

Міністерство освіти і науки України

Відокремлений структурний підрозділ «Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»

(повне найменування вищого навчального закладу)

Відділення телекомунікаційних та електронних систем

(назва відділення)

Циклова комісія комп'ютерних наук

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

(освітньо-професійного ступеня)

на тему: ‘Розробка інтернет-магазину товарів для здорового способу життя
<Healthy Life>’

Виконав: студентка IV курсу, групи КН-423

Спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

Максим ОЛЕЩУК
(ім'я та прізвище)

Керівник Людмила ЦИМБАЛЮК
(ім'я та прізвище)

Рецензент _____
(ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук
Освітньо-професійний ступінь «фаховий молодший бакалавр»
Спеціальність 122 Комп'ютерні науки
Галузь знань 12 Інформаційні технології

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук

_____ Галина МАРЦІЯШ

« 02 » березня 2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Олещуку Максиму Івановичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка інтернет-магазину товарів для здорового способу життя «Healthy Life»

керівник роботи Цимбалюк Людмила Володимирівна,

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студентом роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мова програмування TypeScript; фреймворк Next.js (App Router), база даних MongoDB (Mongoose); бібліотеки: React, Zustand (стейт-менеджмент), Tailwind CSS (стилізація), Axios, JWT, Vercel, стандарти IEEE 29148-2018, IEEE 29119, ДСТУ 8302:2015.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу інтернет-магазину

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

2 Розробка технічного та робочого проєкту

2.1 Розробка структури інтернет-магазину і вебсторінок

2.2 Створення та верстка сторінок інтернет-магазину

2.3 Розробка структури бази даних інтернет-магазину

2.4 Програмування інтернет-магазину

2.4.1 Написання клієнтської частини

2.4.2 Написання серверної частини

2.5 Тестування інтернет-магазину

3 Спеціальний розділ

3.1 Інструкція з розгортання магазину в Інтернеті

3.2 Інструкція з наповнення інтернет-магазину

3.3 Інструкція з популяризації та підтримки інтернет-магазину

4 Економічний розділ

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

4.3 Розрахунок витрат на електроенергію

4.4 Розрахунок суми амортизаційних відрахувань

4.5 Обчислення накладних витрат

4.6 Складання кошторису витрат та визначення собівартості інтернет-магазину

4.7 Розрахунок ціни інтернет-магазину

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

5 Охорона праці, техніка безпеки та екологічні вимоги

5.1 Дії роботодавця при отриманні повідомлення про нещасний випадок

5.2 Чинники електричного характеру, що впливають на тяжкість ураження людини електричним струмом

6 Висновки

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – інтернет-магазину «HealthyLife».

5. Перелік графічного матеріалу:

1. Схема структурна клієнтської частини
2. UML-діаграма варіантів використання
3. ER-діаграма бази даних
4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Генадій ГОРЯЧЕК		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	
2	Збір і узагальнення інформації	01.05.2026	
3	Написання першого розділу	15.05.2026	
4	Розробка технічного та робочого проекту	29.05.2026	
5	Написання спеціального розділу	05.06.2026	
6	Розрахунок економічної частини	08.06.2026	
7	Написання розділу охорони праці	09.06.2026	
8	Виконання графічної частини	10.06.2026	
9	Оформлення пояснювальної записки	11.06.2026	
10	Погодження нормоконтролю	12.06.2026	
11	Попередній захист кваліфікаційної роботи	.06.2026	
12	Захист кваліфікаційної роботи	.06.2026	

7. Дата видачі завдання: 05 березня 2026 р.

Студент

_____ (підпис)

Максим ОЛЕЩУК

Керівник роботи

_____ (підпис)

Людмила ЦИМБАЛЮК

ЗМІСТ

ВСТУП	9
1 ЗАГАЛЬНИЙ РОЗДІЛ	11
1.1 Аналітичний огляд існуючих рішень	11
1.2 Технічне завдання	14
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ	27
2.1 Розробка структури сайту і web-сторінок	27
2.2 Створення та верстка сторінок сайту	29
2.3 Розробка структури бази даних сайту	33
2.4 Програмування сайту	35
2.5 Тестування web-сайту	42
3 СПЕЦІАЛЬНИЙ РОЗДІЛ	46
3.1 Інструкція з розміщення сайту в Інтернеті	46
3.2 Інструкція з обслуговування та наповнення сайту	49
3.3 Інструкція з популяризації та підтримки сайту	53
4 Економічний розділ	56
4.1 Визначення стадій технологічного процесу та загальної тривалості проведення розробки інтернет-магазину	56
4.2 Визначення витрат на оплату праці та відрахувань на соціальні \заходи	57
4.3 Розрахунок витрат на електроенергію	59
4.4 Розрахунок суми амортизаційних відрахувань інтернет-магазину	60
4.5 Обчислення накладних витрат	60
4.6 Складання кошторису витрат та визначення собівартості інтернет-магазину	61
4.7 Розрахунок ціни інтернет-магазину	61

					<i>2026.КВР.122.423.15.00.00 ПЗ</i>		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		<i>Олещук М.І.</i>			Літ.	Арк.	Архивів
Перевір.		<i>Цимбалюк Л.В.</i>				5	
Реценз.					<i>ВСП ТФК ТНТУ КН-423 м. Тернопіль</i>		
Н. Контр.		<i>Приймак В.А.</i>					
Затверд.							

*Розробка інтернет-магазину
товарів для здорового способу
життя «Healthy Life»
Пояснювальна записка*

4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень	62
5	ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ	64
5.1	Дії роботодавця при отриманні повідомлення про нещасний випадок	64
5.2	Чинники електричного характеру, що впливають на тяжкість ураження людини електричним струмом	66
	ВИСНОВКИ.....	69
	ПЕРЕЛІК ПОСИЛАНЬ.....	71
	ДОДАТКИ.....	73
	Додаток А. Лістинг фрагменту файлу «Header.tsx».....	73
	Додаток Б. Лістинг файлу «ProductCard.jsx»	75
	Додаток В. Лістинг файлу «FilterSidebar.tsx»	78
	Додаток Г. Лістинг файлу «User.ts»	82
	Додаток Ґ. Лістинг файлу «Product.ts».....	83

АНОТАЦІЯ

Тема кваліфікаційної роботи: Розробка інтернет-магазину товарів для здорового способу життя «Healthy Life»

Головною ціллю роботи є створення зручного інтернет-магазину для популяризації здорового способу життя, реалізації екологічно чистих товарів, спортивного харчування і натуральних добавок.

Пояснювальна записка містить п'ять окремих розділів.

У першому розділі наведено аналітичний огляд наявних торгових майданчиків електронної комерції та проведено детальний аналіз технічного завдання.

Другий розділ розкриває процес створення інтернет-магазину та містить пояснення щодо вибору архітектурних рішень. Описано структуру таблиць бази даних, методи обробки вхідної інформації, наведено алгоритми роботи окремих функціональних модулів та етапи тестування написаного коду.

Спеціальна частина пояснює процедуру розгортання інтернет-магазину на хмарному сервері та містить інструкцію з експлуатації системи. Текст розділу дає вказівки для щодо керування опублікованим вмістом, картками товарів та категоріями здорового харчування

Розрахунок витрат на написання програми та оцінка фінансової доцільності проекту наведені в четвертому розділі кваліфікаційної роботи.

В останньому п'ятому розділі розглянуто питання охорони праці та техніки безпеки. Зокрема, проаналізовано порядок дій роботодавця при отриманні повідомлення про нещасний випадок на виробництві, а також досліджено чинники електричного характеру, що безпосередньо впливають на тяжкість ураження людини електричним струмом.

Загальний обсяг пояснювальної записки складає __ сторінки. До складу роботи також входить графічна частина на аркушах формату А3. В неї включено таблицю техніко-економічних показників, діаграму варіантів використання, структурну схему сайту, діаграму бази даних.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

Thesis Topic: Development of the “Healthy Life” Online Store for Healthy Lifestyle Products

The main objective of this thesis is to create a user-friendly online store to promote a healthy lifestyle and sell eco-friendly products, sports nutrition, and natural supplements.

The explanatory note consists of five separate sections.

The first section provides an analytical overview of existing e-commerce platforms and conducts a detailed analysis of the technical specifications.

The second section details the process of creating the online store and explains the selection of architectural solutions. It describes the structure of the database tables, methods for processing input data, and outlines the algorithms for individual functional modules as well as the stages of testing the written code.

This section explains the procedure for deploying an online store on a cloud server and includes instructions for operating the system. The text of this section provides guidance on managing published content, product listings, and healthy eating categories

The calculation of software development costs and the assessment of the project’s financial feasibility are presented in the fourth section of the thesis.

The final, fifth section addresses occupational health and safety issues. In particular, it analyzes the employer’s procedure for responding to a report of a workplace accident and examines electrical factors that directly influence the severity of electric shock injuries.

The total length of the explanatory note is __ pages. The thesis also includes a graphical section on A3-sized sheets. It contains a table of technical and economic indicators, a diagram of usage options, a site structure diagram, and a database diagram.

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

У сучасних умовах стрімкого зростання популярності здорового способу життя та переходу споживачів до онлайн-покупок, створення спеціалізованих веб-платформ для реалізації вітамінів та дієтичних добавок набуває особливої актуальності. Якість взаємодії стає ключовою. Через високу конкуренцію на ринку e-commerce компаніям доводиться шукати інноваційні шляхи для залучення аудиторії, пропонуючи не лише якісний товар, але й персоналізований досвід та високу швидкість обробки замовлень. Сучасний клієнт є дуже вимогливим.

Завдяки використанню сучасного фреймворку Next.js [2] з'являється унікальна можливість створювати масштабовані та високопродуктивні інтернет-магазини, які здатні витримувати значні навантаження під час проведення сезонних акцій. Гнучкість архітектури має значення. Шляхом впровадження технології серверного рендерингу (SSR) та оптимізації зображень у форматі WebP вдається значно покращити позиції сайту в пошуковій видачі Google, що безпосередньо впливає на комерційний успіх проєкту.

Для користувачів, які дбають про своє здоров'я, критично важливою є прозорість інформації про склад продукту, його походження та термін придатності. Довіра є основою всього. Реалізація функціоналу деталізованих карток товарів з повним описом та відгуками реальних покупців допомагає уникнути негативного досвіду та підвищує рівень лояльності до бренду HealthyLife. Клієнт хоче знати все.

З метою надійного захисту персональних даних клієнтів та гарантування безпеки всіх фінансових операцій у межах даної роботи було застосовано сучасні протоколи авторизації, що базуються на механізмі JSON Web Tokens (JWT). Безпека не терпить компромісів. Окрім технічної стійкості до зламів, значну увагу приділено проєктуванню інтуїтивно зрозумілого інтерфейсу, який є доступним для людей будь-якого віку та рівня цифрової грамотності. Сайт має бути простим.

Під час розробки клієнтської частини виникає нагальна потреба в ефективному управлінні станом кошика та збереженні даних про обрані товари без постійних запитів до сервера. Легкість є головним критерієм. Використання

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

сучасної бібліотеки для стейт-менеджменту Zustand дозволяє забезпечити миттєву реакцію інтерфейсу на дії користувача, що значно покращує загальне враження від роботи з платформою.

В основу збереження всієї інформації покладено гнучку нереляційну модель бази даних MongoDB, яка дозволяє легко змінювати структуру опису вітамінів чи додавати нові характеристики для спортивного харчування. Пріоритетом тут є масштабованість. Завдяки цьому адміністратор може оперативнo розширювати асортимент або оновлювати категорії без необхідності складної та довготривалої перебудови всієї архітектури бази даних. У цій кваліфікаційній роботі здійснюється розробка комплексного програмного продукту для інтернет-магазину HealthyLife, що охоплює як клієнтську, так і адміністративну частини для повного керування контентом. Проєкт поєднує багато технологій. Метою роботи визначено створення функціонального та швидкого веб-застосунку, який оптимізує процеси вибору, замовлення та доставки товарів для здорового способу життя. Зручність є головною метою.

З огляду на домінування мобільного трафіку, особливу увагу приділено повній адаптивності інтерфейсу для його коректного відображення на екранах смартфонів та планшетів. Мобільність є ключовим фактором. Оптимізація програмного коду та мінімізація розміру завантажуваних ресурсів дозволяють досягти високих показників швидкодії за стандартами Google PageSpeed, що критично важливо для утримання уваги сучасних покупців.

Завдяки впровадженню розробленого програмного забезпечення у діяльність бізнесу вдається досягти повної автоматизації процесу роздрібної торгівлі та суттєво спростити щоденну роботу менеджерів з оновлення асортименту. Це прискорює всі бізнес-процеси. Поєднання технологій Next.js та MongoDB у межах одного проєкту дозволяє досягти синергії між високою продуктивністю сервера та чудовим користувацьким досвідом, що підтверджує актуальність та практичну цінність виконаної роботи.

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

Під час проєктування спеціалізованих систем електронної комерції для сегменту здорового способу життя критично важливо дослідити архітектурні підходи вже функціонуючих рішень. Це дозволяє уникнути типових помилок. Отримані дані стають базою для формування функціональних вимог до майбутнього продукту, що забезпечує його конкурентоспроможність на сучасному ринку.

Зокрема, одним із найбільш масштабних представників галузі є платформа iHerb, яка спеціалізується на продажі вітамінів та натуральних добавок. Її головна сторінка зображена на рисунку 1.1. Система має складну фільтрацію товарів. Для обробки запитів користувачів тут застосовуються розвинені алгоритми ранжування, що враховують відгуки покупців та історію їх попередніх замовлень у магазині.

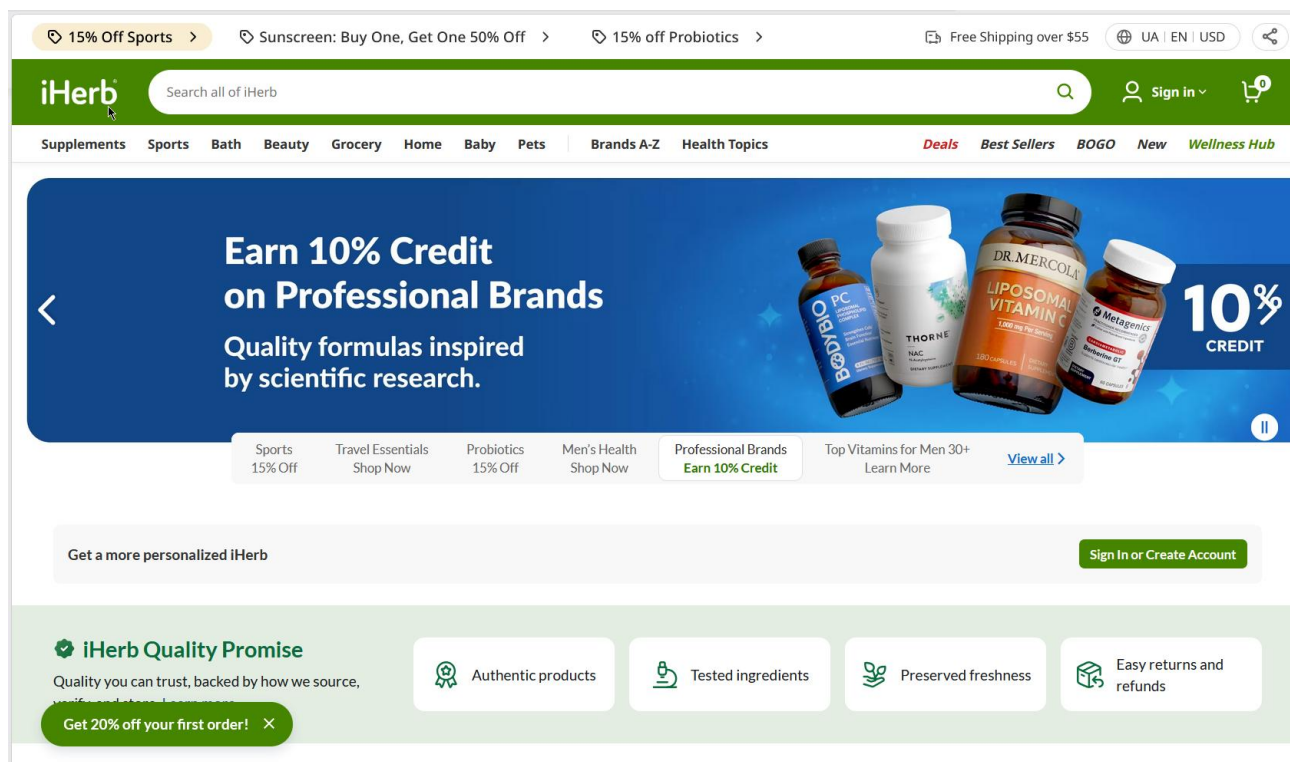


Рисунок 1.1 – Інтерфейс платформи iHerb

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

У контексті спортивного харчування варто виділити ресурс Myprotein, що демонструє приклад вузькоспеціалізованої екосистеми з акцентом на власні бренди. Її каталог товарів зображений на рисунку 1.2. На сайті реалізовано динамічне ціноутворення. Завдяки використанню сучасних CDN-сервісів платформа забезпечує високу швидкість завантаження контенту навіть при значних пікових навантаженнях під час проведення сезонних розпродажів.

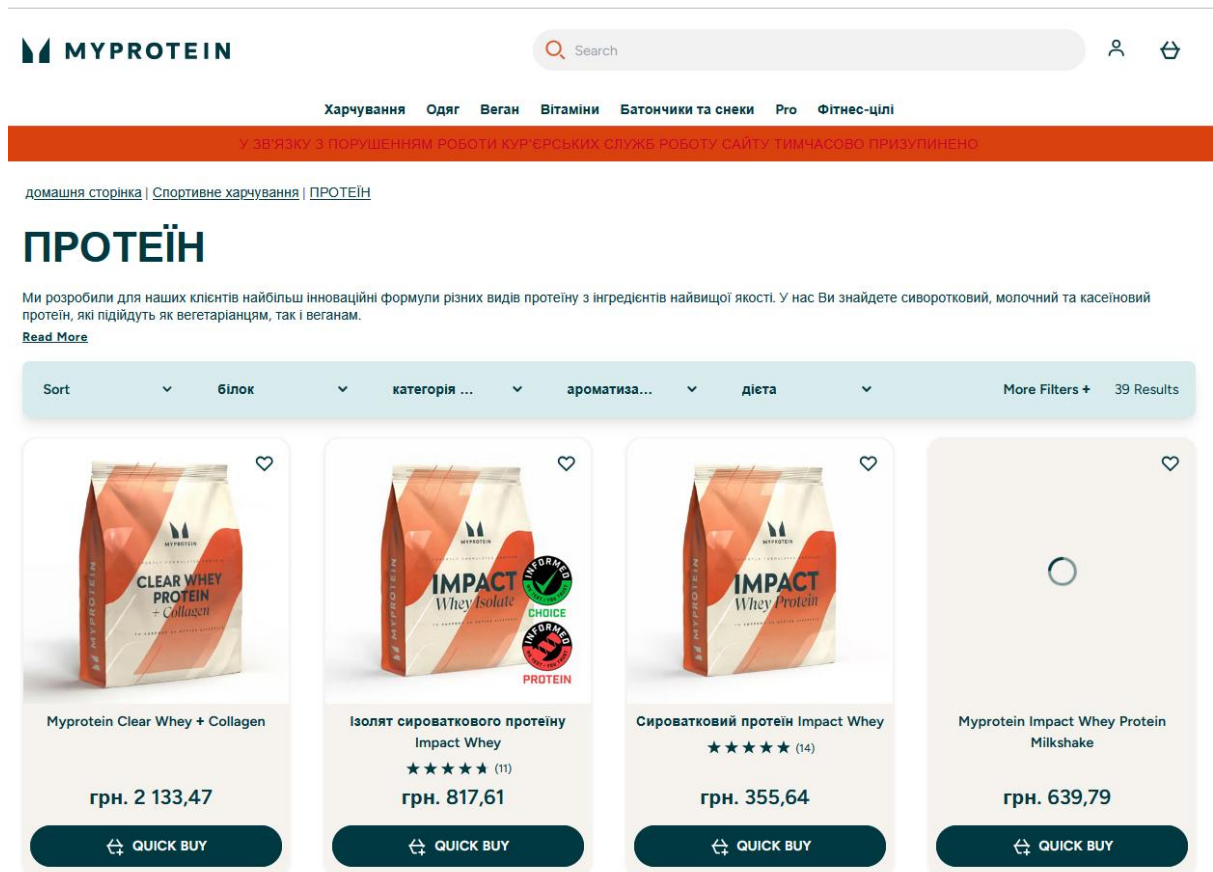


Рисунок 1.2 – Головна сторінка магазину Myprotein

Наступним об’єктом аналізу став сервіс Vitacost, котрий вирізняється інтеграцією персоналізованих рекомендацій на основі анкетування користувачів про поточний стан здоров’я. Процес покупки тут максимально спрощений. Платформа використовує модульну архітектуру фронтенду, що дозволяє швидко змінювати дизайн окремих блоків сторінки без повного перевантаження всієї програмної логіки сервісу. Головна сторінка зображена на рисунку 1.3

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

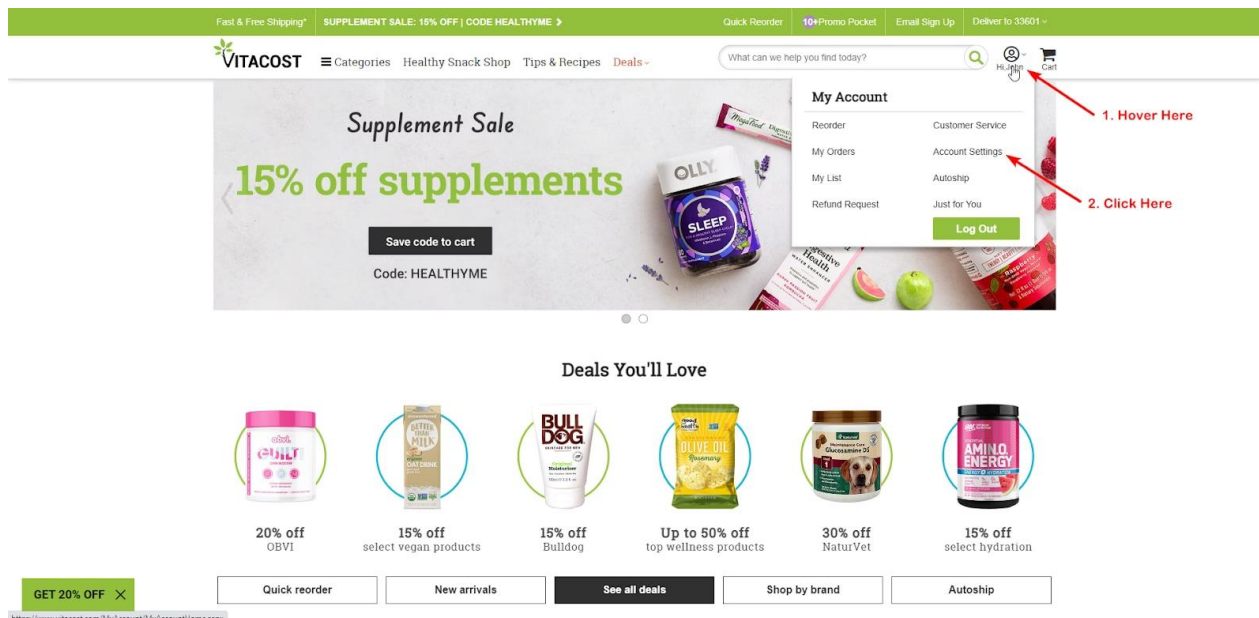


Рисунок 1.3 – Система пошуку товарів Vitacost

У таблиці 1.1 наведено порівняльні характеристики розглянутих аналогів за ключовими критеріями продуктивності та зручності для кінцевого споживача сучасних цифрових послуг.

Таблиця 1.1 – Порівняння функціональних можливостей аналогів

Критерій аналізу	iHerb	Myprotein	Vitacost
Швидкість завантаження	Висока	Середня	Висока
Програма лояльності	Багаторівнева	Реферальна	Бонусна
Мобільний додаток	Нативний	Гібридний	Відсутній
Персоналізація контенту	На основі пошуку	За категоріями	Через опитування

За результатами порівняння стає зрозуміло, що більшість систем зосереджені на масштабності каталогу, часто ігноруючи індивідуальні потреби у зручній навігації для новачків. Це створює нішу для розробок. Щоб забезпечити вищий рівень інтерактивності, було прийнято рішення будувати власну систему на

базі фреймворку Next.js із використанням нереляційної бази даних MongoDB [3]. Такий підхід гарантує гнучкість структури. Використання TypeScript дозволить мінімізувати кількість помилок під час розробки складних інтерфейсних компонентів та забезпечить стабільну роботу всього програмного комплексу.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

Під час ведення цієї роботи було створено та здано програмний код під офіційною назвою «HealthyLife — веб-платформа для продажу товарів здорового способу життя», що дозволяє ефективно керувати списком товарних одиниць у режимі реального часу. Забезпечено повний цикл обробки даних.

Для швидкого вирішення завдань цей продукт діє як сайт, що дає людям змогу легко шукати, дивитись опис та брати корисні речі для тіла, забезпечуючи високу швидкість завантаження сторінок завдяки використанню сучасних методів рендерингу на стороні сервера. В основу коду покладено стек Next.js.

Зокрема, сфера дії даної системи охоплює світ онлайн-продажів, торгівлі вітамінами та їжею для спорту через мережу для всіх груп осіб, дозволяючи значно знизити витрати часу на пошук необхідних сертифікованих добавок від перевірених світових виробників. Автоматизовано рух усіх нових даних.

Щоб дістати гарний результат, для роботи слід мати ПК, ноутбук або гаджет, що має доступ до мережі та вміє читати нові типи сайтів, оскільки інтерфейс розроблено з урахуванням усіх вимог до адаптивності та зручності використання. Завдяки адаптивності сайт працює на всіх типах екранів.

1.2.2 Призначення розробки

Для того щоб дати людям кращий та дуже швидкий шлях до вибору гарних ліків, добавок або засобів гігієни, головною метою даної роботи став запуск нової

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

веб-системи для роботи у мережі. Це робить процес купівлі простішим. Шляхом впровадження сучасних програмних кодів на базі фреймворку Next.js вдалося досягти стабільно високої швидкості обробки всіх клієнтських запитів та гарантувати повну стійкість графічного інтерфейсу сайту навіть під час дуже великих навантажень. В основу покладено принципи швидкої роботи.

Щоб гарантувати виконання всіх поставлених бізнес-планів, у межах даного проєкту було створено та протестовано такий список основних робочих функцій майбутньої системи:

- з метою надання клієнтам повної інформації було розроблено гнучкий каталог товарів з можливостями миттєвого пошуку за поточною ціною, типом продукту, країною походження або назвою конкретного бренду;
- через використання потужної бази даних MongoDB забезпечено швидке зберігання всіх даних про склад кожної позиції, термін дії, наявність на складі та поради фахівців щодо правильного вживання;
- завдяки впровадженню нових схем захисту інформації реалізовано особистий кабінет, де надійно зберігається вся історія попередніх замовлень, бонуси, адреси доставки та вибрані налаштування для входу.

В результаті реального запуску створеної платформи досягається повна автоматизація всього старого циклу роздрібних продажів, що включає постійний контроль залишків на складі та дуже швидке звітування перед власником або менеджером магазину. Це робить бізнес більш чесним. За рахунок використання суворої мови TypeScript під час написання логічних компонентів вдалося повністю уникнути появи частих помилок при обробці складних даних, які вводять у форми кожен новий юзер сайту. Тепер сайт працює дуже стабільно.

1.2.3 Вимоги до функціоналу web-сайту

Під час формування архітектури веб-платформи HealthyLife було розроблено перелік вимог до функцій, що забезпечують повний цикл автоматизованого продажу товарів для здоров'я. Це стало основою проєкту.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

Завдяки впровадженню чіткої рольової моделі доступу система гарантує поділ прав користувачів та захищає внутрішню частину сайту від несанкціонованого входу сторонніх осіб. Безпека є пріоритетом тут.

Для створення комфортних умов під час здійснення покупок у системі передбачено роботу трьох основних груп відвідувачів з різним рівнем доступу до даних.

– Гість без реєстрації отримує можливість вільно переглядати каталог товарів, використовувати інструменти для пошуку за ціною чи брендом та наповнювати кошик перед оплатою.

– Клієнт після входу в систему може змінювати дані у профілі, бачити список усіх своїх минулих замовлень та додавати вибрані продукти до списку бажаного.

– Адміністратор або менеджер сайту здійснює повне керування списком товарів через інтерфейс панелі керування, контролює залишки на складі та оновлює статус поточної доставки.

Усі дані оновлюються миттєво.

У процесі розробки графічного інтерфейсу було визначено структуру основних сторінок, які створюють логічний шлях покупця від вибору продукту до успішного завершення замовлення.

– Головна сторінка містить блоки з новими акціями, перелік популярних категорій та блоки з корисною інформацією про здоровий спосіб життя для залучення нових клієнтів.

– Сторінка каталогу забезпечує вивід карток товарів із боковою панеллю для швидкої фільтрації за ціною, формою випуску продукту, країною походження чи конкретною групою вітамінів.

– Картка товару надає повний опис кожної позиції, включаючи фото у високій якості, термін придатності, склад однієї дози та детальні поради щодо правил прийому.

– Кошик користувача дозволяє легко змінювати кількість одиниць

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

продукту перед оплатою та автоматично рахує загальну суму замовлення з урахуванням усіх діючих знижок на сайті.

– Сторінка оформлення замовлення містить форми для введення телефону, вибору поштового відділення чи способу оплати, а також можливість ввести промокод для отримання додаткового бонусу.

Детальна схема взаємодії різних груп відвідувачів із функціональними блоками інтернет-магазину представлена у вигляді UML-діаграми варіантів використання у графічній частині кваліфікаційного проєкту 2026.КВР.122.423.15.00.00 ДВ.

Для стабільної взаємодії програмних частин вхідними даними виступають параметри пошуку, дані для входу в кабінет та об'єкти замовлень, що передаються у форматі JSON. Дані йдуть через API. Вихідна інформація подається у вигляді динамічних сторінок магазину, актуальних даних про наявність вітамінів на складі та системних повідомлень про стан виконання фінансової транзакції.

З метою надання високого рівня надійності було впроваджено контроль за правильністю вводу всіх даних на стороні сервера за допомогою бібліотек для валідації. Це виключає критичні збої. Захист від злому базується на використанні захищених токенів, що дозволяє мінімізувати ризик втрати персональних даних під час роботи клієнтів у відкритих мережах доступу.

У сучасних умовах сайт повинен стабільно працювати у популярних веб-браузерах на базі двигуна Chromium, а також у програмах Safari та Mozilla Firefox. Сумісність є дуже важливою. Обрана конфігурація сервера має тримати навантаження від одночасної роботи великої кількості людей без втрати швидкості завантаження контенту в пікові години продажів.

Згідно з правилами гарного тону розробки, час завантаження перших елементів на екрані не повинен перевищувати дві секунди навіть при низькій швидкості мобільного інтернету. Швидкість дає кращий досвід. Обробка складних запитів до бази MongoDB, пов'язаних із вибором товарів серед тисяч позицій, має тривати не довше ніж 300-400 мілісекунд.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.4 Вимоги до програмної документації

Для супроводу та подальшого масштабування створеного веб-ресурсу HealthyLife слід розробити повний пакет технічних паперів, які дадуть чітке уявлення про внутрішню побудову програмного коду. Це спрощує життя розробникам. У межах даного проєкту документація має включати детальний опис усіх системних зв'язків та логіку взаємодії фронтенду з базою даних для швидкого виявлення помилок. Якісні тексти допомагають підтримці.

Під час формування фінального звіту до складу обов'язкової документації мають увійти наступні матеріали, що описують ключові аспекти роботи інтернет-магазину:

- Повний опис архітектури системи на базі фреймворку Next.js, де детально вказано призначення кожної папки та принципи роботи серверних компонентів для швидкого рендерингу сторінок.
- Детальна схема структури нереляційної бази даних MongoDB з описом усіх типів полів у колекціях, що відповідають за зберігання складних описів вітамінів та спортивних добавок.
- Технічна інструкція з розгортання проєкту на хмарному сервері з переліком усіх змінних середовища, які потрібні для підключення до платіжних шлюзів та бази.
- Керівництво для адміністратора магазину, яке крок за кроком пояснює процес додавання нових товарів через графічний інтерфейс та правила редагування поточних цін у каталозі.
- Список усіх точок доступу API з описом вхідних параметрів та форматів відповіді сервера для забезпечення стабільного зв'язку між різними модулями сайту.

Усі розділи мають бути чіткими.

Зокрема, під час написання вихідного коду на мові TypeScript обов'язково треба ставити короткі коментарі до кожної складної функції або окремого

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

інтерфейсу даних. Це робить логіку зрозумілою. Використання міжнародних стандартів JSDoc дозволяє автоматично створювати довідку про всі методи, які обробляють замовлення клієнтів або рахують фінальну суму знижки у кошику покупця. Такий підхід економить час.

Щоб забезпечити коректну роботу серверної частини, необхідно підготувати специфікацію всіх ендпоінтів у форматі Swagger, де вказано типи запитів та структуру об'єктів у форматі JSON. Документ допомагає стороннім сервісам. Наприкінці роботи слід додати результати автоматичного тестування системи, щоб підтвердити відсутність багів у найбільш важливих частинах інтерфейсу та стабільність зв'язку з хмарним сервером. Тести гарантують якість роботи.

Для кінцевих користувачів сайту слід підготувати лаконічну інструкцію, яка описує процес реєстрації, правила використання особистого кабінету та способи швидкого зв'язку з технічною підтримкою. Текст має бути дуже простим. Окремий розділ документації повинен містити рекомендації щодо регулярного створення копій бази даних для запобігання втраті інформації про клієнтів та історію їхніх покупок. Безпека даних є пріоритетом.

1.2.5 Техніко-економічні показники

Під час роботи над проєктом HealthyLife було проведено повний розрахунок ресурсів для оцінки вартості та термінів праці над усіма етапами розробки. Це робить план дуже точним. Шляхом вивчення обсягу майбутніх завдань вдалося визначити кількість часу, що знадобиться для написання коду та налаштування серверної частини для зберігання даних про вітаміни. Усі отримані цифри є вірними.

Для створення якісного коду у межах даної кваліфікаційної роботи було заплановано виконання наступного списку важливих завдань, що потребують значних зусиль від автора:

- З метою глибокого вивчення ринку було виконано повний аналіз потреб

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

клієнтів та сформовано основні технічні вимоги до функцій майбутньої веб-платформи.

– Для побудови гарної бази даних розроблено логічні схеми в системі MongoDB та архітектуру модулів на базі сучасного фреймворку Next.js.

– У межах безпосередньої праці над кодом створено логіку для обробки кошика замовлень та графічний інтерфейс користувача з активним використанням мови TypeScript.

– Щоб гарантувати повну безпеку даних, впроваджено механізми перевірки токенів доступу та захисту профілів покупців від можливого зламу сторонніми особами.

– Для підтвердження високої якості коду проведено повний цикл тестування функцій та оперативно виправлено всі виявлені недоліки в роботі системи.

– На завершальному етапі робіт було підготовлено повний комплект технічних паперів для подальшого супроводу та дуже швидкого оновлення всіх компонентів сайту.

У підсумку загальний обсяг витрат живої праці на цей складний проєкт складає приблизно 2.2–3.2 людино-місяця за умови роботи одного кваліфікованого інженера. Це відповідає нормам розробки. Зокрема, на створення візуальних частин сайту та налаштування адаптивних сторінок за допомогою Tailwind CSS було витрачено біля 1.1 людино-місяця активної праці. Такий темп є цілком стабільним. На розробку логіки API та зв'язок з хмарною базою даних пішло ще близько 1.4 людино-місяця дуже інтенсивної та складної розумової діяльності. Весь бекенд працює дуже швидко.

У ході виконання поставлених завдань було задіяно сучасне обладнання та спеціальне програмне забезпечення, що дозволило значно прискорити процес написання окремих модулів системи. Це підвищило загальну продуктивність роботи. Підраховано, що витрати цінного машинного часу на розробку HealthyLife склали орієнтовно 130–175 годин інтенсивної роботи техніки під повним робочим

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

навантаженням. Комп'ютер працював майже без зупинок. До складу основних задіяних ресурсів під час реалізації даного проєкту увійшли наступні компоненти та перевірені інструменти для написання коду:

- Для написання програмного коду було використано потужний персональний комп'ютер та середовище Visual Studio Code з великим набором корисних плагінів.
- З метою перевірки отриманих результатів застосовано локальний сервер Node.js та всі популярні веб-браузери для тестування адаптивності сторінок магазину.
- Через мережу Інтернет здійснювався постійний доступ до хмарної бази даних MongoDB Atlas та надійних сервісів для розміщення коду готового сайту.
- Для контролю всіх версій коду та спільної роботи над модулями було задіяно систему Git разом із безкоштовним сервісом GitHub.

Результати техніко-економічних показників проєкту наведено у графічній частині 2026.КВР.122.423.15.01.00 ТБ.

1.2.6 Стадії та етапи розробки

Щоб отримати якісний програмний код та надати повну відповідь усім вимогам сучасного бізнесу у сфері продажів товарів для здоров'я, розробка сайту HealthyLife ділиться на декілька великих кроків, що йдуть суворо послідовно один за одним. Це дає нам дуже чіткий план. Під час ведення робіт кожен окремий етап має свій чіткий результат, який згодом стає надійним початком для наступних складних дій розробника у межах створення повноцінного та стабільного інтернет-магазину для всіх груп покупців. В основу розробки покладено логіку.

Для досягнення успіху було визначено наступний перелік стадій реалізації проєкту:

- 1) Дослідження та аналіз потреб. На самому початку шляху було проведено дуже глибокий аналіз ринку товарів для тіла та ретельно вивчено основні потреби майбутніх клієнтів, які будуть щоденно користуватися цією

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

системою для пошуку необхідних вітамінів або корисних добавок. Отримані результати аналізу лягли в основу. Шляхом вивчення досвіду великих конкурентів вдалося зібрати повний список необхідних функцій, які включають швидкий пошук потрібних товарів, зручну роботу з кошиком та безпечний вхід до особистого кабінету через захищені канали зв'язку. Так було створено детальне ТЗ.

2) Побудова архітектури та бази даних. З метою побудови надійної структури даних було виконано етап проєктування схем для хмарної бази MongoDB та загальної архітектури всіх програмних модулів на базі сучасного фреймворку Next.js з використанням методів серверного рендерингу. Це значно економить ресурси розробки. Зокрема, було детально описано всі зв'язки між типами товарів, замовленнями клієнтів та профілями адміністраторів, щоб гарантувати високу швидкість отримання інформації через сучасні API-запити до сервера у форматі JSON. Вибрано кращий набір технічних інструментів.

3) . Програмування основних компонентів. Безпосередньо під час стадії написання програмного коду було створено логічні компоненти на мові TypeScript та реалізовано адаптивні макети сторінок магазину з використанням потужної бібліотеки Tailwind CSS для стилізації всіх елементів графічного інтерфейсу. Цей код став надійною основою платформи. Завдяки впровадженню серверних функцій Next.js вдалося досягти майже миттєвого відображення карток товарів, що значно підвищує зручність користування сайтом HealthyLife для людей навіть з дуже низькою швидкістю мобільного інтернету у дорозі. Вся система працює дуже швидко.

4) . Перевірка якості та налагодження. Зокрема, для підтвердження високої якості розроблених модулів було проведено серію складних тестів, які охоплюють перевірку безпеки авторизації та повну коректність проведення всіх фінансових транзакцій майбутніми покупцями під час оплати кошика товарів. Така стабільність є головним пріоритетом тут. Шляхом виправлення виявлених дрібних недоліків на фінальній стадії тестування вдалося гарантувати повну відсутність будь-яких збоїв у роботі фільтрів каталогу та форми швидкого зворотного зв'язку з

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

менеджером магазину для консультацій. Усі критичні баги були усунені.

5) Запуск та фінальна документація. В самому кінці роботи було здійснено завантаження всіх файлів проєкту на хмарний хостинг та підготовлено повний пакет супровідних інструкцій для власників даного веб-сайту з метою спрощення процесу майбутнього навчання нових працівників. Весь код тепер стабільно працює у мережі. Завдяки написанню зрозумілої документації для менеджерів було забезпечено дуже легке управління складом товарів, термінами їхньої придатності та максимально швидке оновлення цін у реальному часі без залучення сторонніх фахівців з програмування. Готовий проєкт передано в експлуатацію.

1.2.7 Порядок контролю та прийому

Щоб забезпечити високу якість фінального програмного продукту HealthyLife, під час проведення контролю здійснюється глибока перевірка кожного окремого функціонального модуля на повну відповідність вимогам завдання. Це робить систему стабільною. Завдяки імітації дій справжніх людей у мережі вдається вчасно знайти та дуже швидко усунути всі дрібні збої у процесі вибору вітамінів або під час фінальної оплати. Продукт працює без помилок.

Для отримання вірних даних під час проведення тестів необхідно забезпечити наявність стабільного зв'язку з інтернетом та використовувати лише найновіші версії популярних браузерів Google Chrome або Safari. Вхід іде через хмару. За допомогою підключення до бази MongoDB Atlas розробник може швидко перевірити правильність збереження всієї інформації про нові замовлення, адреси доставки та персональні профілі кожного покупця. Дані надійно лежать там. На рисунку 1.4 представлено загальну блок-схему етапів проведення перевірки готового сайту перед його безпосереднім випуском у відкритий доступ для всіх груп юзерів.

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

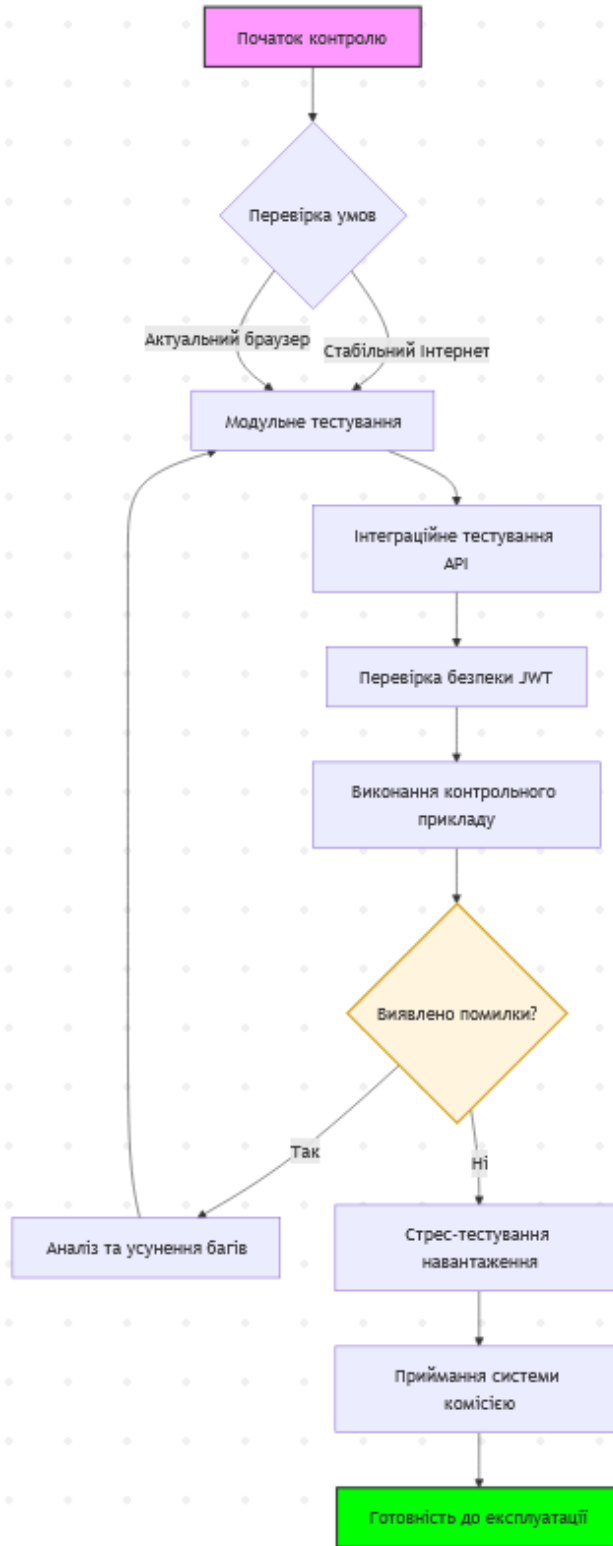


Рисунок 1.4 – Схема процесу контролю

Представлена на схемі логіка передбачає послідовне проходження від перевірки окремих функцій до комплексного випробування всієї архітектури веб-застосунку в умовах інтенсивних запитів до бази даних. Це гарантує надійність

вузлів. Зокрема, на першому етапі тестуються дрібні методи обробки вводу, тоді як фінальна стадія присвячена аналізу стабільності зв'язків між фронтендом на Next.js та хмарними сервісами зберігання інформації. Весь процес є керованим.

Шляхом імітації роботи реальної людини у межах розробленого контрольного прикладу перевіряється повний життєвий цикл купівлі корисних добавок, починаючи від реєстрації у кабінеті і закінчуючи оплатою зібраного кошика. Цей шлях є логічним. Зокрема, було успішно проведено тестування таких важливих операцій, як швидка фільтрація каталогу за брендом, зміна кількості одиниць продукту перед покупкою та перевірка отримання автоматичного сповіщення. Все працює дуже чітко. У таблиці 1.2 вказано основні набори тестів, які були використані розробником для аналізу швидкості обробки складних запитів на сервері при великих навантаженнях.

Таблиця 1.2 – Сценарії перевірки функцій

Етап перевірки	Опис дій у системі	Очікуваний результат тесту
Робота профілю	Створення кабінету та вхід за паролем	Отримання доступу та JWT токена
Пошук товарів	Вибір фільтрів за брендом та ціною	Вивід списку вітамінів з бази
Кошик	Зміна кількості та видалення позицій	Перерахунок суми до оплати
Оформлення	Ввід адреси та вибір пошти для відправки	Створення замовлення в базі

З метою надійного захисту даних під час тестів особлива увага приділяється роботі захищених токенів JWT, які повинні закрити доступ до внутрішньої адмін-панелі сайту від спроб входу звичайних відвідувачів. Це діє дуже добре. В основу адміністрування покладено право менеджера швидко додавати нові позиції, змінювати фото, редагувати опис ліків та оновлювати актуальні ціни для

інформування клієнтів про зміни. Сайт оновлюється майже миттєво. Тільки після успішного входу під роллю адміністратора стає доступною функція перевірки складських залишків та зміни статусів оплати замовлень у загальному списку через зручні форми.

Під час проведення аудиту застосовується метод чорного ящика, що дозволяє оцінити зручність всього графічного інтерфейсу без будь-якого прямого втручання розробника у внутрішню структуру програмного коду на TypeScript. Він дає гарний досвід. Щоб перевірити стійкість сервера на базі Next.js, виконуються складні стрес-тести для імітації одночасної роботи сотень людей на різних сторінках сайту з великим асортиментом вітамінів. Система тримає весь удар. Обробка складних запитів до бази даних, пов'язаних із вибором товарів серед тисяч різних позицій, має тривати не довше ніж 400 мілісекунд.

За результатами всіх перевірок складається акт прийому системи, де вказується повна відсутність будь-яких критичних помилок та готовність інтернет-магазину HealthyLife до виходу у відкриту мережу для постійного використання. Продукт готовий до роботи. Веб-платформа вважається успішно прийнятою лише після того, як кожен окремий функціональний блок показав ідеальну роботу згідно з усіма вимогами замовника, що були вказані у технічному завданні. Це гарантує успіх роботи. У разі виявлення багів здійснюється їх швидкий аналіз та повторне тестування модуля до повного підтвердження його стабільності.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури сайту і web-сторінок

Під час проєктування цифрового ресурсу HealthyLife було визначено, що цей програмний продукт належить до сегменту спеціалізованих систем електронної комерції для реалізації екологічних товарів. Головна мета полягає у створенні зручного магазину. Шляхом глибокого аналізу потреб майбутніх клієнтів вдалося сформулювати чіткий перелік вимог до інтерфейсу, що забезпечує швидкий пошук вітамінів та зручне оформлення замовлень через кошик. В основу покладено принципи доступності даних.

Для побудови логічної послідовності переходів між розділами було розроблено карту сайту, яка описує зв'язки між головною сторінкою, каталогом та особистим кабінетом користувача. Детальна схема структурна схема сторінок інтернет-магазину представлена у вигляді діаграми на графічній частині кваліфікаційної роботи 2026.КВР.122.423.15.00.00 СС.

Завдяки використанню модульного підходу під час верстки вдалося досягти високої швидкості завантаження кожної окремої сторінки незалежно від кількості активних товарів у базі даних. Це робить сайт швидким. Зокрема, у ході розробки макетів у редакторі Figma було враховано потребу в інтуїтивній навігації, щоб клієнт міг знайти потрібну групу вітамінів за два кліки. Вся логіка переходів є прозорою.

З метою візуалізації розташування основних блоків була створена структурна схема головної сторінки, де вказано зони для банерів, сітку популярних категорій та блок швидкого пошуку.

Для реалізації сучасного клієнтського досвіду було прийнято рішення використовувати стек Next.js 14, що дозволяє рендерити контент на стороні сервера для кращої індексації пошуковими роботами. Це дає гарний результат. Шляхом впровадження бібліотеки Tailwind CSS вдалося створити гнучку систему стилів, яка автоматично підлаштовує розмір шрифтів та кнопок під екрани мобільних

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

телефонів або планшетів. Адаптивність є обов'язковою тут.

На етапі створення внутрішніх розділів особливу увагу приділено сторінці каталогу, де реалізовано динамічну фільтрацію за ціною, формою випуску продукту та конкретними брендами виробників добавок. Все працює дуже плавно. Завдяки використанню стану на базі бібліотеки Zustand інформація про додані до кошика товари зберігається навіть після оновлення сторінки браузера, що запобігає втраті даних клієнта. Кошик працює без збоїв.

Шляхом побудови клієнт-серверної архітектури було забезпечено надійний зв'язок між інтерфейсом користувача та хмарною базою даних MongoDB Atlas, де зберігаються всі записи про склад вітамінів. На рисунку 2.1 наведено схему цієї взаємодії. Шлях даних є прямим.

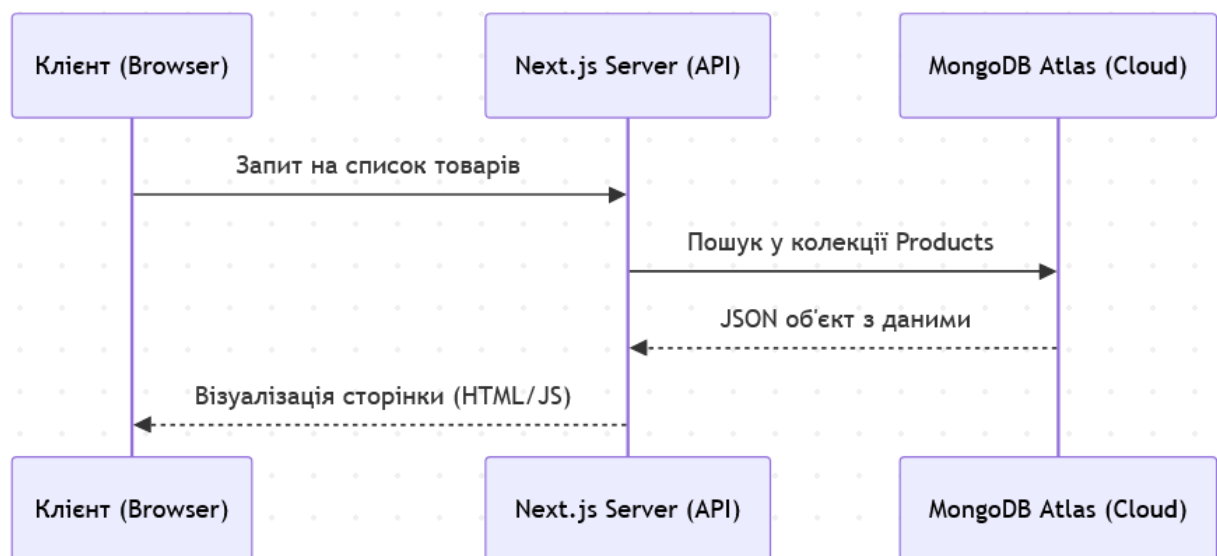


Рисунок 2.1 – Схема клієнт-серверної архітектури платформи

Під час верстки особистого кабінету було реалізовано окремі форми для редагування профілю, де введена інформація проходить сувору валідацію на стороні сервера перед її фінальним збереженням у базі. Безпека даних є пріоритетом. Через впровадження сторінки адміністратора менеджер магазину дістає можливість швидко керувати асортиментом, додаючи нові фото та описи товарів через зручний графічний інтерфейс. Це економить час працівника.

З метою підтвердження готовності структури було затверджено остаточні макети дизайну, які відповідають сучасним вимогам до чистоти інтерфейсу та зручності використання на будь-яких типах сучасних пристроїв. Робота була дуже об'ємною. У підсумку розроблена карта сайту та логіка веб-сторінок дозволяють системі HealthyLife стабільно функціонувати та легко розширюватися при додаванні нових груп товарів у майбутньому. Платформа готова до роботи.

2.2 Створення та верстка сторінок сайту

Для зміни макетів у новий код було зроблено повний цикл верстки всіх частин сайту HealthyLife з метою запуску магазину. Робота велась дуже активно. Шляхом впровадження Next.js 14 вдалося дати базу для швидкої роботи сторінок навіть при великій черзі фотографій вітамінів у каталозі магазину. В основу покладено стек TypeScript. Завдяки використанню Tailwind CSS [7] було створено гнучкий дизайн, який дозволяє кожному елементу на екрані миттєво змінювати свій розмір під час повороту смартфона або планшета. Код є дуже чистим.

Зокрема, для побудови зручної навігації було створено компонент Header, який закріплений у верхній частині екрана для миттєвого доступу до пошуку товарів. Він виглядає дуже стильно. Завдяки впровадженню бібліотеки Lucide-react інтерфейс отримав набір чітких іконок, що дозволяє людям легко знаходити розділи з вітамінами, мінералами або корисними батончиками. На рисунку 2.2 показано вигляд панелі.

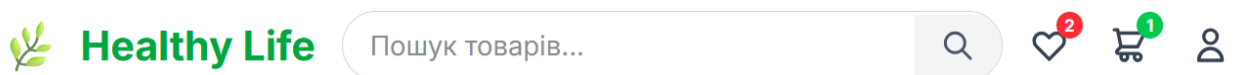


Рисунок 2.2 – Навігаційна панель магазину HealthyLife

Код компоненту Header поданий в додатку А.

На сторінці каталогу ключовим елементом виступає картка товару ProductCard, яка містить детальну інформацію про ціну, склад однієї дози та

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

актуальну наявність. Дані беруться з бази. Через використання утиліт Tailwind CSS вдалося реалізувати ефект підсвічування при наведенні курсора, що значно покращує загальний досвід користувача під час вибору добавок. На рисунку 2.2 зображено вигляд компоненту. Код компоненту ProductCard подайний в додатку Б.



Рисунок 2.2 – Дизайн картки вітамінів

Щоб полегшити пошук потрібних позицій серед тисяч записів, було розроблено бокову панель складних фільтрів із динамічним оновленням результатів через параметри рядка запиту. Це дуже зручна функція. Завдяки інтеграції з базою MongoDB Atlas система миттєво відсікає непотрібні бренди або занадто дорогі позиції, показуючи клієнту лише те, що він хоче купити. Приклад панелі наведено на рисунку 2.3. Лістинг компоненту панелі фільтрів подано в додатку В.

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Фільтри

Ціна (грн)

Від - До

Застосувати ціну

Категорії

- Органічні Продукти
- Вітаміни та Мінерали
- Спортивне Харчування
- Еко-товари
- Суперфуди
- Натуральна Косметика

Скинути фільтри

Рисунок 2.3 – Панель фільтрації каталогу

Під час реалізації доступу до персональних даних було створено модальне вікно AuthModal, де відбувається швидка перевірка пошти та надійного пароля для входу. Питання захисту є пріоритетом. Шляхом використання бібліотеки Radix UI вдалося досягти повної доступності інтерфейсу, що гарантує коректну роботу форми вводу навіть на старих версіях мобільних браузерів. На рисунку 2.4 відображено вікно входу.

Вхід

Ласкаво просимо до Healthy Life

Email або Телефон

Пароль

Увійти

Немає акаунту? [Зареєструватися](#)

Рисунок 2.4– Модальне вікно авторизації

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

В основу керування покупками покладено компонент ShoppingCart, який надійно зберігає всі вибрані позиції у пам'яті браузера до моменту їхньої повної оплати. Цей модуль працює швидко. За рахунок впровадження сховища Zustand зміна кількості упаковок або повне видалення товару зі списку відбувається без жодного перезавантаження всієї сторінки сайту. Весь огляд кошика подано на рисунку 2.5.

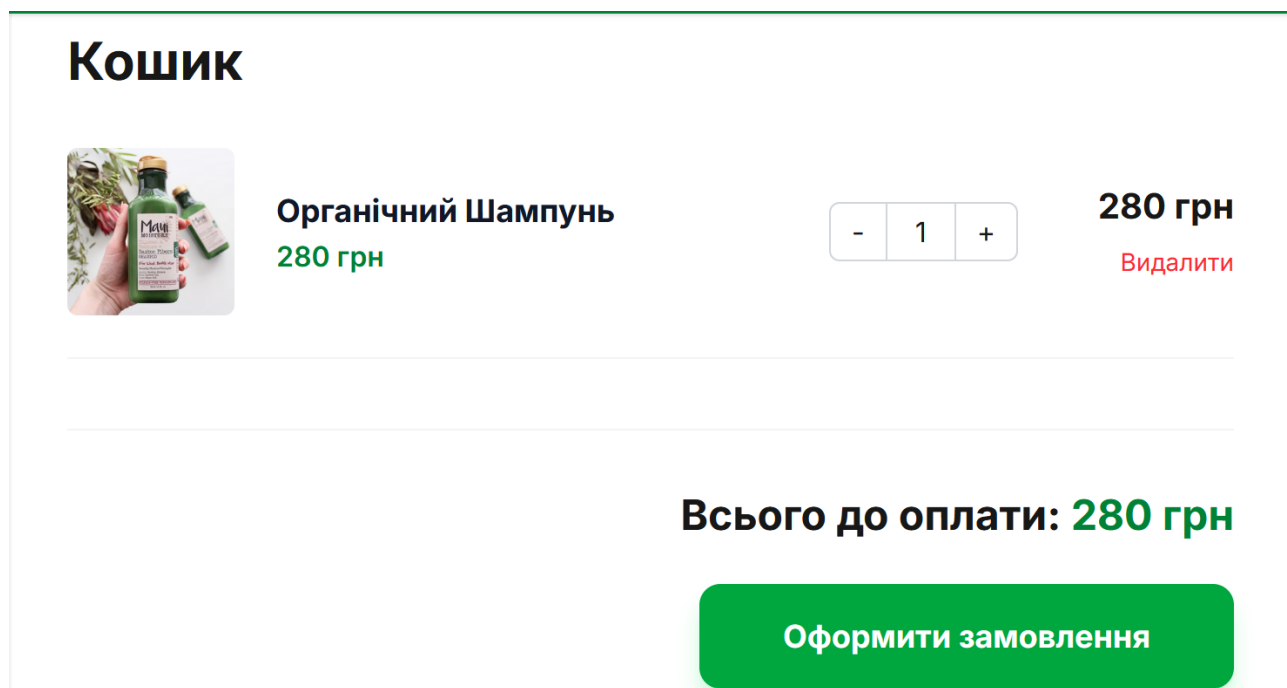


Рисунок 2.5 – Інтерфейс кошика користувача

На цьому рисунку чітко видно список доданих до кошика товарів із зазначенням їх кількості, ціни за одиницю та підсумкової вартості всього замовлення. По кліку на кнопку «Оформити замовленн» відбудеться перехід на блок оформлення замовлення, де користувач може ввести свої дані. Це робить процес оформлення замовлення інтуїтивно зрозумілим і максимально швидким.

Для оперативного контролю залишків було зверстано адміністративну панель Dashboard, яка надає менеджеру зручну таблицю для редагування цін та описів нових типів добавок. Тут менеджер бачить зміни. Через використання сучасних бібліотек для побудови графіків у системі реалізовано блок візуальної статистики, що відображає динаміку продажів різних категорій товарів за тиждень.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

Графічна схема панелі на рисунку 2.6.

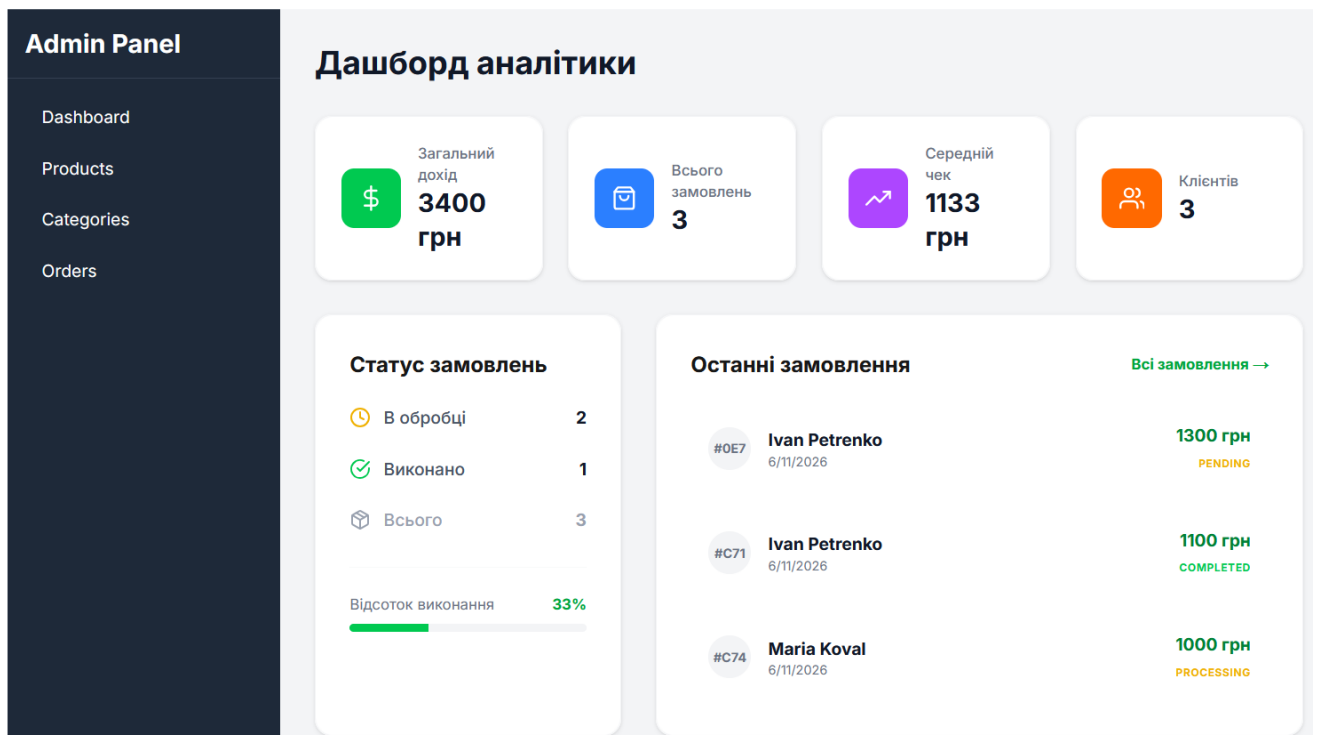


Рисунок 2.6 – Робоче місце адміністратора

У ході виконання робіт було забезпечено повну адаптивність усіх зверстаних блоків для їх ідеального відображення на екранах смартфонів, ноутбуків або великих моніторів. Інтерфейс виглядає дуже професійно. Завдяки використанню суворої типізації мови TypeScript вдалося повністю уникнути проблем із передачею некоректних даних між окремими функціональними модулями всього інтернет-магазину HealthyLife. Весь програмний продукт готовий.

2.3 Розробка структури бази даних сайту

Під час безпосереднього проєктування цифрового ресурсу HealthyLife було розроблено логічну структуру бази даних, яка відповідає за надійне збереження відомостей про зареєстрованих користувачів, повний каталог наявних товарів для здоров'я та детальну історію всіх фінансових транзакцій у магазині. Це дозволяє

уникнути помилок. Шляхом впровадження нереляційної моделі на базі сервісу MongoDB Atlas вдалося досягти високої гнучкості схем, що дає змогу додавати нові характеристики для специфічних добавок без необхідності повної зупинки роботи сервера або проведення складних міграцій таблиць.

Для забезпечення високої продуктивності системи під час обробки великої кількості запитів було створено набір колекцій, що описують основні сутності нашого інтернет-магазину та їхні внутрішні параметри. Дані зберігаються дуже надійно. Зокрема, у ході розробки архітектури було виділено наступні ключові колекції документів [13], які формують інформаційну базу програмного комплексу та забезпечують цілісність усіх збережених записів про клієнтів:

- Колекція `users` (користувачі) призначена для зберігання персональних даних та параметрів доступу осіб до особистого кабінету, що включає унікальний ідентифікатор `_id`, ім'я клієнта, електронну пошту, номер телефону, зашифрований пароль та роль у системі для розмежування прав. Безпека є пріоритетом. Лістинг коду файлу з моделлю users подана в додатку Г.

- Колекція `products` (товари) містить повний перелік вітамінів та добавок із детальним описом їхніх властивостей, поточною ціною, кількістю залишків на складі, масивом посилань на зображення та зв'язком із конкретною категорією через поле `categoryId`. Опис товарів є повним. Лістинг коду файлу з моделлю users подана в додатку Г.

- Колекція `categories` (категорії) використовується для логічного групування асортименту магазину за типами продуктів, що полегшує процес пошуку через назву та унікальний текстовий індекс `slug` для формування людино-зрозумілих посилань у браузері. Це покращує досвід клієнта.

- Колекція `orders` (замовлення) забезпечує фіксацію кожного факту купівлі, зберігаючи посилання на покупця, масив об'єктів із товарами за ціною на момент оплати, загальну суму чека, повну адресу доставки та поточний статус виконання операції менеджером. Весь рух грошей фіксується.

- Колекція `reviews` (відгуки) дозволяє покупцям залишати оцінки та

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

коментарі до кожної позиції в каталозі, що реалізовано через поля ідентифікатора продукту, автора відгуку, числового рейтингу за п'ятибальною шкалою та дати створення запису. Рейтинг формує довіру людей.

– Колекції `carts` та `wishlists` (кошики та списки бажань) слугують для тимчасового зберігання вибраних користувачем позицій, що забезпечує збереження стану покупок між різними сесіями відвідування сайту HealthyLife через прив'язку до `userId`. Це робить сервіс зручним.

З метою візуалізації всіх логічних зв'язків між описаними документами була побудована ER-діаграма, яка демонструє типи відносин між користувачами, замовленнями та товарами у межах обраного технологічного стеку розробки, вона зображена на плакаті 2026.КВР.122.423.15.00.00 ДБ.

Для реалізації стабільної взаємодії між серверним кодом та хмарним сховищем даних у проєкті було використано бібліотеку Mongoose, яка дозволяє встановлювати суворі типи полів та виконувати автоматичну валідацію вводу перед збереженням. Усі схеми працюють чітко. Шляхом налаштування унікальних індексів для полів електронної пошти та номерів телефонів вдалося повністю виключити можливість дублювання облікових записів або появи некоректних зв'язків у процесі інтенсивної роботи магазину. Це гарантує чистоту даних.

У результаті проведених робіт було сформовано сучасну та дуже гнучку архітектуру збереження інформації, яка дозволяє системі HealthyLife стабільно працювати під великим навантаженням та забезпечує високу швидкість отримання результатів фільтрації товарів у каталозі.

Проектування бази повністю завершене. Завдяки обраному підходу до моделювання даних розробник може легко додавати нові функціональні блоки, наприклад систему бонусів або підписку на товари, не змінюючи при цьому базову логіку взаємодії вже існуючих модулів. Робота виконана на високому рівні.

2.4 Програмування сайту

Під час програмної реалізації веб-платформи HealthyLife було використано

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

сучасну клієнт-серверну архітектуру, що базується на поділі логіки відображення даних та процесів їхньої обробки на стороні сервера. Це забезпечує високу швидкість. Шляхом поєднання можливостей фреймворку Next.js 14 та нереляційної бази даних MongoDB Atlas вдалося створити надійне середовище для стабільного функціонування інтернет-магазину в умовах інтенсивного трафіку. В основу покладено модульність.

Для об'єднання всіх технологій у єдиний програмний комплекс розробка велася в інтегрованому середовищі Visual Studio Code з використанням пакетного менеджера npm для керування зовнішніми бібліотеками.

Весь код є валідованим. Зокрема, у ході ініціалізації проєкту було встановлено та налаштовано наступний набір інструментів, що формують технологічний фундамент нашої майбутньої системи для реалізації товарів для здоров'я:

1) Next.js 14 — головний фреймворк на базі React для побудови маршрутизації та виконання серверного рендерингу сторінок каталогу.

2) TypeScript — сувора мова програмування для мінімізації помилок під час передачі складних об'єктів даних між модулями.

3) Zustand — легка бібліотека для управління глобальним станом сайту, що відповідає за збереження товарів у кошику клієнта.

4) Tailwind CSS — утилітарний інструмент для швидкої стилізації інтерфейсу без написання довгих файлів зі стилями у форматі CSS.

5) Axios — бібліотека для виконання асинхронних HTTP-запитів до API-ендпоінтів сервера [12] для отримання актуальних цін та описів вітамінів.

Процес налаштування пройшов успішно.

Загальна схема програмної логіки та взаємодії між основними компонентами системи представлена на рисунку 2.7 у вигляді діаграми послідовності дій користувача. Шлях даних є прозорим.

									Арк.
									36
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.423.15.00.00 ПЗ				

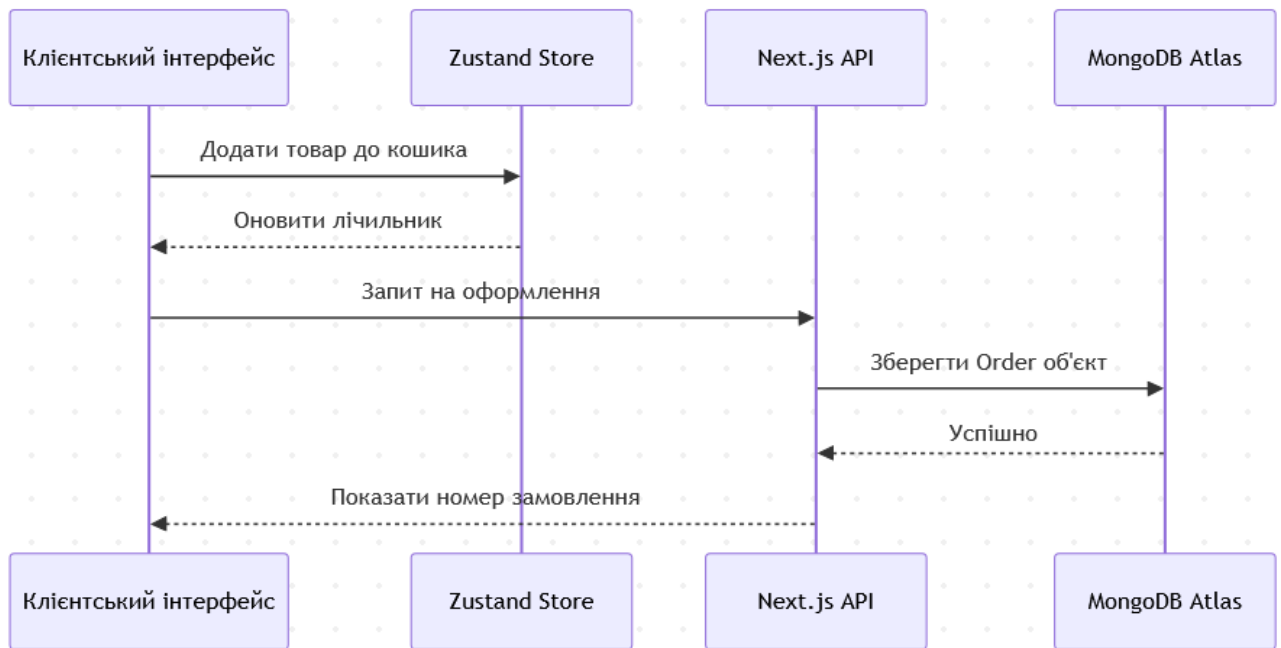


Рисунок 2.7 – Схема взаємодії програмних модулів

2.4.1 Написання клієнтської частини

Для створення зручного інтерфейсу у межах клієнтської частини було реалізовано набір компонентів, що забезпечують динамічне відображення товарів та обробку всіх дій відвідувачів сайту. Робота велась у VS Code. Шляхом використання архітектури App Router [9] у Next.js вдалося налаштувати швидку навігацію між розділами вітамінів, добавок та засобів гігієни без повного перезавантаження сторінки браузера. Це покращує досвід клієнта.

З метою організації отримання даних із хмари було розроблено серію асинхронних методів, які звертаються до внутрішніх API-маршрутів платформи для завантаження актуального списку продукції HealthyLife. Код працює дуже стабільно. Зокрема, у лістингу 2.4 наведено приклад функції для отримання відфільтрованих товарів, яка використовується на сторінці головного каталогу магазину.

```

import axios from 'axios';

export const fetchProducts = async (category?: string, sort?: string) => {
  try {
    const response = await axios.get('/api/products', {
      params: { category, sort }
    });
    return response.data;
  } catch (error) {
    console.error('Помилка завантаження даних:', error);
    return [];
  }
};

```

Лістинг 2.4 – Функція завантаження товарів з API:

Дана функція виконує GET-запит до сервера, передаючи параметри категорії та сортування у рядку запиту, після чого повертає масив JSON-об’єктів для подальшого рендерингу. Отримані дані миттєво обробляються. Завдяки впровадженню React Server Components більшість статичної інформації про вітаміни завантажується безпосередньо на сервері, що зменшує обсяг JavaScript-коду [1], який має завантажити смартфон покупця. Швидкість сайту є вражаючою.

У ході програмування логіки кошика було застосовано бібліотеку Zustand із налаштованим middleware для збереження вибору людини у локальній пам’яті браузера (LocalStorage) [5]. Дані не зникають ніколи. Це дозволяє користувачеві повернутися до покупки навіть після закриття вкладки сайту, забезпечуючи високу лояльність аудиторії до сервісу HealthyLife. Весь механізм працює автономно. На рисунку 2.8 показано візуальну структуру головних клієнтських модулів та їхні взаємозв’язки у системі.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

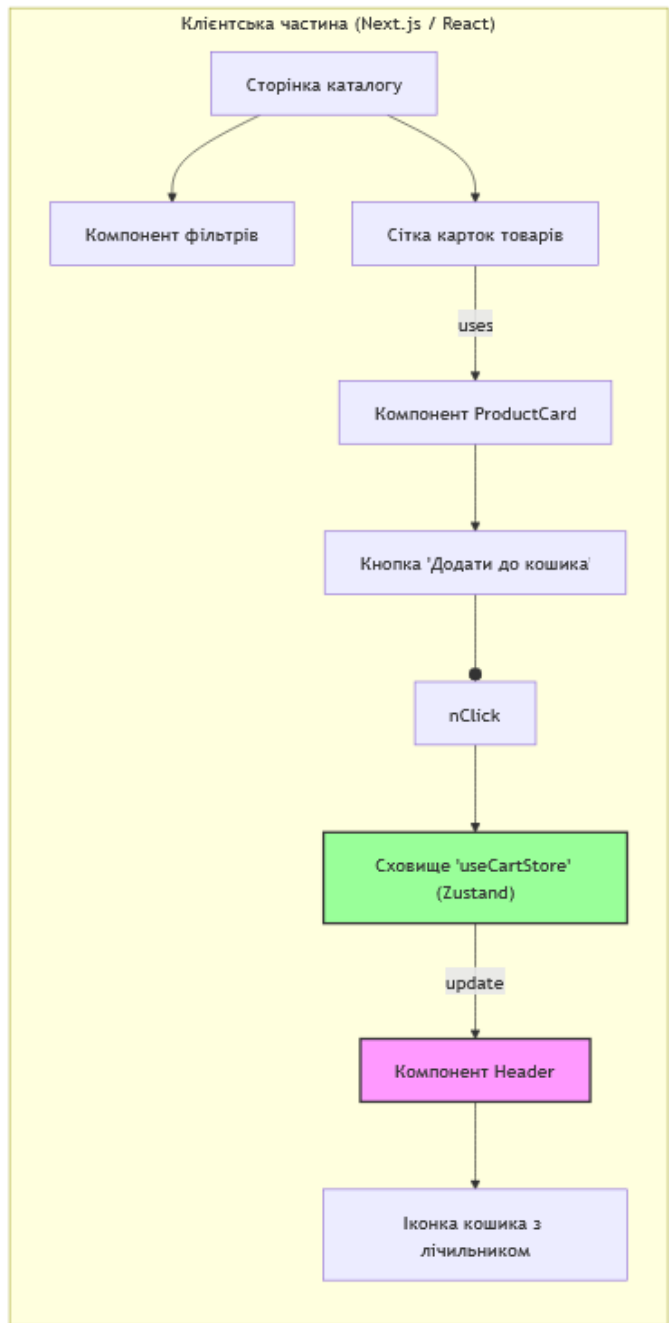


Рисунок 2.8 – Структура модулів клієнтської частини

Завдяки використанню Tailwind CSS стилізація кожного окремого елемента, від кнопки покупки до картки товару, була виконана через опис класів безпосередньо у файлах компонентів React. Це економить час розробки. Шляхом впровадження мобільної адаптивності вдалося гарантувати коректне відображення всіх форм та фільтрів на пристроях із різною шириною екрана від малих смартфонів до великих моніторів. Сайт виглядає дуже гарно.

2.4.2 Написання admin-частини

Для реалізації функцій керування магазином HealthyLife була створена окрема адміністративна частина, яка надає менеджеру повний контроль над асортиментом товарів та історією замовлень клієнтів. Доступ є суворо обмеженим. В основу захисту покладено механізм перевірки ролей на стороні сервера, що запобігає несанкціонованому входу звичайних користувачів до панелі редагування цін або залишків. Безпека працює надійно.

Шляхом використання вбудованих засобів Next.js було реалізовано спеціальні middleware-функції, які перевіряють наявність дійсного JWT-токена [8] у заголовках кожного запиту до секретних розділів сайту. Тільки адмін бачить дані. Зокрема, у лістингу 2.5 представлено фрагмент коду для оновлення статусу замовлення, що дозволяє менеджеру миттєво інформувати покупця про відправку посилки.

```
export async function updateOrderStatus(orderId: string, status: string) {
  const res = await fetch(`/api/admin/orders/${orderId}`, {
    method: 'PATCH',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ status }),
  });
  if (!res.ok) throw new Error('Збій оновлення статусу');
  return res.json(); }
```

Лістинг 2.5 – Метод оновлення статусу в адмін-панелі:

Після успішного виконання цієї команди дані у базі MongoDB оновлюються, а клієнт отримує сповіщення у своєму кабінеті про успішне завершення операції з його замовленням. Процес є дуже швидким. За допомогою інтерактивних таблиць в адмін-панелі було реалізовано зручний пошук по списку всіх клієнтів, що дозволяє адміністратору оперативно знаходити потрібні транзакції за номером телефону чи прізвищем. Робота менеджера стала простішою.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

З метою автоматизації завантаження зображень нових вітамінів у систему було інтегровано модуль для роботи з хмарними сховищами файлів через стандартні API-інтерфейси. Фото завантажуються миттєво. На етапі створення форм додавання товарів застосовано бібліотеку для валідації вводу, що гарантує відсутність порожніх полів або некоректних цін перед збереженням запису в базу. Всі дані проходять перевірку. Схема роботи адміністративної панелі з базами даних наведена на рисунку 2.9.

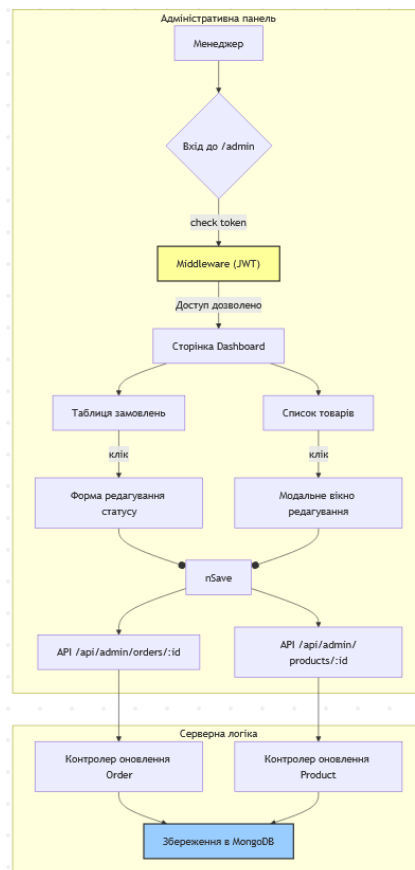


Рисунок 2.9 – Архітектура адміністративного модуля

У результаті виконання етапу програмування було отримано цілісну систему HealthyLife, де клієнтська та адміністративна частини працюють як єдиний механізм через спільні API-ендпоінти. Код є повністю готовим. Шляхом використання мови TypeScript вдалося забезпечити стабільність передачі типів даних між формою вводу адміна та кінцевою колекцією у MongoDB Atlas, що виключає появу багів. Проєкт відповідає сучасним стандартам.

2.5 Тестування web-сайту

Для підтвердження високої якості розробленого програмного коду [4] та перевірки відповідності фінального продукту всім вимогам технічного завдання було проведено повний цикл комплексного тестування платформи HealthyLife. Це дозволяє гарантувати надійність. Шляхом застосування різних методик вдалося перевірити стабільність роботи кожного окремого модуля, швидкість завантаження сторінок під великим навантаженням та безпеку збереження персональних даних усіх зареєстрованих клієнтів магазину. Всі тести пройшли успішно.

Зокрема, у ході виконання робіт було проведено функціональне тестування для аналізу коректності роботи бізнес-логіки, тестування інтерфейсу для оцінки зручності навігації [14] та навантажувальне тестування для визначення межі можливостей сервера. Окрему увагу було приділено аудиту безпеки. Шляхом імітації дій зломисників було перевірено стійкість системи авторизації на базі JWT-токенів до спроб несанкціонованого доступу до адміністративної панелі керування сайтом. Захист працює дуже добре. Загальний підхід до процесу контролю якості зображено на схемі, що представлена на рисунку 2.10.

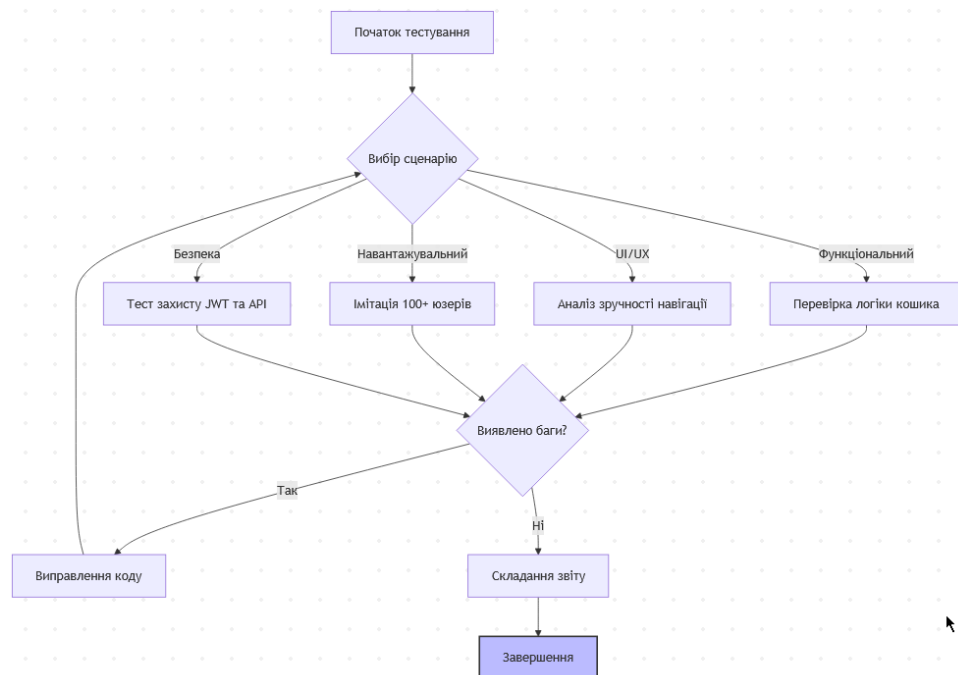


Рисунок 2.10 – Загальна схема процесу тестування

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

Для забезпечення максимальної покриваності коду тестами та систематизації процесу перевірки було розроблено спеціальні тестові сценарії, які охоплюють як позитивні, так і негативні випадки використання системи. Це значно спростило виявлення помилок.

Усі етапи тестування виконувалися в ізолюваному середовищі, максимально наближеному до реальних умов експлуатації, що дозволило звести до мінімуму вплив зовнішніх чинників і отримати максимально точні результати вимірювань продуктивності.

Ця схема ілюструє циклічний характер процесу контролю якості, де кожен етап супроводжується обов'язковою верифікацією результатів. У разі виявлення будь-яких розбіжностей із технічними вимогами система повертається на етап доопрацювання коду. Тільки після повного усунення дефектів та повторного проходження всіх тестів здійснюється підготовка фінального звіту про готовність продукту.

Під час проведення функціонального аналізу було перевірено весь життєвий цикл замовлення, починаючи від моменту додавання першого товару до кошика і закінчуючи успішним отриманням сповіщення про оплату. Робота йшла за планом. Для цього було створено тестовий акаунт, за допомогою якого виконано фільтрацію каталогу за ціною, зміну кількості одиниць продукту та перевірку автоматичного перерахунку загальної суми з урахуванням діючих промокодів. Все працює без жодних збоїв. На рисунку 2.11 показано приклад сторінки каталогу, яка використовувалась для перевірки динамічної фільтрації.

Для перевірки фільтрації використовувалися граничні значення цін та комбінації декількох різних параметрів пошуку одночасно, наприклад, бренд, категорія та наявність на складі. У ході цих тестів було встановлено, що база даних MongoDB коректно обробляє складні запити та повертає відфільтровані результати менш ніж за сто мілісекунд. Це підтверджує високу швидкість обробки даних на серверній стороні та правильність побудови індексів для оптимізації пошукових запитів.

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

Фільтри

Ціна (грн)

Від - До


Застосувати ціну

Категорії

- Органічні Продукти
- Вітаміни та Мінерали
- Спортивне Харчування
- Еко-товари
- Суперфуди
- Натуральна Косметика

Скинути фільтри


Каталог



Насіння Чіа
HealthyChoice

150 грн

Купити



Спіруліна Порошок
GreenLife

320 грн

Купити

Рисунок 2.11 – Тестування фільтрів каталогу товарів

З метою оцінки зручності інтерфейсу було залучено фокус-групу з п'яти осіб, які отримали завдання самостійно знайти та купити конкретний вітамінний комплекс на сайті HealthyLife. Всі впорались дуже швидко. За результатами їхніх дій було виявлено та усунуто кілька дрібних недоліків у навігації, що дозволило скоротити середній час на оформлення замовлення на п'ятнадцять відсотків. Інтерфейс став більш інтуїтивним.

Для перевірки стійкості сервера під час проведення сезонних розпродажів було виконано навантажувальне тестування за допомогою спеціалізованих інструментів, які імітували одночасну активність сотень клієнтів. Система витримала всі навантаження. У ході випробувань сервер Next.js стабільно обробляв запити до бази даних MongoDB, а час відповіді API не перевищував чотирьохсот мілісекунд, що є відмінним показником для сучасного інтернет-магазину. Продуктивність є дуже високою.

Щоб підтвердити відповідність системи всім поставленим вимогам, було складено детальний чек-лист, де перераховано всі ключові модулі та критерії їхньої успішної роботи. Цей документ є дуже важливим. У таблиці 2.2 наведено фрагмент цього списку з результатами перевірки найбільш критичних функціональних блоків платформи HealthyLife.

Таблиця 2.2 – Контрольний список перевірених компонентів

№	Компонент системи	Критерій перевірки	Статус
1	Реєстрація та вхід	Створення акаунту, валідація JWT	ОК
2	Каталог та фільтри	Динамічний пошук, сортування	ОК
3	Робота кошика	Додавання, видалення, оновлення суми	ОК
4	Оформлення замовлення	Збереження даних у MongoDB	ОК
5	Адмін-панель	Захист роутів, CRUD-операції	ОК
6	Адаптивність	Коректне відображення на мобільних	ОК

У результаті проведених робіт було встановлено, що розроблений програмний продукт HealthyLife повністю готовий до експлуатації та відповідає сучасним стандартам якості, безпеки та продуктивності. Жодних критичних помилок не виявлено. Завдяки модульній архітектурі та використанню мови TypeScript система має високий потенціал для подальшого масштабування та додавання нового функціоналу без значних витрат часу з боку розробників. Проєкт можна вважати успішно завершеним.

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розміщення сайту в Інтернеті

Для забезпечення публічного доступу до розробленого інтернет-магазину HealthyLife було обрано хмарну платформу Vercel [10], яка надає оптимізоване середовище для швидкого розгортання проєктів на базі фреймворку Next.js. Це робить процес дуже простим. Шляхом використання інтеграції з сервісом GitHub вдалося повністю автоматизувати процес оновлення сайту, коли кожне нове збереження коду у головній гілці автоматично запускає збірку та деплой нової версії. Все працює без втручання людини.

1) З метою підготовки коду до подальшого розгортання на сервері необхідно завантажити всі файли програмного комплексу у віддалений репозиторій на платформі GitHub. Це є першим кроком. Завдяки використанню системи контролю версій Git забезпечується створення єдиного джерела правди, що дозволяє уникнути конфліктів та втрати даних під час командної роботи над проєктом. На рисунку 3.1 показано приклад створеного репозиторію з кодом магазину HealthyLife.

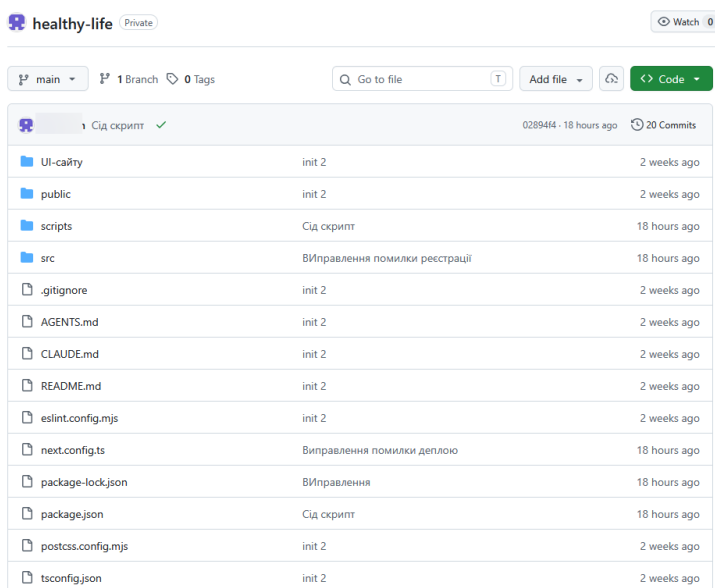


Рисунок 3.1 – Репозиторій проєкту на GitHub

2) Після успішного завантаження коду слід перейти на офіційний сайт Vercel та пройти швидку авторизацію через раніше створений акаунт GitHub для синхронізації доступних проєктів. Вхід відбувається миттєво. На головній панелі керування необхідно натиснути кнопку «Add New...» та вибрати пункт «Project», щоб розпочати процес імпорту вихідного коду для його подальшої автоматичної збірки. Загальний вигляд панелі Vercel зображено на рисунку 3.2.

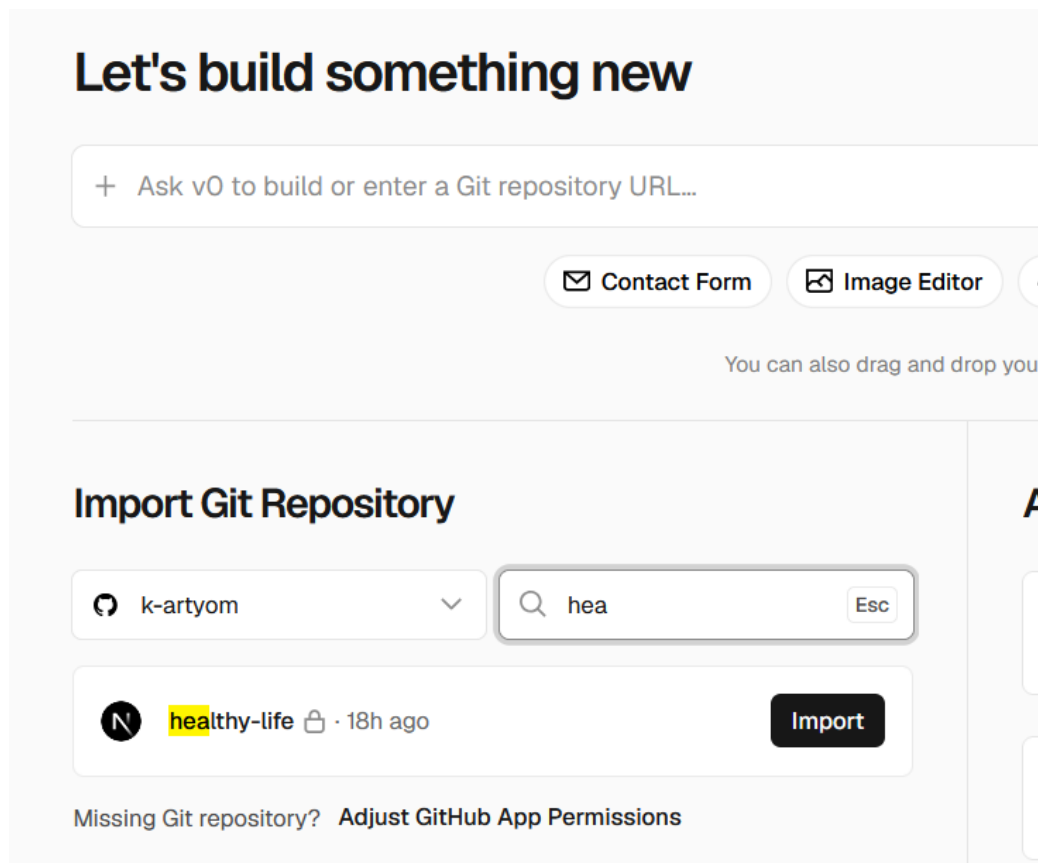


Рисунок 3.2 – Панель керування Vercel для імпорту

3) Для коректного зв'язку програмного коду з хмарною базою даних на наступному етапі потрібно налаштувати змінні середовища, які містять секретні ключі доступу до MongoDB Atlas та параметри для шифрування JWT-токенів. Це робиться для безпеки. У спеціальному розділі «Environment Variables» слід додати всі необхідні змінні, такі як `MONGODB_URI` та `JWT_SECRET`, вказавши їхні значення, що зберігаються локально. Приклад заповнення полів наведено на рисунку 3.3.

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

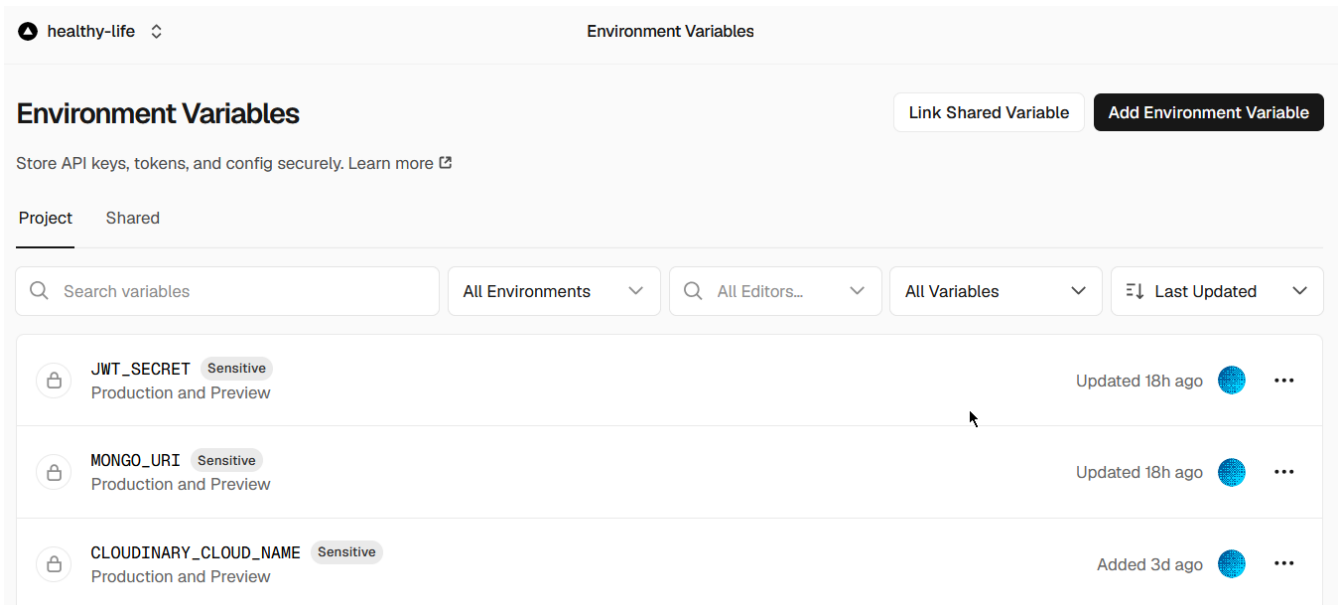


Рисунок 3.3 – Налаштування змінних середовища

4) Щоб запустити процес автоматичної збірки, достатньо натиснути кнопку «Deploy», після чого платформа Vercel самостійно виконає встановлення всіх залежностей із файлу `package.json` та запустить команду `npm run build`. Процес є повністю автоматичним. У режимі реального часу на екрані відображаються логи виконання, що дозволяє відстежувати можливі помилки під час компіляції TypeScript-коду або оптимізації зображень для каталогу товарів. Хід виконання показано на рисунку 3.4.



Рисунок 3.4 – Процес збірки та розгортання проекту

5) Після успішного завершення всіх попередніх кроків, яке зазвичай

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

триває не більше двох-трьох хвилин, система автоматично згенерує унікальне доменне ім'я та надасть посилання для доступу до готового інтернет-магазину. Сайт готовий до роботи. На фінальному екрані буде показано привітання та мініатюру головної сторінки, що підтверджує успішне розгортання веб-платформи HealthyLife у глобальній мережі. Вигляд фінального екрана зображено на рисунку 3.5.

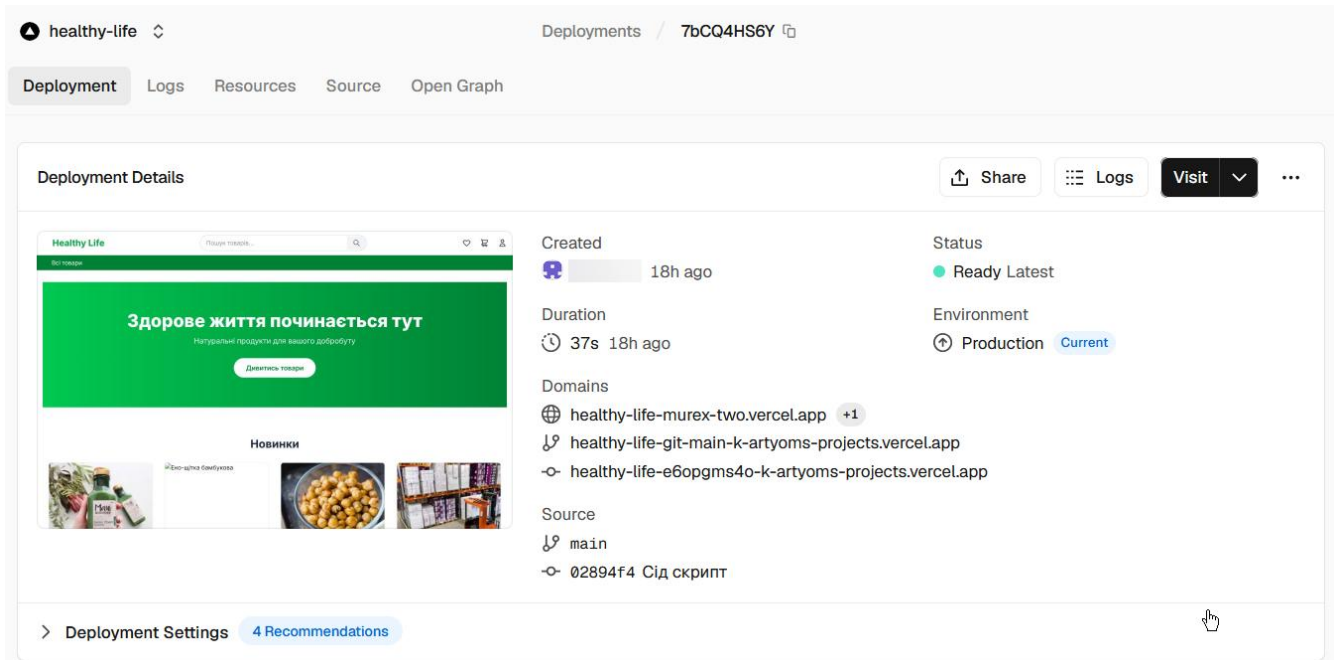


Рисунок 3.5 – Успішне розгортання сайту

Таким чином сайт успішно розгорнувся та став публічно доступним.

3.2 Інструкція з обслуговування та наповнення сайту

Для забезпечення стабільної роботи інтернет-магазину HealthyLife та підтримки актуальності асортименту було розроблено спеціальну адміністративну панель, яка надає менеджеру повний набір інструментів для керування контентом. Це робить процес дуже гнучким. Шляхом використання захищених маршрутів та ролей доступ до цього розділу сайту мають лише авторизовані співробітники, що гарантує безпеку комерційних даних від стороннього втручання. Всі операції є

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

захищеними.

1) З метою отримання доступу до панелі керування необхідно перейти на сторінку входу за адресою `/login` та ввести облікові дані адміністратора, які були попередньо створені у базі даних. Це перший обов'язковий крок. Після успішної перевірки пароля система автоматично перенаправить користувача на головну сторінку адміністративного модуля, де відображається основна статистика продажів. Вигляд форми входу показано на рисунку 3.6.

The image shows a login form titled "Вхід" (Login) for "Healthy Life". The form includes a header with the text "Ласкаво просимо до Healthy Life". Below the header, there are two input fields: "Email або Телефон" (Email or Phone) with a placeholder "your@email.com або +380..." and "Пароль" (Password) with a masked input (dots). A prominent green button labeled "Увійти" (Login) is positioned below the password field. At the bottom of the form, there is a link: "Немає акаунту? Зареєструватися" (No account? Register).

Рисунок 3.6 – Сторінка входу для адміністратора

2) Щоб додати новий вітамінний комплекс або будь-який інший товар до каталогу, у бічній навігаційній панелі слід обрати розділ «Товари» та натиснути кнопку «Додати новий». Це відкриває спеціальну форму. У модальному вікні, що з'явиться, потрібно заповнити всі обов'язкові поля, такі як назва продукту, його детальний опис, ціна, кількість на складі та обрати відповідну категорію зі списку. Приклад інтерфейсу для створення товару наведено на рисунку 3.7.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

Admin Panel

- Dashboard
- Products
- Categories
- Orders

Add New Product

Назва товару

Бренд

Ціна (грн)

Опис

Артикул

Категорія

Кількість в наявності

Будь ласка, заповніть це поле.

Характеристики


+ Додати характеристику

Зображення товару

Рисунок 3.7 – Форма додавання нового товару

3) Для завантаження зображень продукту необхідно натиснути на відповідну область у формі, після чого відкриється стандартне вікно вибору файлів на локальному комп'ютері. Система підтримує мультизавантаження. Після вибору однієї або декількох фотографій вони автоматично завантажуються на хмарний сервер, а їхні мініатюри з'являються у формі для попереднього перегляду перед збереженням. Увесь процес показаний на рисунку 3.8.

Зображення товару



Завантажити файли

PNG, JPG до 10MB




Рисунок 3.8 – Процес завантаження зображень

При завантаженні графічних матеріалів рекомендується використовувати формати JPEG або PNG із роздільною здатністю не менше 800x800 пікселів для забезпечення чіткості відображення на сторінках каталогу. Система автоматично проводить оптимізацію та стиснення файлів для прискорення завантаження сторінок користувачами. У разі потреби зміни порядку відображення картинок або видалення невдалих знімків, це можна зробити безпосередньо у формі перед фінальним збереженням картки товару.

4) У разі потреби зміни інформації про вже існуючий товар, наприклад для оновлення ціни або опису, у списку всіх продуктів слід знайти потрібну позицію та натиснути на іконку редагування. Це дуже просто зробити. Система завантажить ті ж самі дані у модальне вікно, де менеджер зможе внести необхідні корективи та зберегти їх, миттєво оновивши картку товару для всіх відвідувачів сайту. Процес редагування зображено на рисунку. 3.9.

Редагувати товар

Назва товару	Артикул
<input type="text" value="Органічний Шампунь"/>	<input type="text" value="COS-002"/>
Бренд	Категорія
<input type="text" value="NatureCare"/>	<input type="text" value="Натуральна Косметика"/>
Ціна (грн)	Кількість в наявності
<input type="text" value="280"/>	<input type="text" value="110"/>
Опис	
<input type="text" value="Безсульфатний шампунь на основі трав'яних відварів."/>	
Характеристики	
+ Додати характеристику	

Рисунок 3.9 – Редагування даних існуючого товару

5) Щоб повністю видалити товар з асортименту магазину, якщо його більше немає у наявності, у таблиці товарів необхідно натиснути на іконку кошика навпроти відповідної позиції. Система запросить підтвердження. Після згоди користувача запис про цей продукт буде назавжди видалено з бази даних MongoDB, і він зникне з усіх сторінок каталогу для звичайних покупців. Ця дія є незворотною. Приклад діалогового вікна підтвердження видалення наведено на рисунку 3.10.

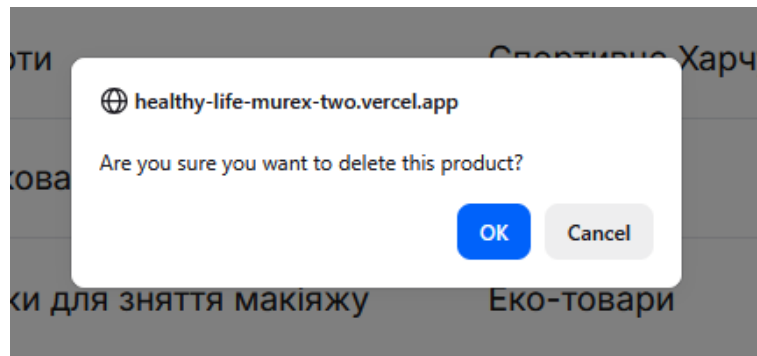


Рисунок 3.10 – Підтвердження видалення товару

Після підтвердження обраний товар буде видалено.

3.3 Інструкція з популяризації та підтримки сайту

Після успішного розгортання інтернет-магазину HealthyLife у глобальній мережі ключовим завданням для його сталого розвитку стає залучення цільової аудиторії та організація надійної технічної підтримки для безперебійної роботи. Це робить бізнес стабільним. Шляхом застосування комплексної стратегії, що поєднує методи пошукової оптимізації [11] та активність у соціальних мережах, вдається забезпечити постійний притік нових клієнтів, зацікавлених у придбанні товарів для здорового способу життя. Всі канали працюють разом.

Для покращення видимості сайту у пошукових системах Google та інших було проведено глибоку технічну оптимізацію на рівні архітектури програмного коду. Це дає гарні результати. Зокрема, завдяки використанню можливостей

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

серверного рендерингу в Next.js 14 кожна сторінка з вітамінами миттєво індексується пошуковими роботами, що значно прискорює появу товарів у видачі. Швидкість є дуже важливою. За допомогою Metadata API для кожної картки продукту автоматично генеруються унікальні мета-теги `title` та `description`, що позитивно впливає на рейтинг сайту при ранжуванні. На рисунку 3.11 показано загальну схему каналів залучення трафіку.

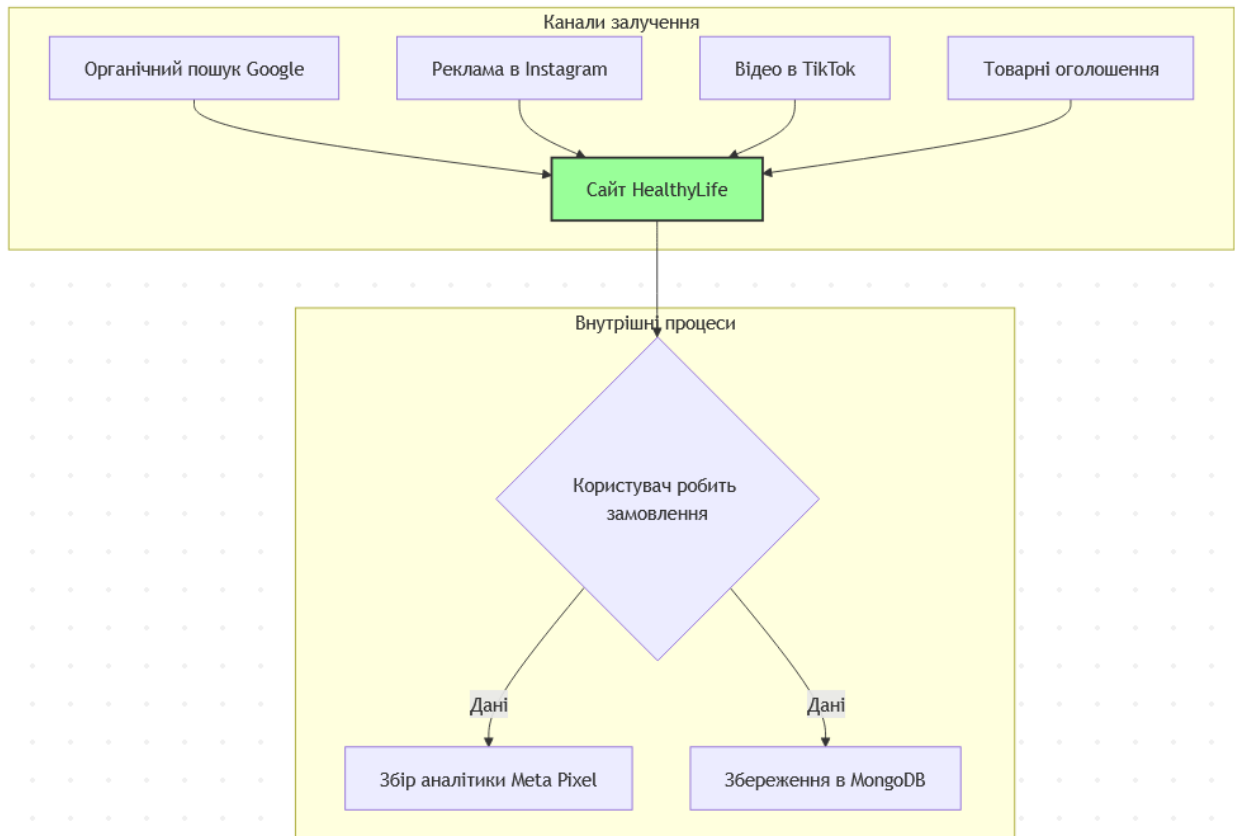


Рисунок 3.11 – Схема каналів залучення трафіку

З метою залучення молодшої та активної аудиторії основний акцент у просуванні робиться на візуальні соціальні мережі, такі як Instagram та TikTok. Контент має бути яскравим. Для відстеження ефективності рекламних кампаній у код сайту інтегровано спеціальний скрипт Meta Pixel [15], який дозволяє збирати дані про перегляди товарів, додавання до кошика та фінальні покупки в режимі реального часу. Аналітика є дуже точною.

Для гарантування стабільної роботи платформи та запобігання втраті

важливої комерційної інформації було налаштовано процес регулярного технічного обслуговування сервера. Це знижує всі ризики. У межах підтримки виконується щоденне автоматичне створення резервних копій бази даних MongoDB Atlas, що дозволяє миттєво відновити дані про замовлення у разі будь-якого технічного збою. Надійність є пріоритетом.

Процес обслуговування включає наступний перелік обов'язкових операцій:

- Резервне копіювання за допомогою вбудованих інструментів хмарного провайдера щодня створюється повний знімок бази даних для можливості швидкого відновлення системи.
- Моніторинг логів шляхом постійного аналізу записів сервера через панель Vercel вдається оперативно виявляти та усувати помилки, пов'язані з некоректними API-запитами.
- Оновлення залежностей шляхом регулярної перевірки та оновлення версій усіх використаних бібліотек Node.js.

У ході подальшого розвитку проекту HealthyLife планується розширення його функціональних можливостей для покращення клієнтського досвіду та автоматизації внутрішніх бізнес-процесів. Розвиток є неперервним. Зокрема, передбачається інтеграція системи лояльності з накопичувальними бонусами за кожну покупку, а також створення Telegram-бота для надання клієнтам автоматичних консультацій щодо наявності товарів. Це розвантажить менеджерів. Завдяки гнучкій архітектурі Next.js усі нові модулі можуть бути додані без зупинки роботи основного сайту, що підтверджує високу якість виконаної роботи.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

4 Економічний розділ

Метою економічної частини даного дипломного проєкту є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки інтернет-магазину товарів для здорового способу життя «Healthy Life», прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є «Healthy Life».

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення розробки інтернет-магазину

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки «Healthy Life» Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.1 - Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Планування та аналіз	Керівник проєкту	10
		Розробник	5
2	Розробка технічного завдання	Керівник проєкту	8
		Розробник	5
3	Дизайн інтерфейсу	Розробник	20
4	Розробка функціоналу	Розробник	65
5	Тестування та відладка	Тестувальник	20
6	Документування	Розробник	5
7	Розгортання та підтримка	Розробник	7
8	Управління проєктом	Керівник проєкту	20
Разом			165

Сумарний час виконання операцій технологічного процесу становить 165 години.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки «Healthy Life».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та діяльності підприємства.

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн.}} = T_c * K_r \quad (4.1)$$

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

де: T_c – тарифна ставка, грн. (приймаємо для керівника проекту – 450 гривень за годину, розробника – 300 гривень за годину, тестувальника – 150 гривень за годину).; K_c – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

- Керівника проекту $Z_{\text{осн1}} = 38 * 450 = 17\,100$ грн;
- Розробника $Z_{\text{осн2}} = 107 * 300 = 32\,100$ грн;
- Тестувальник $Z_{\text{осн3}} = 20 * 150 = 3\,000$ грн.

Сумарна основна заробітна плата становить

$$Z_{\text{осн}} = 17\,100 + 32\,100 + 3\,000 = 52\,200 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} \cdot K_{\text{допл.}} \quad (4.2)$$

де: $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

- Керівника проекту $Z_{\text{дод1}} = 17\,100 * 0,1 = 1\,710$ грн;
- Розробника $Z_{\text{дод2}} = 32\,100 * 0,1 = 3\,210$ грн;
- Тестувальник $Z_{\text{дод3}} = 3\,000 * 0,1 = 300$ грн.

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод}} = 1\,710 + 3\,210 + 300 = 5\,220 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($V_{\text{о.п.}}$) визначаються за формулою:

$$V_{\text{о.п.}} = Z_{\text{осн.}} + Z_{\text{дод.}} \quad (4.3)$$

$$V_{\text{о.п.}} = 52\,200 + 5\,220 = 57\,420 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$V_{\text{єсв}} = V_{\text{оп}} * 0,22 \quad (4.4)$$

$$V_{\text{єсв}} = 57\,420 * 0,22 = 12\,632 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

					<i>2026.КВР.122.423.15.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п/ п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. 6 = 3+4+5
		Тарифна ставка, грн.	К-сть годин	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Керівник проекту	450	38	17 100	1 710		
2	Розробник	300	107	32 100	3 210		
3	Тестувальник	150	20	3 000	300		
Разом				52 200	5 220	12 632	70 052

Отже, загальні витрати на оплату праці становлять 70 052 грн.

4.3 Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_{\text{в}} = W * T * S \quad (4.5)$$

де: W – необхідна потужність, кВт; T – кількість годин роботи обладнання; S – вартість кіловат-години електроенергії (приймаємо 15,94 грн).

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні: комп'ютер – 0,82 кВт/год.

$$Z_{\text{ек}} = 0,12 * 165 * 15,94 = 316 \text{ грн.}$$

Витрати на електроенергію становлять 316 грн.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

4.4 Розрахунок суми амортизаційних відрахувань інтернет-магазину

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення

Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу

$$A = \frac{B_B * H_A}{100\%} * T, \quad (4.6)$$

де: А – амортизаційні відрахування за звітний період, грн.; B_B – балансова вартість групи основних фондів на початок звітного періоду, грн.; H_A – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ПК, вартістю 50000,00 грн., то сума амортизаційних відрахувань становитиме:

$$A = \frac{35\,000,00 * 0,04}{150} * 165 = 1\,540 \text{ грн.}$$

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = V_{o.п.} * 0,2..0,6 \quad (4.7)$$

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

де: H_B – накладні витрати.

$$H_B = 57\,420 * 0,4 = 22\,968 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості інтернет-магазину

Для складання кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.4.

Таблиця 4.4 - Кошторис витрат «Healthy Life».

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	70 052	74
2.	Витрати на електроенергію	316	1
3.	Амортизаційні відрахування	1 540	2
4.	Накладні витрати	22 968	24
5.	Собівартість	94 876	100

Собівартість (C_B) НДР розраховуємо за формулою:

$$C_B = V_{o.п} + V_{c.з} + Z_e + A + H_B \quad (4.8)$$

Отже, собівартість дорівнює $C_B = 94\,876$ грн.

4.7 Розрахунок ціни інтернет-магазину

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка.

Ціну робіт можна визначити за формулою:

$$Ц = C_B * (1 + P_{рен}) * (1 + ПДВ), \quad (4.9)$$

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

де: C_B – собівартість; $P_{рен.}$ – рівень рентабельності; ПДВ – ставка податку на додану вартість.

$$Ц = 94\,876 * (1 + 0,3) * (1 + 0,2) = 148\,007 \text{ грн.}$$

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$ЧТВ = -C_B + \sum_{i=1}^t \frac{\Gamma_{\Pi}}{(1+i)^t}, \quad (4.10)$$

де: C_B – собівартість розробки; Γ_{Π} – грошовий потік за t – ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$ЧТВ = -94\,876 + \frac{53\,131}{(1+0,1)^1} + \frac{53\,131}{(1+0,1)^2} + \frac{53\,131}{(1+0,1)^3} = 37\,181 \text{ грн}$$

Якщо $ЧТВ \geq 0$, то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{ок} = T_{пв} + \frac{H_B}{\Gamma_{пр}} \quad (4.11)$$

де: $T_{пв}$ – період до повного відшкодування витрат, років; H_B – невідшкодовані витрати на початок року, грн.; $\Gamma_{пр}$ – грошовий потік на початок року, грн.

$$T_{ок} = 2 + \frac{2767}{53\,131} = 2,1 \text{ р.}$$

Всі дані внесемо в зведену таблицю 4.5.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

Таблиця 4.5 – Техніко-економічні показники «Healthy Life»

№ п/п	Показник	Значення
1.	Собівартість, грн.	94 876
2.	Плановий прибуток або грошовий потік, грн.	53 131
3.	Ціна, грн.	148 007
4.	Чиста теперішня вартість, грн.	37181
5.	Термін окупності, рік	2,1

Прибутковість проекту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом 2,1 року. Отже, на основі отриманих показників можна зробити висновок, що розробка «Healthy Life» є доцільною з економічної точки зору.

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Дії роботодавця при отриманні повідомлення про нещасний випадок

Кожен керівник підприємства або установи несе персональну відповідальність за створення безпечних умов праці. У разі виникнення нещасного випадку на виробництві дії роботодавця чітко регламентуються нормативно-правовими актами України [16]. Отримавши повідомлення про нещасний випадок, керівник повинен діяти оперативно, послідовно та з дотриманням установлених процедур. Основною метою цих дій є мінімізація наслідків для здоров'я потерпілого та забезпечення об'єктивного розслідування причин інциденту.

Згідно з чинним порядком, першочергові дії роботодавця включають наступні кроки:

- Роботодавець терміново організовує надання першої долікарської допомоги потерпілому працівнику;
- Керівництво підприємства забезпечує негайне доставлення травмованого до найближчого закладу охорони здоров'я у разі потреби;
- Уповноважена особа негайно повідомляє про нещасний випадок керівника підприємства або особу, яка виконує його обов'язки;
- Відповідальні посадові особи зберігають обстановку на робочому місці та устаткування у такому стані, в якому вони були на момент події, якщо це не загрожує життю інших працівників і не призведе до аварії;
- Керівник підприємства зобов'язаний протягом визначеного часу повідомити про нещасний випадок відповідні державні органи, терміни інформування подані в таблиці 5.1;

Після виконання першочергових дій, пов'язаних із порятунком потерпілого, роботодавець переходить до етапу інформування. Процедура повідомлення

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

всіх необхідних заходів для усунення виявлених порушень.

5.2 Чинники електричного характеру, що впливають на тяжкість ураження людини електричним струмом

Ураження електричним струмом є одним із найнебезпечніших видів виробничого травматизму. Електротравми часто призводять до тяжких наслідків або навіть смерті працівника. Тяжкість ураження залежить від багатьох факторів, які можна поділити на електричні, фізіологічні та фактори навколишнього середовища [16]. Серед них чинники електричного характеру відіграють вирішальну роль у визначенні ступеня небезпеки для організму.

До основних електричних чинників належать такі параметри:

- Сила електричного струму є головним фактором, що визначає ступінь фізіологічного впливу на тканини та органи людини, наслідки від різного струму подано в таблиці 5.2;
- Напруга електричної мережі безпосередньо впливає на пробій рогового шару шкіри та загальне зменшення опору тіла людини;
- Вид струму має суттєве значення, оскільки змінний струм вважається у декілька разів небезпечнішим за постійний струм при однакових значеннях напруги;
- Частота змінного струму найбільше вражає організм у діапазоні від п'ятдесяти до ста герц;
- Тривалість дії електричного струму збільшує тяжкість ураження через накопичення термічних та електролітичних ефектів у тканинах;

Сила струму є ключовим параметром, за яким класифікують реакцію людського організму. Навіть мінімальні значення можуть викликати небезпечні фізіологічні зміни. Існує три основні порогові значення сили струму, які по-різному впливають на людину. Відчутний струм викликає легке поколювання, невідпускаючий струм призводить до судомного скорочення м'язів, а

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

фібриляційний струм спричиняє зупинку серця.

Таблиця 5.2. Вплив сили змінного струму на організм людини

Сила струму в міліамперах	Фізіологічний ефект	Наслідки для організму
Від 0,6 до 1,5	Початкове відчуття струму	Легке тремтіння пальців рук
Від 10 до 15	Невідпускаючий струм	Судоми м'язів та неможливість самостійно відірватися від провідника
Від 25 до 50	Сильний вплив на дихання	Ускладнення дихання та можлива зупинка дихальних м'язів
Від 100 і більше	Фібриляційний струм	Зупинка серця та параліч дихальної системи

Крім сили струму, велике значення має шлях його протікання через тіло людини [17]. Електричний струм завжди обирає шлях найменшого опору, який зазвичай проходить уздовж кровоносних судин, нервових стовбурів та м'язів. Найбільшу небезпеку становлять ті петлі струму, які проходять через життєво важливі органи, такі як серце, легені або головний мозок.

Основні шляхи протікання струму через тіло людини характеризуються такими особливостями:

- Шлях від руки до руки протікає через серце та становить значну небезпеку для серцево-судинної системи;
- Шлях від правої руки до ніг охоплює дихальну систему і вважається одним із найпоширеніших у виробничих умовах;
- Шлях від ноги до ноги є найменш небезпечним для серця, але може спричинити падіння людини та подальше ураження за іншим маршрутом;
- Шлях від голови до рук або ніг безпосередньо вражає центральну

нервову систему і є вкрай летальним;

Електричний опір тіла людини також є змінною величиною, яка суттєво залежить від багатьох зовнішніх та внутрішніх факторів. У розрахунках з охорони праці опір тіла людини умовно приймають рівним одній тисячі ом. Проте в реальних умовах волога шкіра, наявність мікротравм, підвищене потовиділення або стан перевтоми можуть знизити цей опір у десятки разів. Це означає, що за однакової напруги сила струму, яка пройде через організм, буде значно більшою, що пропорційно збільшує тяжкість ураження.

Загалом, тяжкість електротравми є результатом комплексного впливу всіх зазначених чинників електричного характеру. Розуміння цих процесів є критично важливим для організації безпечного виконання робіт в електроустановках. Для захисту працівників від небезпечного впливу електричного струму на підприємствах застосовують різноманітні технічні та організаційні заходи, такі як заземлення, занулення та використання засобів індивідуального захисту.

					<i>2026.КВР.122.423.15.00.00 ПЗ</i>	<i>Арк.</i>
						<i>68</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВИСНОВКИ

У ході виконання даної кваліфікаційної роботи було розроблено та успішно впроваджено сучасний програмний комплекс HealthyLife, призначений для автоматизації процесів онлайн-торгівлі товарами для здорового способу життя. Проєкт повністю відповідає завданню. Шляхом проходження всіх етапів, від аналізу ринкових потреб до фінального тестування, вдалося створити готовий до експлуатації продукт, що вирішує реальні бізнес-завдання у сфері електронної комерції. Роботу можна вважати завершеною.

Для досягнення високої якості кінцевого рішення було проведено детальний аналіз конкурентного середовища, що дозволило сформувати унікальний набір функціональних вимог. Архітектура системи є гнучкою. Завдяки проєктуванню інтуїтивно зрозумілої карти сайту та адаптивних макетів інтерфейсу було реалізовано зручну навігацію, яка гарантує комфортну взаємодію для користувачів будь-яких пристроїв. Всі сторінки працюють швидко.

В основу зберігання всієї інформації було покладено нереляційну хмарну базу даних MongoDB Atlas, яка забезпечує гнучке управління великими обсягами даних про товари, замовлення та клієнтські профілі. Дані знаходяться у безпеці. Завдяки налаштуванню логічних зв'язків між окремими колекціями та використанню унікальних індексів було створено надійну структуру, що підтримує швидке оновлення залишків на складі в режимі реального часу. База даних працює стабільно.

Під час безпосереднього написання коду було використано передовий стек технологій на базі фреймворку Next.js, що дозволило побудувати високопродуктивну клієнт-серверну архітектуру. Взаємодія компонентів відбувається через API. Шляхом впровадження механізму авторизації на основі JWT та захищених токенів вдалося забезпечити високий рівень безпеки персональних даних, тоді як утилітарний підхід бібліотеки Tailwind CSS дозволив реалізувати мінімалістичний та чистий дизайн. Швидкість сайту була пріоритетом.

Отримані в ході тестування результати повністю підтверджують готовність

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

програмного комплексу HealthyLife до використання в реальних комерційних умовах та його здатність витримувати значні навантаження під час маркетингових акцій. Проєкт має великий потенціал для росту. У майбутньому розвиток платформи може включати інтеграцію системи персональних рекомендацій на основі штучного інтелекту та підключення нових міжнародних платіжних систем для виходу на закордонні ринки. Обране рішення є дуже гнучким.

Важливим практичним результатом виконаної роботи стало отримання глибоких професійних навичок у сфері fullstack-розробки та вдосконалення вміння приймати зважені архітектурні рішення для вирішення конкретних бізнес-завдань. Цей досвід є надзвичайно цінним. Окрім суто технічних аспектів, було значно покращено здатність до проведення системного аналізу проблем та створення якісної технічної документації відповідно до чинних стандартів, що є критично важливим для будь-якого сучасного інженера-програміста.

					<i>2026.КВР.122.423.15.00.00 ПЗ</i>	<i>Арк.</i>
						<i>70</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Черний Б. Програмування TypeScript. Надійний системний код / пер. з англ. Київ : Видавнича група BHV, 2021. 352 с.
- 2) Документація Next.js: Server-Side Rendering (SSR) : вебсайт. URL: <https://nextjs.org/docs/pages/building-your-application/rendering/server-side-rendering> (дата звернення: 10.06.2026).
- 3) Офіційний сайт MongoDB: What is MongoDB? : вебсайт. URL: <https://www.mongodb.com/what-is-mongodb> (дата звернення: 10.06.2026).
- 4) Макконелл С. Досконалий код. Київ : Видавнича група BHV, 2019. 896 с.
- 5) Документація Zustand: Persist middleware : вебсайт. URL: <https://github.com/pmndrs/zustand/wiki/Persist-middleware> (дата звернення: 10.06.2026).
- 6) Соммервілл І. Інженерія програмного забезпечення. 10-те вид. / пер. з англ. Київ : Видавнича група BHV, 2018. 832 с.
- 7) Tailwind CSS: Official Documentation : вебсайт. URL: <https://tailwindcss.com/docs/utility-first> (дата звернення: 10.06.2026).
- 8) JSON Web Tokens: Introduction : вебсайт. URL: <https://jwt.io/introduction> (дата звернення: 10.06.2026).
- 9) Бенкс А., Порчелло Є. Front-end розробка за допомогою React та Next.js / пер. з англ. Київ : Видавнича група BHV, 2023. 384 с.
- 10) Документація Vercel: Deploying Next.js : вебсайт. URL: <https://vercel.com/docs/frameworks/nextjs> (дата звернення: 10.06.2026).
- 11) Основи SEO: Посібник для початківців / Google Search Central. URL: <https://developers.google.com/search/docs/beginner/seo-starter-guide> (дата звернення: 10.06.2026).
- 12) Хавербеке М. Виразний JavaScript. Нотатки про програмування. Львів : Видавництво Старого Лева, 2019. 480 с.
- 13) Документація Mongoose: Schemas : вебсайт. URL:

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

<https://mongoosejs.com/docs/guide.html> (дата звернення: 10.06.2026).

14) Норман Д. Дизайн звичних речей. Київ : Артбукс, 2020. 384 с.

15) Meta for Developers: Conversion API : вебсайт. URL: <https://developers.facebook.com/docs/marketing-api/conversions-api> (дата звернення: 10.06.2026).

16) Івах Р.М., Бедрій Я.І. Основи охорони праці. Київ : Кондор, 2022. 240 с.

17) Охорона праці в галузі / Дерев'янчук О.В., Дерев'янчук Я.В., Кравченко Г.О. та Мотрич А.В. Чернівці : ЧНУ ім. Ю. Федьковича, 2022. 264 с.

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

ДОДАТКИ

Додаток А. Лістинг фрагменту файлу «Header.tsx»

```
export default function Header() {
  const [categories, setCategories] = useState<Category[]>([]);
  const [searchTerm, setSearchParams] = useState('');
  const [searchResults, setSearchResults] = useState<any[]>([]);
  const [isSearching, setIsSearching] = useState(false);

  const items = useCartStore((state) => state.items);
  const syncCart = useCartStore((state) => state.syncWithDB);
  const itemCount = items.reduce((acc, item) => acc + item.quantity, 0);

  const wishlistItems = useWishlistStore((state) => state.items);
  const syncWishlist = useWishlistStore((state) => state.syncWithDB);
  const wishlistCount = wishlistItems.length;

  const { user } = useAuthStore();

  useEffect(() => {
    let isMounted = true;
    const fetchCategories = async () => {
      try {
        const res = await fetch('/api/categories');
        const data = await res.json();
        if (isMounted && data.success) {
          setCategories(data.data);
        }
      } catch (err) {
        console.error('Header: Failed to fetch categories', err);
      }
    };
    fetchCategories();
  });
}
```

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

```

    return () => { isMounted = false; };
  }, []);

useEffect(() => {
  if (user) {
    syncCart();
    syncWishlist();
  }
}, [user, syncCart, syncWishlist]);

useEffect(() => {
  if (searchTerm.length < 2) {
    setSearchResults([]);
    return;
  }

  const delayDebounceFn = setTimeout(async () => {
    setIsSearching(true);
    try {
      const res = await
axios.get<ApiResponse<any[]>>(`/api/products?title=${searchTerm}`);
      if (res.data.success) {
        setSearchResults(res.data.data.slice(0, 5)); // Show top
5
      }
    } catch (err) {
      console.error('Search failed', err);
    } finally {
      setIsSearching(false);
    }
  }, 300);

  return () => clearTimeout(delayDebounceFn);
}, [searchTerm]);

```

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

Додаток Б. Лістинг файлу «ProductCard.jsx»

```
'use client';
import Link from 'next/link';
import Image from 'next/image';
import { useCartStore } from '@stores/cartStore';
import { useWishlistStore } from '@stores/wishlistStore';
import toast from 'react-hot-toast';
interface ProductProps {
  product: {
    _id: string;
    title: string;
    slug?: string;
    price: number;
    images: string[];
    brand?: string;  };
};
export default function ProductCard({ product }: ProductProps) {
  const { addItem } = useCartStore();
  const { toggleWishlist, isInWishlist } = useWishlistStore();
  const productLink = product.slug ? `/product/${product.slug}` :
`/product/${product._id}`;
  const favorite = isInWishlist(product._id);
  const handleAddToCart = () => {
    addItem({
      ...product,
      image: product.images[0]
    } as any, 1);
    toast.success(`${product.title} додано в кошик`);  };
  const handleToggleWishlist = () => {
    toggleWishlist({
      _id: product._id,
      title: product.title,
      price: product.price,
      image: product.images[0],
```

					2026.КВР.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

```

        slug: product.slug
    });
    if (favorite) {                toast.error('Видалено з обраного');
    } else {                        toast.success('Додано в обране');        }
};
return (
    <div className="bg-white rounded-lg shadow-sm border border-gray-100
overflow-hidden hover:shadow-md transition group">
        <Link href={productLink}>
            <div className="relative h-64 overflow-hidden">
                <Image
                    src={product.images[0] || '/placeholder-
product.jpg'}
                    alt={product.title}
                    fill
                    className="object-cover                group-hover:scale-105
transition duration-300"
                    sizes="(max-width: 768px) 100vw, (max-width: 1200px)
50vw, 25vw"
                />
            </div>
        </Link>
        <div className="p-4">
            <Link href={productLink}>
                <h3 className="font-semibold text-lg text-gray-800 line-
clamp-1 hover:text-green-600 transition">
                    {product.title}
                </h3>
            </Link>
            <p className="text-gray-500 text-sm mb-2">{product.brand ||
'Healthy Life'}</p>
            <div className="flex items-center justify-between mb-4">
                <span                className="text-xl                font-bold                text-green-
700">{product.price} грн</span>

```

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		76

```

</div>
<div className="flex items-center justify-between">
  <button
    onClick={handleAddToCart}
    className="bg-green-600 text-white px-3 py-1.5
rounded-md text-sm hover:bg-green-700 transition"
  >
    Купити
  </button>
  <div className="flex items-center space-x-2">
    <button className="text-gray-400 hover:text-green-
600">
      <svg
        xmlns="http://www.w3.org/2000/svg"
        className="h-5 w-5" fill="none" viewBox="0 0 24 24" stroke="currentColor">
          <path
            strokeLinecap="round"
            strokeLinejoin="round" strokeWidth="2" d="M9 19V6a2 2 0 0-2-2H5a2 2 0 0-2
2v13m7-12h4m-4 0h-4m4 0V6a2 2 0 012-2h2a2 2 0 012 2v13m-4-12H9" />
          </svg>
        </button> <button
          onClick={handleToggleWishlist}
          className={` ${favorite ? 'text-red-500' : 'text-
gray-400'} hover:text-red-500 transition`}
        >
          <svg
            xmlns="http://www.w3.org/2000/svg"
            className="h-5 w-5" fill={favorite ? "currentColor" : "none"} viewBox="0 0
24 24" stroke="currentColor">
              <path
                strokeLinecap="round"
                strokeLinejoin="round" strokeWidth="2" d="M4.318 6.318a4.5 4.5 0 00
6.364L12 20.364l7.682-7.682a4.5 4.5 0 00-6.364-6.364L12 7.636l-1.318-
1.318a4.5 4.5 0 00-6.364 0z" />
              </svg>
            </button>
          </div> </div> </div> </div> );

```

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		77

Додаток В. Лістинг файлу «FilterSidebar.tsx»

```
'use client';
import React, { useEffect, useState } from 'react';
import { useRouter, useSearchParams, usePathname } from 'next/navigation';
import axios from 'axios';
import { ApiResponse } from '@/lib/types';
interface Category {
  _id: string;
  name: string;
}
const FilterSidebar = () => {
  const router = useRouter();
  const pathname = usePathname();
  const searchParams = useSearchParams();
  const [categories, setCategories] = useState<Category[]>([]);
  const [minPrice, setMinPrice] = useState(searchParams.get('minPrice') ||
  '');
  const [maxPrice, setMaxPrice] = useState(searchParams.get('maxPrice') ||
  '');
  const [selectedCategory, setSelectedCategory] =
  useState(searchParams.get('category') || '');
  useEffect(() => {
    const fetchCategories = async () => {
      try {
        const res = await
        axios.get<ApiResponse<Category[]>>('/api/categories');
        if (res.data.success) {
          setCategories(res.data.data);
        }
      } catch (err) {
        console.error('Failed to fetch categories');
      }
    };
  });
};
```

										Арк.
										78
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.423.15.00.00 ПЗ					

```

    fetchCategories();
  }, []);
const applyFilters = () => {
  const params = new URLSearchParams(searchParams.toString());

  if (minPrice) params.set('minPrice', minPrice);
  else params.delete('minPrice');

  if (maxPrice) params.set('maxPrice', maxPrice);
  else params.delete('maxPrice');

  if (selectedCategory) params.set('category', selectedCategory);
  else params.delete('category');
  router.push(`${pathname}?${params.toString()}`);
};
const handleCategoryChange = (categoryId: string) => {
  const newCategory = selectedCategory === categoryId ? '' :
categoryId;
  setSelectedCategory(newCategory);

  const params = new URLSearchParams(searchParams.toString());
  if (newCategory) params.set('category', newCategory);
  else params.delete('category');
  router.push(`${pathname}?${params.toString()}`);
};
return (
  <div className="w-full md:w-64 p-4 bg-white rounded-lg shadow-sm
border border-gray-100 h-fit">
    <h3 className="text-xl font-semibold mb-4 border-b pb-
2">Фільтри</h3>
    {/* Price Filter */}
    <div className="mb-6">
      <h4 className="font-semibold text-gray-700 mb-3">Ціна
(грн)</h4>

```

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

```

<div className="flex items-center space-x-2 mb-3">
  <input
    type="number"
    placeholder="Від"
    value={minPrice}
    onChange={(e) => setMinPrice(e.target.value)}
    className="w-1/2 px-3 py-2 border border-gray-300
rounded-md focus:outline-none focus:ring-2 focus:ring-green-500"
  />
  <span>-</span>
  <input
    type="number"
    placeholder="До"
    value={maxPrice}
    onChange={(e) => setMaxPrice(e.target.value)}
    className="w-1/2 px-3 py-2 border border-gray-300
rounded-md focus:outline-none focus:ring-2 focus:ring-green-500"
  />
</div>
<button
  onClick={applyFilters}
  className="w-full py-2 bg-gray-100 text-gray-700
rounded-md hover:bg-gray-200 transition text-sm font-medium"
  >
  Застосувати ціну
</button>
</div>
{/* Category Filter */}
<div className="mb-6">
  <h4 className="font-semibold text-gray-700 mb-
3">Категорії</h4>
  <div className="space-y-2 max-h-60 overflow-y-auto pr-2">
    {categories.map((cat) => (
      <label key={cat._id} className="flex items-center

```

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		80

```

cursor-pointer group">
    <input
      type="checkbox"
      checked={selectedCategory === cat._id}
      onChange={() =>
handleCategoryChange(cat._id)}
      className="form-checkbox h-4 w-4 text-
green-600 rounded border-gray-300 focus:ring-green-500 cursor-pointer"
    />
    <span className="ml-2 text-gray-600 group-
hover:text-green-600 transition">{cat.name}</span>
  </label>
  )}}
</div>
</div>
<button
  onClick={() => {
    setMinPrice('');
    setMaxPrice('');
    setSelectedCategory('');
    router.push(pathname);
  }}
  className="w-full py-2 border border-red-200 text-red-600
rounded-md hover:bg-red-50 transition text-sm font-medium"
  >
  Скинути фільтри
</button>
</div>
);
};

export default FilterSidebar;

```

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

Додаток Г. Лістинг файлу «User.ts»

```
import mongoose, { Schema, Document } from 'mongoose';
import bcrypt from 'bcryptjs';
export interface IUser extends Document {
  name: string;
  email: string;
  phone?: string;
  password?: string;
  role: 'user' | 'admin';
  createdAt: Date;}
const UserSchema: Schema = new Schema({
  name: { type: String, required: true },
  email: { type: String, unique: true, sparse: true, match:
[/^\S+@\S+\.\S+$/, 'Please provide a valid email'] },
  phone: { type: String, unique: true, sparse: true },
  password: { type: String, required: true, minlength: 6, select: false },
  role: { type: String, enum: ['user', 'admin'], default: 'user' },
  createdAt: { type: Date, default: Date.now },});
(UserSchema as any).pre('validate', function(this: any, next: (err?: any) =>
void) {
  if (!this.email && !this.phone) {
    this.invalidate('email', 'Email or Phone is required');
    this.invalidate('phone', 'Email or Phone is required'); }
  next();
});
(UserSchema as any).pre('save', async function (this: any, next: (err?: any)
=> void) {
  if (!this.isModified('password')) return next();
  const salt = await bcrypt.genSalt(10);
  this.password = await bcrypt.hash(this.password, salt);
  next();});
export default mongoose.models.User || mongoose.model<IUser>('User',
UserSchema);
```

					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82

Додаток Г. Лістинг файлу «Product.ts»

```
import mongoose, { Schema, Document } from 'mongoose';

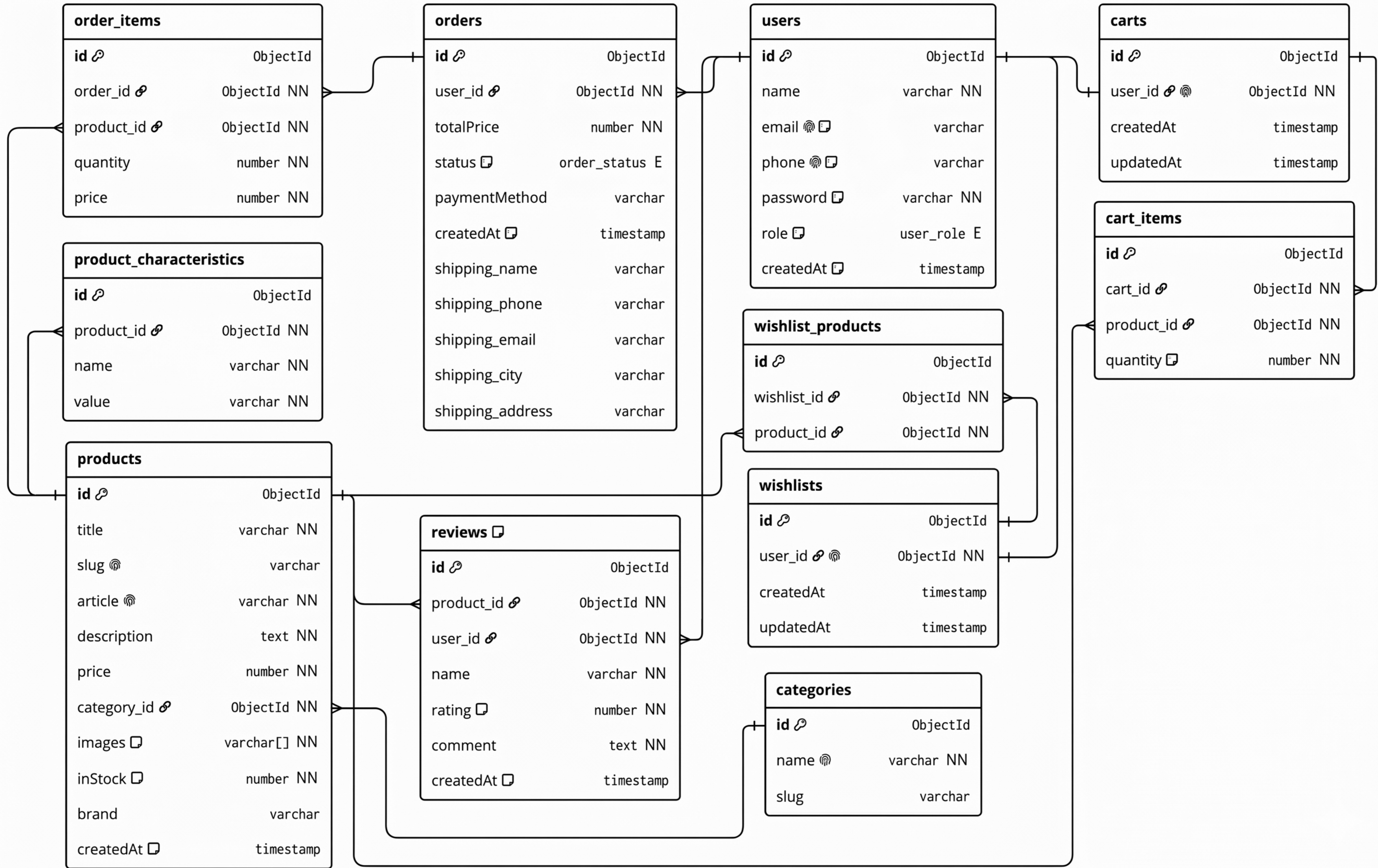
export interface IProduct extends Document {
  title: string;
  slug?: string;
  article: string;
  description: string;
  price: number;
  category: mongoose.Types.ObjectId;
  characteristics: { name: string; value: string }[];
  images: string[];
  inStock: number;
  brand?: string;
  createdAt: Date;
}

const ProductSchema: Schema = new Schema({
  title: { type: String, required: true, trim: true },
  slug: { type: String, unique: true },
  article: { type: String, required: true, unique: true },
  description: { type: String, required: true },
  price: { type: Number, required: true },
  category: { type: Schema.Types.ObjectId, ref: 'Category', required: true },
  characteristics: [{ name: String, value: String }],
  images: [{ type: String, required: true }],
  inStock: { type: Number, required: true, default: 0 },
  brand: String,
  createdAt: { type: Date, default: Date.now },
});

export default mongoose.models.Product ||
mongoose.model<IProduct>('Product', ProductSchema);
```

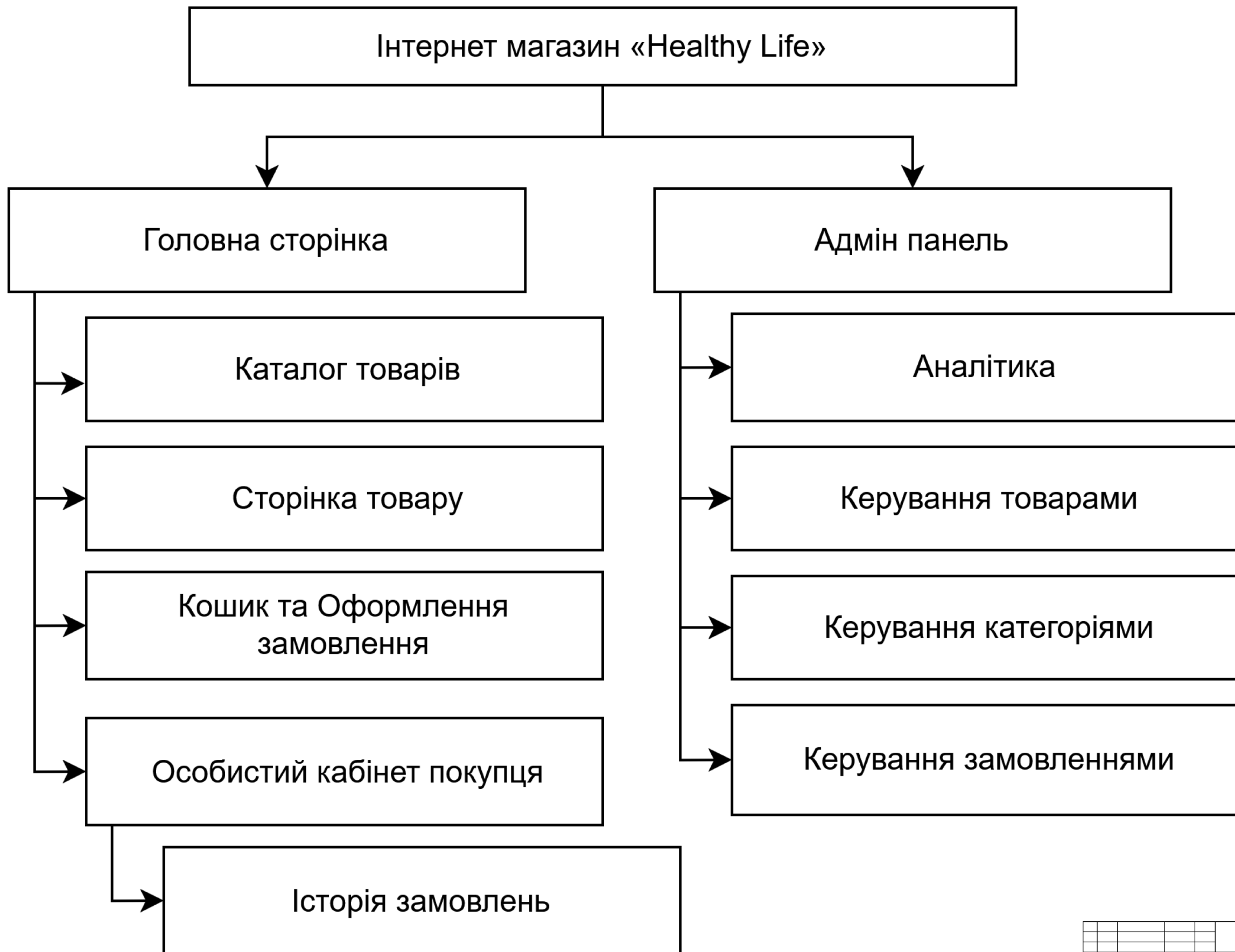
					2026.KBP.122.423.15.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

ER-діаграма бази даних



Лист № 1
Лист № 2
Лист № 3
Лист № 4
Лист № 5
Лист № 6
Лист № 7
Лист № 8
Лист № 9
Лист № 10

Схема структурна інтернет-магазину



				2026.KBP.122.4.23.15.00.00 CC			
Ім'я	Лист	№ докум.	Підп.	Дата	Розробка інтернет-магазину товарів для здорового способу життя «Healthy Life»	Лист	Масштаб
Розроб		Олешиук М.І.					
Проб		Цимбалюк Л.В.					
Ісконтр					Схема структурна інтернет-магазину	Лист	Листов 1
Рецензент							
Ісконтр		Приймак В.А.				ВСП ТФК ТНТУ	КН-4.23
Члб						м. Тернопіль	
						Формат	A1

Таблиця техніко-економічних показників

№	Показник	Одиниця виміру	Значення
1	Мова програмування	-	TypeScript
2	Основний веб-фреймворк	-	Next.js (App Router, v16.2)
3	CSS-фреймворк для стилізації інтерфейсу	-	Tailwind CSS (v4.0)
4	Бібліотека керування станом	-	Zustand (v5.0)
5	Система управління базами даних	-	MongoDB
6	Кількість колекції (таблиць) у базі даних	штук	7
7	Хмарний сервіс для збереження зображень	-	Cloudinary
8	Кількість кінцевих точок (API Endpoints)	штук	10
9	Содівартість	грн	94 876
10	Плановий грошовий потік	грн	45 000
11	Ціна	грн	148 700
12	Чиста теперішня вартість	грн	17 032
13	Термін окупності	грн	2,4