lang="scss">
@import "@assets/styles/widgets/header/header-bottom.scss";
@import "@assets/styles/common.scss"
  </style>;

```

Додаток В

Реалізація Vuex сховища Products.js

```

import { fetchProducts, fetchCategories } from "@api/products";

const products = {
  namespaced: true,
  state: {
    products: [],

    categories: {},
  },
  actions: {
    async loadProducts({ commit }, params) {
      const products = await fetchProducts(params);
      commit("setProducts", products);
    },
    async loadCategories({ commit, getters }) {
      const categories = await fetchCategories();
      const products = getters.getProducts
      categories.forEach((category) => {
        category.count = products.filter(p =>
p.category === category.id).length;
      });
      commit("setCategories", categories);
    },
  },
  mutations: {
    setProducts(state, products) {
      state.products = products;
    },
    setCategories(state, categories) {
      state.categories = categories;
    },
  },
  getters: {
    getProducts(state) {
      return state.products;
    },
    getCategories: (state) => state.categories,
    sortedProducts: (state) => (filters) => {
      const {
        selectedCategory = "all",
        minPrice = 0,
        maxPrice = Infinity,
        selectedRatings = [],
        selectedTags = [],

```

```

        sortBy = null, // 'priceAsc', 'priceDesc',
        'ratingAsc', 'ratingDesc'
      } = filters;
      return state.products
        .filter((product) => {
          return (
            selectedCategory === "all" ||
            product.category ===
selectedCategory
          );
        })
        .filter((product) => {
          return (
            product.price >= minPrice &&
product.price <= maxPrice
          );
        })
        .filter((product) => {
          return (
            selectedRatings.length === 0 ||
            selectedRatings.includes(product.rating)
          );
        })
        .filter((product) => {
          return (
            selectedTags.length === 0 ||
            product.tags.includes(...selectedTags)
          );
        })
        .sort((a, b) => {
          switch (sortBy) {
            case "priceAsc":
              return a.price - b.price;
            case "priceDesc":
              return b.price - a.price;
            case "ratingAsc":
              return a.rating - b.rating;
            case "ratingDesc":
              return b.rating - a.rating;
            default:
              return 0;
          }
        });
    },
  },
};

export default products;

```

Додаток Г

Структура моделей бази даних у Django

```

from django.db import models
from django.contrib.auth.models import User
from django.db.models.signals import post_save
from django.dispatch import receiver

class Category(models.Model):
    name = models.CharField(max_length=100)
    visibleName = models.CharField(max_length=100)
    description = models.TextField(blank=True)
    parent = models.ForeignKey('self',
on_delete=models.SET_NULL, null=True, blank=True,
related_name='children')

    def __str__(self):
        return self.name

class Product(models.Model):
    name = models.CharField(max_length=255)
    description = models.TextField(blank=True)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    sale_price = models.DecimalField(max_digits=10,
decimal_places=2, null=True, blank=True)
    is_sale = models.BooleanField(default=False)
    sale_percent = models.PositiveSmallIntegerField(default=0)
    stock = models.PositiveIntegerField(default=0)
    rating = models.FloatField(default=0)
    thumbnail = models.ImageField(upload_to='products/')
    is_active = models.BooleanField(default=True)
    category = models.ForeignKey(Category,
on_delete=models.SET_NULL, null=True, blank=True,
related_name='products')
    tags = models.CharField(max_length=255, blank=True,
help_text="Список тегів через кому")
    calories = models.FloatField(null=True, blank=True,
help_text="Калорійність (ккал на 100г)")
    protein = models.FloatField(null=True, blank=True,
help_text="Білки (г на 100г)")
    fat = models.FloatField(null=True, blank=True,
help_text="Жири (г на 100г)")
    carbs = models.FloatField(null=True, blank=True,
help_text="Вуглеводи (г на 100г)")
    manufacturer = models.CharField(max_length=255, blank=True,
help_text="Виробник")

    def __str__(self):

```

```

        return self.name

class ProductImage(models.Model):
    product = models.ForeignKey('Product',
    on_delete=models.CASCADE, related_name='images')
    image = models.ImageField(upload_to='products/')
    alt_text = models.CharField(max_length=255, blank=True,
    default='')

    def __str__(self):
        return f"{self.product.name} image"

class Address(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE,
    null=True, blank=True, related_name='addresses')
    first_name = models.CharField(max_length=64)
    last_name = models.CharField(max_length=64)
    company_name = models.CharField(max_length=128, blank=True)
    street = models.CharField(max_length=255)
    city = models.CharField(max_length=64)
    country = models.CharField(max_length=64)
    # state = models.CharField(max_length=64)
    zip_code = models.CharField(max_length=20)
    email = models.EmailField()
    phone = models.CharField(max_length=32)

    def __str__(self):
        return f"{self.first_name} {self.last_name},
    {self.street}, {self.city}"

class Cart(models.Model):
    user = models.ForeignKey(User, on_delete=models.SET_NULL,
    null=True, blank=True)
    session_id = models.CharField(max_length=255, blank=True,
    null=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

class CartItem(models.Model):
    cart = models.ForeignKey(Cart, on_delete=models.CASCADE,
    related_name='items')
    product = models.ForeignKey(Product,
    on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField(default=1)
    price = models.DecimalField(max_digits=10, decimal_places=2)

class PaymentMethod(models.Model):
    name = models.CharField(max_length=64)

class ShippingMethod(models.Model):
    name = models.CharField(max_length=64)

```

```

    price = models.DecimalField(max_digits=10, decimal_places=2,
                                default=0)

class Order(models.Model):
    class Status(models.TextChoices):
        NEW = 'new', 'Нове'
        PAID = 'paid', 'Оплачено'
        PROCESSING = 'processing', 'В обробці'
        SHIPPED = 'shipped', 'Відправлено'
        DELIVERED = 'delivered', 'Доставлено'
        CANCELED = 'canceled', 'Скасовано'

    user = models.ForeignKey(User, on_delete=models.SET_NULL,
                             null=True, blank=True, verbose_name="Користувач")
    session_id = models.CharField(max_length=255, blank=True,
                                   null=True, verbose_name="Сесія (для аноніма)")
    first_name = models.CharField(max_length=64,
                                   verbose_name="Ім'я")
    last_name = models.CharField(max_length=64,
                                  verbose_name="Прізвище")
    company = models.CharField(max_length=128, blank=True,
                               null=True, verbose_name="Компанія")
    country = models.CharField(max_length=64,
                               verbose_name="Країна")
    city = models.CharField(max_length=64, verbose_name="Місто")
    street = models.CharField(max_length=128,
                              verbose_name="Вулиця, номер будинку")
    zip = models.CharField(max_length=16, verbose_name="Поштовий
індекс")
    address = models.ForeignKey('Address',
                                on_delete=models.SET_NULL, null=True, blank=True,
                                verbose_name="Адреса користувача (для зареєстрованих)")
    email = models.EmailField(verbose_name="Email", null=True,
                               blank=True)
    phone = models.CharField(max_length=32,
                              verbose_name="Телефон")
    payment_method = models.ForeignKey('PaymentMethod',
                                        on_delete=models.SET_NULL, null=True, blank=True)
    shipping_method = models.ForeignKey('ShippingMethod',
                                        on_delete=models.SET_NULL, null=True, blank=True)
    shipping_price = models.DecimalField(max_digits=10,
                                         decimal_places=2, default=0)
    total_price = models.DecimalField(max_digits=10,
                                      decimal_places=2)
    created_at = models.DateTimeField(auto_now_add=True)
    comment = models.TextField(blank=True)
    status = models.CharField(
        max_length=32,
        choices=Status.choices,
        default=Status.NEW,
        verbose_name="Статус замовлення"
    )

```

```

    def __str__(self):
        return f"Заказовлення #{self.pk} - {self.phone}"
    [{self.get_status_display()}]]"

class OrderItem(models.Model):
    order = models.ForeignKey(Order, on_delete=models.CASCADE,
related_name='items')
    product = models.ForeignKey(Product,
on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField(default=1)
    price = models.DecimalField(max_digits=10, decimal_places=2)

class Review(models.Model):
    user = models.ForeignKey(User, on_delete=models.SET_NULL,
null=True, blank=True)
    product = models.ForeignKey(Product,
on_delete=models.CASCADE, related_name='reviews')
    rating = models.PositiveSmallIntegerField()
    text = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)

class UserProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE,
related_name='profile')
    phone_number = models.CharField(max_length=20, blank=True,
null=True)

    def __str__(self):
        return self.user.username + "'s profile"

@receiver(post_save, sender=User)
def create_user_profile(sender, instance, created, **kwargs):
    if created:
        UserProfile.objects.create(user=instance)

@receiver(post_save, sender=User)
def save_user_profile(sender, instance, **kwargs):
    instance.profile.save()

```


Додаток Д

Панель адміністратора

Django administration

Home > Shop > Products > Зелений чілі

Start typing to filter...

AUTH TOKEN

Tokens [+ Add](#)

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

SHOP

Address [+ Add](#)

Cart items [+ Add](#)

Carts [+ Add](#)

Categorys [+ Add](#)

Order items [+ Add](#)

Orders [+ Add](#)

Payment methods [+ Add](#)

Products [+ Add](#)

Reviews [+ Add](#)

Shipping methods [+ Add](#)

Change product

Зелений чілі

Name: Зелений чілі

Description: Органічний зелений чілі — натуральне джерело гостроти та аромату. Вирощений без пестицидів і хімічних добрив, він має яскравий смак із пряним післясмаком. Ідеально доповнює соуси, маринади, м'ясні та овочеві страви. Насичений вітамінами А і С, а також корисними антиоксидантами, зелений чілі не лише додає пікантності, а й підтримує імунітет.

Price: 40,00

Sale price:

☐ Is sale

Sale percent: 0

Stock: 0

Rating: 0,0

Thumbnail: Currently: products/greenChili.jpg
Change: [Выберите файл](#) Файл не выбран

☒ Is active

Category: vegetables [✎](#) [+](#) [✖](#) [👁](#)

Рисунок Г.1 – Редагування інформації про товар

Start typing to filter...

AUTH TOKEN

Tokens

+ Add

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

Users

+ Add

SHOP

Addresss

+ Add

Cart items

+ Add

Carts

+ Add

Categorys

+ Add

Order items

+ Add

Orders

+ Add

Payment methods

+ Add

Products

+ Add

Reviews

+ Add

Shipping methods

+ Add

Add product

Name:

Description:

Price:

Sale price:

☐ Is sale

Sale percent:

0

Stock:

0

Rating:

0

Thumbnail:

Выберите файл

Файл не выбран

☒ Is active

Category:

Tags:

Список тегів через кому

Рисунок Г.2 – Додавання нового товару

Start typing to filter...

AUTH TOKEN

Tokens

+ Add

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

Users

+ Add

SHOP

Addresss

+ Add

Cart items

+ Add

Carts

+ Add

Categorys

+ Add

Order items

+ Add

Orders

+ Add

Payment methods

+ Add

Products

+ Add

Reviews

+ Add

Shipping methods

+ Add

Select order to change

Search

Search

Action:

Go

0 of 17 selected

☐	ID	CREATED AT	EMAIL	ТЕЛЕФОН	СТАТУС ЗАМОВЛЕННЯ	TOTAL PRICE	ТОВАРИ	COMMENT
☐	19	June 17, 2025, 6:03 p.m.	test@gmail.com	test	Нове	12.50	Перець солодийкий (x1)	-
☐	18	June 17, 2025, 5:56 p.m.	-	фів	Нове	35.80	Баклажани (x1)	-
☐	17	June 17, 2025, 5:34 p.m.	-	+38098571658	Нове	94.36	Перець солодийкий (x5), Цвітна капуста (x6), Курурудза (x5)	Побажання
☐	16	June 10, 2025, 11:46 p.m.	-	анів	Нове	105.00	Апельсини (x10)	ваприрап тСцион цЗей5нуке в
☐	15	June 10, 2025, 11:44 p.m.	guest@gmail.com	666777888	Скасовано	17.25	Курурудза (x15)	-
☐	14	June 10, 2025, 11:42 p.m.	guest@gmail.com	55112233	Нове	35.77	Зелені яблука (x3), Цвітна капуста (x2), Курурудза (x9)	-
☐	11	June 10, 2025, 10:41 p.m.	guest@gmail.com	55112233	Оплачено	3.50	Баклажани (x1)	-
☐	10	June 10, 2025, 10:38 p.m.	-	dfgs	Нове	24.00	Зелені яблука (x5)	-
☐	9	June 10, 2025, 10:35 p.m.	test@user.com	0443332211	Нове	33.33	Перець солодийкий (x3)	-
☐	8	June 10, 2025, 10:33 p.m.	test@user.com	0991112233	Нове	56.70	Перець солодийкий (x5), Курурудза (x1)	-
☐	7	June 10, 2025, 10:31 p.m.	test@user.com	0991112233	Нове	55.55	Перець солодийкий (x5)	-
☐	6	June 10, 2025, 2:43 p.m.	-	0681112233	В обробці	105.00	Апельсини (x10)	just a comment to test how it works lol
☐	5	June 10, 2025, 2:36 p.m.	-	cnvb	Нове	5.51	Цвітна капуста (x1)	fgdfg456 23456 345

Рисунок Г.3 – Перегляд надісланих замовлень

Додаток Е

Наукова стаття

8

СЕКЦІЯ 2

Комп'ютерні науки

РОЗРОБКА ВЕБ-СИСТЕМИ ДЛЯ ПРОДАЖУ ОРГАНІЧНОЇ ЇЖИ З
ВИКОРИСТАННЯМ VUE.JS

Ціль сталого розвитку №12: Відповідальне споживання

Васковський А. С. artem.vaskovskyi@tsatu.edu.ua*Таврійський державний агротехнологічний університет імені Дмитра Моторного*

Попит на органічну їжу з кожним роком зростає все більше і більше. Це зумовлено зацікавленістю людей споживати корисні продукти харчування та дбати про своє здоров'я та здоров'я своїх рідних. Як показують дослідження, 75% людей в Україні потенційно готові купувати органічну продукцію та навіть переплачувати на 10% більше у порівнянні зі звичайними продуктами [1]. Але не весь потрібний асортимент може бути у наявності та в одному місці. Часто для того щоб знайти деякі позиції товарів необхідно відвідати не тільки великі мережеві магазини, але й ринки, ярмарки або інтернет магазини.

Кількість людей які віддають перевагу онлайн шопінгу зростає з кожним роком. У 2024 році українці витратили на онлайн-покупки 239 мільярдів гривень, що на 25% більше, ніж у 2023 році [2].

Інтернет торгівля пропонує споживачам у зручний спосіб переглянути увесь наявний асортимент, прочитати детальну інформацію про конкретні продукти, ознайомитися з виробниками та умовами виробництва. Також онлайн шопінг дозволяє зекономити кошти, до нього можна отримати доступ у будь який час, є можливість дізнатися думку інших користувачів а також немає потреби витрачати час на традиційні відвідування магазинів. До того ж наявність інтернет-магазину допомагає підвищити рівень довіри та обізнаності покупця щодо бренду та виробників продукції. Тож створення інтернет-магазину є важливим етапом для розширення бізнесу, залучення ширшої аудиторії та інформування споживачів про переваги здорової органічної їжі.

Створення інтернет магазину включає в себе створення структури сайту, розробку користувацького інтерфейсу, верстки та програмування, підключення бази даних, наповнення контентом, тестування і розміщення сайту в інтернеті.

Для розробки інтерфейсу користувача використовують мову розмітки HTML (англ. Hypertext Markup Language), мову стилю сторінок CSS (англ. Cascading Style Sheets) і мову програмування JavaScript. HTML використовується для розробки структури сторінок, він описує як елементи будуть розташовані на сторінці. CSS використовується для задання стилів і загального виду сторінки. Він допомагає задавати відступи від елементів, який колір, розмір або шрифт буде у тексті і т.д.. JavaScript додає інтерактивності і відповідає за те, як сторінка буде реагувати на дії користувача. Наприклад, при замовленні товарів в інтернет магазині, JavaScript допоможе відправити запит на сервер, який повідомить персонал про надходження нового замовлення.

Vue.js – це фреймворк для JavaScript, який використовується для побудови користувацьких інтерфейсів і односторінкових програм [3]. Він значно полегшує створення інтерфейсів за допомогою «модульного» підходу. Vue.js дозволяє створювати окремі компоненти сторінки, які відповідають за свою частину інтерфейсу. Це значно зменшує повторне використання коду, полегшує тестування і дає змогу краще орієнтуватися у великих проєктах.

Кожен компонент Vue містить у собі 3 основні секції: <template>, <script>, та <style>.

Блок <template> відповідає за визначення структури інтерфейсу користувача, тобто HTML-розмітки. Вміст цього блоку також може включати в себе динамічні прив'язки (data bindings), які дозволяють динамічно відображати актуальну інформацію на екрані

користувача без необхідності перезавантажувати сторінку.

Блок `<script>` реалізовує логіку компонента і визначає його поведінку. У цьому блоці описується JavaScript код, визначаються змінні компонента та його методи.

Блок `<style>` містить у собі CSS стилі та відповідає за візуальне оформлення компонента.

На рисунку 1 наведено графічне зображення структури компонента Vue.js.



Рис. 1. Графічне зображення структури компонента Vue.js

Для забезпечення повноцінного функціонування ІНТЕРНЕТ-магазину необхідно також реалізувати і серверну частину. Сервер виконує роль посередника між користувацьким інтерфейсом та базою даних, оброблює запити, автентифікує користувачів і виконує бізнес логіку. Серверна частина може бути реалізована багатьма мовами програмування, серед яких: Python, JavaScript, PHP, Java та інші [4].

На таблиці 1 наведена порівняльна таблиця мов програмування, які використовуються для розробки серверної частини веб-сайту.

Таблиця 1.

Порівняльна характеристика мов програмування для розробки серверу.

Мова програмування	Основне призначення	Переваги	Недоліки	Популярні фреймворки
JavaScript	Універсальна мова для фронтенду та бекенду	Єдина мова для фронтенду та бекенду Велика спільнота асинхронність багато бібліотек	Відсутність строгої типізації Складність масштабування Однопоточність	Express.js; Nest.js

PHP	Розробка динамічних веб-сайтів	Простий у вивченні Підтримка багатьох баз даних Велика спільнота	Менш структурований код; Проблеми з розширюваністю та підтримкою коду для великих проєктів	Laravel; Symfony
Python	Широкий спектр застосувань, включаючи веб-розробку	Чистий, легкий, зрозумілий синтаксис Потужні фреймворки Велика екосистема Динамічна типізація та автоматичне керування пам'яттю Висока швидкість розробки	Менша продуктивність Повільніший при великих навантаженнях	Django; Flask
Java	Розробка масштабованих корпоративних систем	Висока продуктивність і масштабованість Платформонезалежність Потужні фреймворки Строга типізація	Громіздкий синтаксис Складність та велика крива навчання Складне розгортання	Spring, Jakarta EE

Тож можна зробити висновок, що кожна мова програмування має свої переваги та недоліки і вибір мови залежить від багатьох факторів, зокрема, від рівню досвідченості розробника, масштабів та характеру проєкту, а також від наявності відповідних інструментів і фреймворків. З огляду на ці аспекти, мова Python є доречним і оптимальним вибором для розробки інтернет-магазину з продажу органічної їжі, оскільки він поєднує у собі простоту використання, високу швидкість розробки та надійність.

Зберігання даних про користувачів, їх замовлення, контактну інформацію, а також дані про наявні товари відбувається у базі даних. Системи управління базами даних бувають реляційними, документо-орієнтованими, графовими, колонковими тощо [5]. Реляційні СУБД використовують таблиці з чітко визначеними схемами і підтримують мову SQL (англ. Structured Query Language) для виконання запитів. Вони добре підходять для структурованих даних і забезпечують надійність та цілісність даних.

Для розробки інтернет-магазину з продажу органічної їжі оптимальним вибором буде обрати реляційну базу даних MySQL або PostgreSQL. Вони є безкоштовними, мають велику спільноту, прості у налаштуванні і підтримують транзакції, що важливо для коректного оброблення замовлень і збереження даних користувачів.

Можна зробити висновок, що розробка веб-системи для продажу органічної їжі з використанням Vue.js є актуальним напрямом, що відповідає сучасним тенденціям зростання попиту на здорове харчування та розвитку електронної комерції в Україні.

Список використаних джерел

1. Андреа Бішоф, Олена Дейнеко, Катерина Загривенко, Назіля Ісмаїлова, д-р Торальф Піхтер, Катерина Шор. Обізнаність та сприйняття органічних продуктів в Україні. URL: <https://organicinitiative.org.ua/wp-content/uploads/2024/05/Consumer-study-presentation-2024-UA.pdf> (дата звернення 07.05.2025)
2. Павло Бартос. E-commerce: як змінився ринок за рік. URL: <https://speka.media/e-commerce-yak-zminivsya-rinok-za-rik-9qyrwk> (дата звернення 07.05.2025)
3. Jstify Community. Vue.js — навіщо він нам? URL: <https://medium.com/@jstify.community/vue-js-981b5e17af39> (дата звернення 07.05.2025)
4. Volodymyr Zhyliayev. Мови програмування для веброзробки та супутні технології. URL: <https://drukarnia.com.ua/articles/movi-programuvannya-dlya-vebrozrobki-ta-suputni-tekhnologiyi-wzeAr> (дата звернення 07.05.2025)
5. Ivanna Tkachuk. Типи баз даних: особливості, відмінності та приклади. URL: <https://dou.ua/lenta/articles/types-of-databases/> (дата звернення 07.05.2025)

Науковий керівник: *Гешева Г.В., асистент кафедри комп'ютерних наук Таврійського державного агротехнологічного університету імені Дмитра Моторного.*

Рисунок Е.4 – Сторінка 4