

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 вебсайту магазину спортивного взуття «Comfort»”

Виконав: студентка IV курсу, групи КН-423

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

Микола МІРОШНИЧЕНКО

(ім'я та прізвище)

Керівник Людмила ЦИМБАЛЮК

(ім'я та прізвище)

Рецензент _____

(ім'я та прізвище)

Тернопіль – 2026

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

_____ Мірошниченку Миколі Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебсайту магазину спортивного взуття «Comfort»,

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 вебсайту

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 проекту

2.1 Розробка структури вебсайту і вебсторінок

2.2 Створення та верстка сторінок вебсайту

2.3 Розробка структури бази даних вебсайту

2.4 Програмування вебсайту

2.4.1 Написання клієнтської частини

2.4.2 Написання серверної частини

2.5 Тестування вебсайту

3 Спеціальний розділ

3.1 Інструкція з розгортання м вебсайту в Інтернеті

3.2 Інструкція з наповнення вебсайту

3.3 Інструкція з популяризації та підтримки вебсайту

4 Економічний розділ

4.1 Визначення стадій технологічного процесу та загальної тривалості

проведення розробки вебсайту

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 вебсайту

4.8 Визначення економічної ефективності і терміну окупності капітальних

вкладень

5 Охорона праці, техніка безпеки та екологічні вимоги

5.1 Професійні захворювання та травми

5.2 Види електричних травм

6 Висновки

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – вебсайту «Comfort».

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)

Микола МІРОШНИЧЕНКО

Керівник роботи

(підпис)

Людмила ЦИМБАЛЮК

ЗМІСТ

АНОТАЦІЯ.....	7
ABSTRACT.....	8
ВСТУП.....	9
1 ЗАГАЛЬНИЙ РОЗДІЛ.....	11
1.1 Аналітичний огляд існуючих рішень.....	11
1.2 Технічне завдання.....	15
1.2.1 Найменування та область застосування.....	15
1.2.2 Призначення розробки.....	15
1.2.3 Вимоги до функціоналу web-сайту.....	16
1.2.4 Вимоги до програмної документації.....	19
1.2.5 Техніко-економічні показники.....	20
1.2.6 Стадії та етапи розробки.....	22
1.2.7 Порядок контролю та прийому.....	24
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.....	36
2.4 Програмування серверної частини сайту.....	40
2.5 Тестування web-сайту.....	46
3 СПЕЦІАЛЬНИЙ РОЗДІЛ.....	51
3.1 Інструкція з розміщення сайту в Інтернеті.....	51
3.2 Інструкція з обслуговування та наповнення сайту.....	54
3.3 Інструкція з популяризації та підтримки сайту.....	58

					<i>2026.КВР.122.423.14.00.00 ПЗ</i>		
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка вебсайту магазину спортивного взуття «Comfort» Пояснювальна записка		
Розроб.		Мірошніченко М.М.					
Перевір.		Цимбалюк Л.В.					
Реценз.							
Н. Контр.		Приймак В.А.					
Затверд.					Літ.	Арк.	Архивів
						5	
					ВСП ТФК ТНТУ КН-423 м. Тернопіль		

4	ЕКОНОМІЧНИЙ РОЗДІЛ.....	60
4.1	Визначення стадій технологічного процесу та загальної тривалості проведення розробки вебсайту.	60
4.2	Визначення витрат на оплату праці та відрахувань на соціальні заходи	61
4.3	Розрахунок витрат на електроенергію	63
4.4	Розрахунок суми амортизаційних відрахувань вебсайту	64
4.5	Обчислення накладних витрат	65
4.6	Складання кошторису витрат та визначення собівартості вебсайту	65
4.7	Розрахунок ціни вебсайту	66
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень	66
5	ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ	68
5.1	Професійні захворювання та травми.	68
5.2	Види електричних травм	69
	ВИСНОВКИ.....	72
	ПЕРЕЛІК ПОСИЛАНЬ	74
	ДОДАТКИ.....	76
	Додаток А. Текст головного модуля сервера «main.ts»	76
	Додаток Б. Схема моделі даних користувача «user.schema.ts»	77
	Додаток В. Логіка сервісу авторизації «service.ts».....	78
	Додаток Г. Сервіс обробки замовлень «orders.service.ts».....	79
	Додаток Д. Сховище управління станом кошика «useCartStore.ts».....	80

АНОТАЦІЯ

Тема кваліфікаційної роботи: Розробка вебсайту магазину спортивного взуття «Comfort».

Метою кваліфікаційної роботи є проектування та програмна реалізація сучасного, інтуїтивно зрозумілого інтернет-магазину спортивного взуття, що дозволить автоматизувати процес продажів, забезпечить зручний пошук товарів для клієнтів та оптимізує управління замовленнями.

Пояснювальна записка складається з п'яти розділів.

У загальній частині проведено аналітичний огляд ринку аналогічних рішень електронної комерції та детальний аналіз технічного завдання. Визначено ключові вимоги до функціоналу магазину спортивного взуття.

У другому розділі висвітлено безпосередній процес розробки вебсайту. Описано та обґрунтовано вибір архітектури системи, структури бази даних та методів організації інформаційних потоків. Подано опис створення клієнтської частини (frontend) з її компонентами, реалізації серверної частини (backend) та API інтернет-магазину, а також наведено етапи тестування й налагодження коду.

У спеціальній частині викладено етапи розгортання вебсайту на хостингу. Складено інструкції для наповнення каталогу товарів і обслуговування системи, а також рекомендації з просування та підтримки вебсайту.

В економічному розділі здійснено розрахунок собівартості розробки програмного продукту, визначено обсяг капіталовкладень та проведено оцінку техніко-економічної ефективності впровадження сайту.

У п'ятому розділі розглянуто два питання охорони праці, що стосуються професійних захворювання та видів електричних травм.

Обсяг пояснювальної записки становить 80 сторінок формату А4.

До складу кваліфікаційної роботи входить графічна частина, яка виконана на окремих аркушах формату А1 і містить: таблицю техніко-економічних показників, діаграму варіантів використання, структурну схему інтерфейсу вебсайту, схему бази даних.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

Thesis Topic: Development of a Website for the “Comfort” Sports Footwear Store.

The objective of this thesis is to design and develop a modern, intuitive online store for sports footwear that will automate the sales process, provide customers with a convenient product search, and optimize order management.

The explanatory note consists of five chapters.

The general section provides an analytical overview of the market for similar e-commerce solutions and a detailed analysis of the technical specifications. The key requirements for the functionality of the sports shoe store are identified.

The second chapter describes the actual process of developing the website. It describes and justifies the choice of system architecture, database structure, and methods for organizing information flows. It provides a description of the creation of the frontend with its components, the implementation of the backend and the online store’s API, and outlines the stages of testing and debugging the code.

The special section outlines the steps for deploying the website on a hosting platform. Instructions have been prepared for populating the product catalog and maintaining the system, as well as recommendations for promoting and supporting the website.

The economic section calculates the cost of developing the software product, determines the amount of capital investment, and assesses the technical and economic efficiency of the website’s implementation.

The fifth section addresses two occupational safety issues related to occupational diseases and types of electrical injuries.

The explanatory note consists of 80 A4-sized pages.

The thesis includes a graphical section, which is presented on separate A1-sized sheets and contains: a table of technical and economic indicators, a diagram of usage options, a structural diagram of the website interface, and a database diagram.

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

У зв'язку з активним переходом роздрібної торгівлі у цифровий простір та швидкою зміною споживчих звичок, побудова ефективних інструментів для цифрової взаємодії з покупцями стає базовою умовою стабільного функціонування сучасного бізнесу. Якість контактів стає іншою тепер. Через високу конкуренцію на ринку спортивного взуття компаніям доводиться шукати нові методи утримання аудиторії, пропонуючи персоналізований досвід та високу швидкість обробки замовлень у реальному часі.

Завдяки використанню фреймворків Next.js [6] та NestJS, з'являється можливість розробляти масштабовані веб-застосунки, що витримують значні навантаження під час пікових продажів. Важливо підтримувати гнучкість структури завжди. Шляхом впровадження серверного рендерингу та методів оптимізації зображень вдається покращити показники пошукової видачі, що безпосередньо впливає на прибутковість комерційного ресурсу в довгостроковій перспективі.

Для магазинів з фізичними точками продажу критично важливою залишається синхронізація залишків товарів та підтримка гібридних методів отримання покупок, зокрема самовивозу. Користувачу це забезпечує необхідний комфорт. Реалізація функціоналу перевірки наявності розмірів у реальному часі допомагає уникнути негативного досвіду клієнтів та підвищує рівень довіри до торговельної марки через прозорість інформації.

З метою захисту конфіденційної інформації та гарантування безпеки фінансових операцій у роботі застосовано сучасні протоколи авторизації на основі механізму JSON Web Tokens. Щодо безпеки даних, тут не може бути компромісів. Окрім технічної стійкості, увагу приділено проектуванню інтерфейсу, який повинен бути інтуїтивно зрозумілим незалежно від рівня підготовки відвідувача сайту. Доступним він має бути кожному.

Під час розробки фронтенд-частини виникає потреба у ефективному управлінні станом кошика та збереженні даних без зайвого перевантаження

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

сторінок інформацією з сервера. Легким та швидким має бути обране рішення. Використання сучасних бібліотек для стейт-менеджменту дозволяє забезпечити миттєву реакцію інтерфейсу на дії користувача, що значно покращує загальне сприйняття програмного продукту.

В основу збереження даних покладено нереляційну модель MongoDB, яка забезпечує високу гнучкість структури товарів та дозволяє легко вносити зміни у характеристики взуття. Пріоритетом тут виступає гнучкість схем. Завдяки цьому адміністратор може оперативно додавати нові категорії або змінювати набір атрибутів для окремих моделей без необхідності складної перебудови всієї структури таблиць.

У роботі здійснюється розробка комплексного програмного рішення для магазину «Comfort», що охоплює як клієнтську, так і серверну частини. Багато складних модулів поєднує цей проєкт. Метою роботи визначено створення функціонального веб-застосунку, який оптимізує процеси вибору, замовлення та реалізації взуття для кінцевого користувача. Зручним та практичним має вийти результат.

З огляду на вимоги до сучасних платформ, особливу увагу приділено адаптивності інтерфейсу для коректного відображення на мобільних пристроях з різною роздільною здатністю екрана. Домінуючим зараз виступає саме мобільний трафік. Оптимізація коду та мінімізація завантажуваних ресурсів дозволяють досягти високих показників швидкодії, що критично важливо для конверсії відвідувачів у постійних покупців.

Завдяки впровадженню розробленого програмного забезпечення у діяльність реального бізнесу вдається досягти повної автоматизації торговельних операцій та суттєво спростити повсякденну роботу персоналу торговельного залу. Це значно прискорює всі процеси. Поєднання технологій NestJS та Next.js у межах одного проєкту дозволяє досягти синергії між високою продуктивністю сервера та чудовим досвідом користувача, що підтверджує актуальність проведеної роботи.

					<i>2026.KBP.122.423.14.00.00 ПЗ</i>	Арк.
						10
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

У сучасній практиці побудови систем електронної комерції особлива увага приділяється вибору оптимальної стратегії розробки, яка б забезпечувала гнучкість налаштувань та швидкість масштабування бізнес-процесів. Це стає критично важливою умовою сьогодні. Під час проєктування нового програмного виробу [15] зазвичай орієнтуються на існуючі прототипи, оскільки їхня експлуатація дозволяє виявити типові помилки в архітектурі та інтерфейсі. Додатковим джерелом знань виступає аналіз спеціалізованої літератури та вивчення документації сучасних фреймворків для розробки високонавантажених систем.

В основу кваліфікаційної роботи покладено аналіз технічного завдання, у якому чітко визначено відмінності майбутнього продукту від готових шаблонних рішень. Важливо розуміти потреби бізнесу. При промисловому проєктуванні вивчення ринку починається з огляду вже реалізованих систем, що допомагає сформулювати логічні схеми взаємодії компонентів. Традиційні підходи до створення онлайн-магазинів сьогодні розділяються на використання конструкторів та розробку власних унікальних проєктів на базі сучасного стеку технологій.

Варто зауважити, що готові платформи (наприклад, Shopify або Wix) мають суттєві обмеження у кастомізації складної бізнес-логіки. Гнучкість тут є досить низькою. Для великих магазинів взуття, де необхідно підтримувати специфічні сітки розмірів та складні умови доставки, перевага зазвичай надається власним розробкам. Це дозволяє повністю контролювати безпеку даних та продуктивність серверної частини.

Для детального аналізу було обрано два популярні майданчики, які спеціалізуються на продажі взуття та використовують власні програмні рішення замість стандартних шаблонів.

Першим об'єктом дослідження став інтернет-магазин Intertop, його головна сторінка зображена на рисунку 1.1. Дана платформа є прикладом складного

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

багатомодульного застосунку з розвинуеною системою фільтрації та інтеграцією з програмами лояльності.

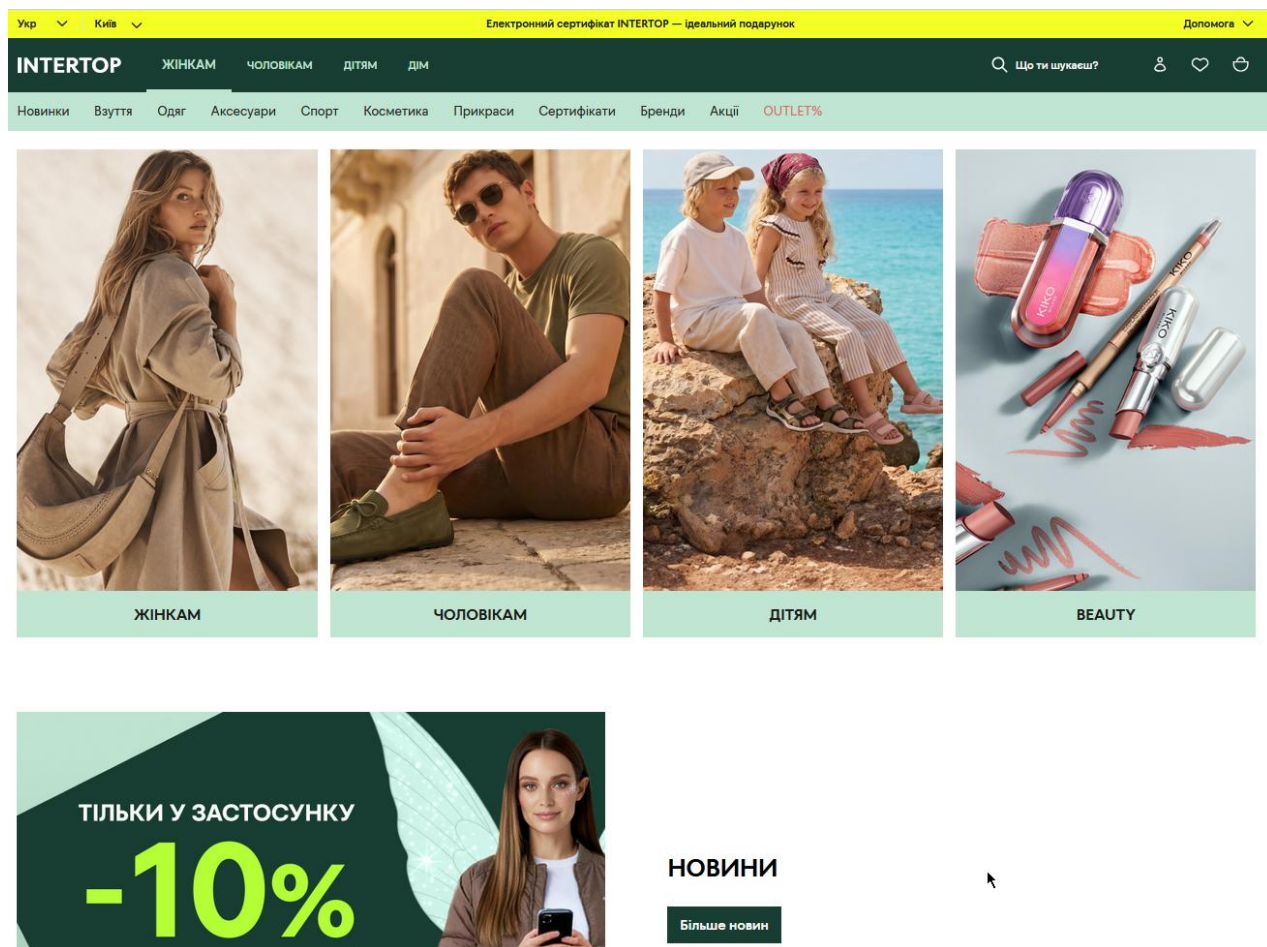


Рисунок 1.1 – Інтерфейс головної сторінки інтернет-магазину Intertop

На основі аналізу Intertop було виявлено такі переваги:

- наявність глибокої системи фільтрації за багатьма параметрами (бренд, сезон, матеріал, розмір);
- стабільна робота під час великого напливу користувачів завдяки оптимізованій серверній архітектурі;
- зручна синхронізація з фізичними точками видачі товарів по всій країні.

До недоліків цього рішення можна віднести надмірну завантаженість головної сторінки візуальними елементами, що може сповільнювати взаємодію на слабких пристроях. Інтерфейс виглядає дещо перевантаженим іноді. Окрім того,

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

процес оформлення замовлення містить забагато кроків, що збільшує ймовірність відмови покупця на фінальному етапі.

Другим аналогом було обрано спеціалізований магазин Sneakerstudio Його сторінка з карткою товару зображена на рисунку 1.2. Цей ресурс орієнтований на вузьку нішу преміального взуття та вирізняється акцентом на візуальному поданні товару.

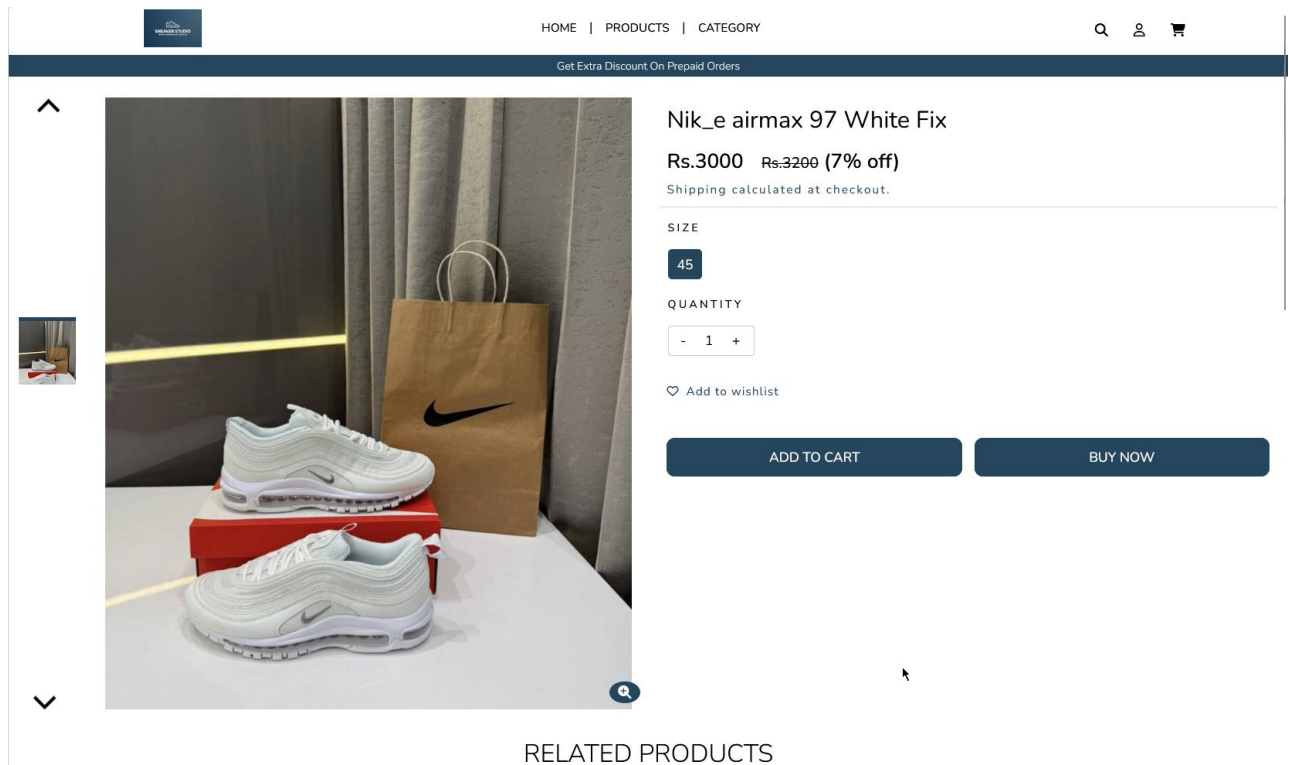


Рисунок 1.2 – Картка товару в магазині Sneakerstudio

Серед позитивних сторін даного рішення виділяються наступні:

- висока якість графічного контенту та наявність 360-градусних оглядів моделей;
- швидка робота клієнтської частини завдяки використанню сучасних SPA-фреймворків;
- лаконічний дизайн, який не відволікає від процесу вибору взуття.

Недоліки Sneakerstudio проявляються у менш гнучкій системі пошуку порівняно з великими маркетплейсами. Пошук працює не завжди ідеально. Також

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

було помічено проблеми з відображенням деяких елементів кошика при використанні застарілих версій мобільних браузерів.

У табл. 1.1 наведено порівняння функціональних можливостей розглянутих аналогів та розроблюваного проєкту «Comfort».

Таблиця 1.1 – Порівняння функціональних можливостей

Критерій порівняння	Intertop	Sneakerstudio	Магазин «Comfort»
Технологічна основа	Custom Engine	React/SPA	Next.js 14 / NestJS
Швидкість завантаження	Середня	Висока	Висока (SSR)
Гнучкість фільтрів	Дуже висока	Середня	Висока
Оформлення замовлення	Складне	Просте	Максимально спрощене

Після вивчення існуючих рішень стає зрозумілою доцільність створення власного проєкту на базі Next.js та NestJS. Це забезпечить високу швидкість роботи. Застосування серверного рендерингу (SSR) дозволить поєднати швидкість реакції інтерфейсу з відмінними показниками індексації сайту пошуковими системами. На відміну від закритих платформ, власний код дає можливість реалізувати специфічний алгоритм обробки замовлень для опції самовивозу, що є ключовою вимогою технічного завдання для магазину «Comfort».

За рахунок використання нереляційної бази даних MongoDB вдасться досягти гнучкості у зберіганні інформації про різні типи взуття. Схеми даних будуть динамічними. Такий підхід робить систему стійкою до майбутніх змін у каталозі та дозволяє легко інтегрувати нові модулі аналітики або маркетингові інструменти.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

У межах представленої кваліфікаційної роботи здійснюється розробка програмного продукту під повною назвою «Comfort – інформаційна система керування продажами спортивного взуття з інтегрованим модулем електронної комерції». Для зручності ідентифікації використовується стисла назва проєкту — «Comfort». Створений веб-застосунок забезпечує автоматизацію торговельних операцій, управління складськими залишками та безпосередню взаємодію з покупцями через сучасний цифровий інтерфейс. Він працює надзвичайно швидко.

Щодо сфери використання, розроблене програмне рішення орієнтоване на роздрібну торгівлю спортивними товарами, електронну комерцію та організацію гібридних схем продажу з підтримкою фізичних точок самовивозу. Це значно полегшує логістику. У якості головного об'єкта впровадження виступає діюче торговельне підприємство, яке потребує розширення каналів збуту шляхом створення стабільного онлайн-представництва з інтегрованими сервісами оплати та доставки.

Завдяки впровадженню адаптивного дизайну, експлуатація системи передбачається на широкому спектрі пристроїв, включаючи персональні комп'ютери, ноутбуки, планшети та смартфони з доступом до мережі Інтернет. Усі браузери підтримують застосунок. Під час практичного використання основне призначення виробу полягає у забезпеченні безперервного циклу продажу взуття від моменту перегляду каталогу до фіксації фінального статусу замовлення.

1.2.2 Призначення розробки

Під час цифровізації торговельних процесів основним призначенням розробки виступає надання користувачам надійного інструменту для вибору та придбання спортивного взуття через мережу Інтернет. Це робить покупки

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

зручнішими. Шляхом впровадження сучасних технологій NestJS та Next.js вдається створити середовище, що забезпечує безперебійну обробку клієнтських запитів та високу швидкість відгуку інтерфейсу. Пріоритетом тут виступає продуктивність.

Для досягнення поставлених бізнес-цілей у межах проєкту передбачено реалізацію наступних функціональних можливостей:

- за допомогою гнучкої системи фільтрації забезпечується швидкий пошук моделей за розміром, брендом та ціновим діапазоном, що значно скорочує шлях клієнта до здійснення покупки;
- через використання нереляційної моделі даних MongoDB організовано детальне представлення карток товарів із повним описом технічних характеристик та фотогалереєю кожної пари взуття;
- шляхом впровадження механізму JWT-авторизації [13] гарантується безпека персональних профілів та цілісність історії замовлень кожного зареєстрованого відвідувача магазину «Comfort»;
- за рахунок створення окремої адміністративної панелі досягається повний контроль над складськими залишками та актуалізацією цін у режимі реального часу.

Такий підхід дозволяє бізнесу розвиватися.

У практичному аспекті використання системи дозволяє автоматизувати процес оформлення замовлень, включаючи вибір методів доставки та опцію самовивозу з фізичного магазину «Comfort». Це суттєво полегшує логістику. Завдяки модульній структурі серверної частини на базі NestJS з'являється можливість подальшого розширення функціоналу, наприклад, через підключення аналітичних сервісів або нових платіжних шлюзів. В основу покладено гнучкість архітектури.

1.2.3 Вимоги до функціоналу web-сайту

Під час проєктування архітектури веб-застосунку було визначено ключові функціональні вимоги, реалізація яких дозволяє розв'язати завдання з

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

автоматизації роздрібної торгівлі спортивним взуттям. Це основа всього проєкту. Шляхом впровадження чіткої рольової моделі доступу забезпечується розмежування прав користувачів та надійний захист конфіденційних даних адміністративної частини магазину.

Для реалізації повноцінного циклу електронної комерції передбачено роботу наступних категорій користувачів:

– Гість (Незарєєстрований користувач): отримує можливість вільного перегляду каталогу, використання інструментів фільтрації товарів за брендом чи розміром та оформлення замовлення без обов'язкового створення облікового запису.

– Клієнт (Авторизований користувач): може редагувати персональний профіль, переглядати повну історію своїх покупок, зберігати обрані моделі у списку бажаного та залишати оцінки після отримання товару.

– Адміністратор або Менеджер: здійснює оперативне керування асортиментом через CRUD-операції, контролює фактичні складські залишки та змінює статуси замовлень для інформування покупців про етапи доставки.

Права доступу суворо розмежовані.

У межах розробки інтерфейсу визначено перелік обов'язкових сторінок, які забезпечують зручну навігацію та логічну послідовність дій покупця:

– Головна сторінка: містить інтерактивні блоки з акційними пропозиціями, сітку категорій та інформаційні розділи про фізичний магазин «Comfort».

– Сторінка каталогу: забезпечує відображення списку взуття з панеллю фільтрації за ціною, кольором та іншими характеристиками.

– Картка товару: надає вичерпну інформацію про модель, включаючи фотогалерею та динамічний вибір наявних розмірів.

– Кошик користувача: дозволяє керувати кількістю доданих одиниць товару та автоматично розраховує загальну вартість покупки.

– Сторінка оформлення замовлення: містить форми для введення

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

контактних даних та вибору методів доставки.

– Особистий кабінет та Адмін-панель: забезпечують персоналізоване керування даними та контентом.

Діаграма варіантів використання зображено на плакаті 2026.КВР.122.423.14.00.00 ДВ.

Для забезпечення коректної взаємодії компонентів вхідними даними виступають параметри пошуку, облікові дані користувачів та об'єкти замовлень у форматі JSON. Дані передаються через API. Натомість вихідна інформація подається відвідувачам у вигляді динамічно відрендерених сторінок каталогу, актуальних залишків товарів та системних сповіщень про успішність транзакцій.

З метою підвищення надійності функціонування реалізовано контроль валідності вхідних параметрів на стороні сервера за допомогою спеціалізованих декораторів NestJS. Це запобігає критичним збоєм. Захист від несанкціонованого доступу базується на використанні токенів авторизації, що зберігаються у захищених httpOnly cookies для мінімізації ризиків викрадення даних через XSS-атаки.

Щодо умов експлуатації, програмний продукт має стабільно функціонувати в середовищі сучасних веб-браузерів [14] на базі двигуна Chromium, а також у Safari та Firefox. Сумісність є пріоритетом тут. Оптимальна конфігурація сервера повинна підтримувати одночасну роботу щонайменше 100 активних користувачів без суттєвого зниження швидкості реакції інтерфейсу під час пікових навантажень.

Згідно з сучасними стандартами продуктивності, час завантаження інтерактивних елементів сторінки при першому запиті не повинен перевищувати 2 секунди. Швидкість безпосередньо впливає на конверсію. Обробка складних API-запитів, пов'язаних із фільтрацією великого масиву взуття в базі MongoDB, має відбуватися протягом 300-500 мілісекунд для забезпечення комфортної роботи користувача.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.4 Вимоги до програмної документації

Під час розробки інформаційної системи «Comfort» особлива увага приділяється формуванню повної програмної документації, яка відповідає чинним державним стандартам та забезпечує цілісність подальшого супроводу проєкту. Це гарантує стабільність. Склад документів визначається складністю архітектури на базі NestJS та Next.js, а також призначенням продукту як комерційного інструменту для автоматизації роздрібно́ї торгівлі.

Для забезпечення ефективної експлуатації та можливості майбутнього масштабування інтернет-магазину до складу програмної документації включено наступні матеріали:

- Технічне завдання: визначає сукупність вимог до функціоналу, надійності та умов експлуатації системи;
- Пояснювальна записка: містить детальний опис архітектурних рішень, структури бази даних MongoDB та логіки взаємодії компонентів;
- Текст програми: оформлений вихідний код із дотриманням стандартів типізації TypeScript [5];
- Керівництво оператора: детальна інструкція для клієнтів щодо процесу вибору взуття, реєстрації та оформлення замовлень;
- Керівництво системного програміста: технічний посібник для адміністраторів щодо розгортання серверної частини та моніторингу стану системи;
- Опис API-ендпоінтів: перелік усіх доступних методів взаємодії між фронтендом та бекендом із описом параметрів запитів.

Це спрощує підтримку.

Шляхом впровадження суворих стандартів написання коду забезпечується легке розуміння логіки роботи кожного окремого модуля системи іншими розробниками або технічним персоналом. Коментарі мають бути стислими. Кожен програмний модуль або процедурний блок повинен мати початковий блок

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

коментарів [10] , який містить:

- призначення (короткий опис функціональної задачі, яку вирішує даний блок коду);
- склад вхідних та вихідних даних (опис типів аргументів та структури об'єктів, що повертаються);
- обмеження та умови використання (вимоги до середовища виконання або наявності специфічних змінних оточення);
- дату введення в дію та інформацію про останнє оновлення.

Такий підхід дисциплінує розробника.

Відповідно до принципів структурного програмування, кожен модуль системи «Comfort» має чітко визначену одну точку входу та одну точку виходу для забезпечення передбачуваності виконання коду. Це підвищує надійність. Для успішного розгортання та тривалої підтримки веб-застосунку розроблено низку спеціалізованих інструкцій:

- інструкція з деплою: описує процес розміщення компонентів на серверних потужностях (наприклад, Vercel або Render) та налаштування доменних імен;
- посібник із налаштування бази даних: містить вказівки щодо створення колекцій у MongoDB та забезпечення з'єднання через Mongoose;
- регламент оновлення контенту: описує механізм додавання нових моделей взуття та актуалізації складських залишків через адміністративну панель;
- стратегія резервного копіювання: надає рекомендації щодо регулярного збереження стану бази даних для запобігання втраті критичної інформації про замовлення.

Це робить систему життєздатною.

1.2.5 Техніко-економічні показники

Для об'єктивного визначення обсягу ресурсів, необхідних для створення та успішного впровадження інформаційної системи «Comfort», виконується детальний розрахунок основних техніко-економічних показників розробки, що

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

дозволяє раціонально планувати етапи проєктування. Це дозволяє ефективно планувати роботу. Через складність архітектурних рішень на базі NestJS та потребу у високій якості кінцевого продукту, оцінка витрат інтелектуальних та часових ресурсів виконавця стає обов'язковим етапом перед початком написання коду. Ресурси використовуються максимально продумано та раціонально. Загальний обсяг трудових витрат становить орієнтовно 2.5–3.5 людино-місяці, що повністю відповідає циклу створення комерційного веб-застосунку в межах дипломного проєкту для одного розробника.

Під час реалізації проєкту передбачено виконання наступних етапів трудової активності:

- за допомогою аналізу предметної області та детального вивчення технічного завдання формується повний перелік функціональних вимог до майбутньої системи для забезпечення її відповідності бізнес-завданням;

- шляхом проєктування динамічних схем MongoDB та архітектури серверних модулів NestJS створюється надійна логічна структура бази даних, яка витримує значні навантаження під час пікових продажів;

- завдяки використанню фреймворку Next.js 14 та стилізації засобами TailwindCSS реалізується клієнтська частина з адаптивним дизайном для зручного перегляду каталогу взуття на мобільних пристроях;

- через написання сервісів, контролерів та логіки захисту даних на стороні сервера забезпечується обробка замовлень, безпечна авторизація користувачів та стабільна робота API-інтерфейсів;

- шляхом проведення комплексного модульного та інтеграційного тестування виявляються та оперативно усуваються помилки в роботі програмного коду для гарантування високої якості кінцевого цифрового продукту;

- за допомогою офісних пакетів та текстових редакторів готується повний комплект програмної, технічної та експлуатаційної документації згідно з державними стандартами та вимогами до кваліфікаційних робіт.

- Шляхом розробки клієнтської частини, яка охоплює побудову

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

адаптивного інтерфейсу та управління станом через Zustand, витрачається приблизно 1.0 людино-місяця робочого часу розробника. Взаємодія компонентів налаштовується дуже швидко. Під час створення серверної логіки на основі фреймворку NestJS та налаштування безпеки авторизації через механізм JWT, витрати ресурсів зазвичай становлять 1.2 людино-місяця. Це забезпечує стабільність бекенду. Для завершального етапу тестування, виправлення помилок та підготовки супровідної документації виділяється ще 0.5 людино-місяця трудових витрат, що дозволяє випустити готовий до експлуатації продукт.

Витрати машинного часу охоплюють експлуатацію апаратних потужностей та хмарних сервісів, необхідних для повноцінної розробки, налагодження та фінального розгортання проєкту в мережі Інтернет. Обладнання працює практично безперервно. Загальний обсяг становить приблизно 140–190 годин реальної роботи обладнання, що включає використання персонального комп'ютера, локальних серверів Node.js та систем контролю версій Git. Ефективність використання техніки доведена. Завдяки розгортанню бази даних у хмарі через сервіс MongoDB Atlas забезпечується постійний доступ до інформації під час розробки та тестування системи.

Результати оцінки техніко-економічних показників проєкту наведено у графічній частині (2026.КВР.122.423.14.00.00 ТБ).

1.2.6 Стадії та етапи розробки

Під час створення інформаційної системи «Comfort» передбачено дотримання чіткої послідовності дій, що дозволяє мінімізувати ризики виникнення помилок на фінальних стадіях реалізації продукту. Кожен крок має значення. Весь процес розробки розділено на логічні етапи, результати яких формують фундамент для подальшого функціонального наповнення веб-застосунку.

Першим етапом розробки визначено аналіз предметної області та детальне планування. На цій стадії проводиться дослідження ринку електронної комерції у сегменті спортивного взуття для виявлення актуальних потреб користувачів та

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

формування технічних вимог. У межах етапу встановлюється перелік необхідних модулів, таких як система фільтрації, кошик та механізми авторизації через JWT. Результатом виступає затверджене технічне завдання.

На другому етапі здійснюється архітектурне проєктування та моделювання бази даних. За допомогою інструментів моделювання формується структура нереляційної бази даних MongoDB, де визначаються схеми для товарів, користувачів та об'єктів замовлень. Окремо розробляється модульна структура серверної частини на базі NestJS, що забезпечує гнучкість і масштабованість майбутнього API. Це критично для стабільності. Також створюються прототипи інтерфейсу для визначення логіки навігації користувача.

Третім етапом є безпосередня програмна реалізація клієнтської та серверної частин. Шляхом написання коду на TypeScript створюються сервіси та контролери для обробки бізнес-логіки на стороні бекенду. Паралельно виконується верстка адаптивних компонентів на Next.js із використанням бібліотеки TailwindCSS для забезпечення коректного відображення на мобільних пристроях. Для управління станом застосунку інтегрується бібліотека Zustand. Результатом стає повнофункціональний прототип системи.

Четвертим етапом визначено тестування, налагодження та оптимізацію коду. Для перевірки стійкості системи проводиться тестування REST API за допомогою інструменту Postman, що дозволяє переконатися у правильності обробки всіх HTTP-запитів. Особлива увага приділяється безпеці даних та швидкості завантаження сторінок з використанням серверного рендерингу (SSR). Виявлені дефекти оперативно усуваються.

П'ятий етап передбачає фінальне розгортання та підготовку до демонстрації результатів. На цій стадії виконується деплой застосунку на хмарні платформи та оформлення пакета програмної документації. Важливо підготуватися до захисту. Процес демонстрації проєкту членам комісії включає наступні покрокові дії:

- 1) Запуск веб-застосунку та демонстрація швидкості завантаження головної сторінки з акційними пропозиціями.

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

- 2) Демонстрація роботи каталогу взуття з використанням фільтрів за брендом, ціною та розміром у реальному часі.
- 3) Показ процесу авторизації клієнта та додавання обраної пари взуття до кошика з подальшим переглядом сторінки оформлення замовлення.
- 4) Перехід до адміністративної панелі для демонстрації процесу додавання нової моделі товару та зміни статусів поточних замовлень.
- 5) Огляд реалізації адаптивного інтерфейсу через режим розробника в браузері для перевірки відображення на смартфонах.

Це підтверджує готовність проєкту.

Завершальним результатом є стабільно працюючий інтернет-магазин «Comfort», який повністю відповідає вимогам технічного завдання та готовий до промислової експлуатації.

1.2.7 Порядок контролю та прийому

Перевірка працездатності інформаційної системи «Comfort» здійснюється шляхом проведення комплексного тестування всіх функціональних модулів на відповідність вимогам технічного завдання. Це робиться ретельно. Під час проведення контрольних випробувань особлива увага приділяється стабільності взаємодії між серверною частиною на NestJS та клієнтським інтерфейсом, побудованим на Next.js.

Для забезпечення об'єктивного контролю працездатності системи встановлюються наступні умови експлуатації:

- наявність стабільного доступу до мережі Інтернет зі швидкістю не менше 5 Мбіт/с;
- використання актуальних версій веб-браузерів Google Chrome, Mozilla Firefox або Safari;
- наявність розгорнутого серверного середовища та підключення до хмарної бази даних MongoDB Atlas;
- підготовка набору тестових даних для перевірки різних сценаріїв

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

покупки та адміністрування.

Всі вузли мають бути активними.

У межах контрольного прикладу передбачено виконання покрокової демонстрації роботи проєкту, яка імітує повний життєвий цикл замовлення спортивного взуття:

- 1) Реєстрація нового користувача через форму на сайті з перевіркою валідації полів та подальша авторизація для отримання доступу до профілю.
- 2) Перегляд головної сторінки з банерами та перехід до каталогу товарів для ознайомлення з асортиментом магазину «Comfort».
- 3) Використання панелі фільтрації для відбору взуття за конкретним брендом, розміром та ціновим діапазоном для перевірки точності роботи API-запитів.
- 4) Вибір моделі взуття, перехід на сторінку товару та додавання пари потрібного розміру до кошика з візуальним підтвердженням дії.
- 5) Редагування вмісту кошика (зміна кількості одиниць або видалення позиції) та перехід до форми оформлення замовлення.
- 6) Введення даних для доставки, вибір способу отримання товару (самовивіз або пошта) та фіксація замовлення в базі даних.
- 7) Авторизація під обліковим записом адміністратора для доступу до закритої панелі керування магазином.
- 8) Додавання нової моделі взуття до каталогу через форму завантаження з перевіркою коректності відображення нового товару на сайті.
- 9) Редагування існуючого товару (зміна ціни або описів) та повне видалення тестової позиції з каталогу для перевірки CRUD-операцій.
- 10) Зміна статусу нещодавно створеного замовлення на «Відправлено» або «Готово до видачі» для перевірки логіки оновлення даних.

Це підтверджує функціональність.

За допомогою інструменту Postman додатково проводиться тестування

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

REST API для контролю коректності передачі JWT-токенів та захисту приватних маршрутів сервера. Безпека є пріоритетом. Шляхом імітації запитів від неавторизованих осіб перевіряється стійкість системи до спроб несанкціонованого доступу до адміністративних ресурсів магазину.

Приймання розробленого програмного продукту здійснюється комісією після успішного проходження всіх етапів контрольних випробувань без виявлення критичних помилок. Виконавець представляє документацію. Процедура прийому передбачає коротку доповідь про архітектурні рішення, демонстрацію роботи веб-застосунку на різних пристроях та відповіді на запитання щодо реалізації технічного завдання.

Система вважається успішно прийнятою, якщо:

- реалізовано всі функціональні можливості, описані у розділах вимог до сайту;
- час відгуку інтерфейсу при виконанні основних операцій відповідає встановленим нормам;
- дані про замовлення та користувачів надійно зберігаються в базі MongoDB без втрат;
- інтерфейс коректно адаптується під мобільні платформи та різні роздільні здатності екрана.

Результат має бути стабільним.

					<i>2026.КВР.122.423.14.00.00 ПЗ</i>	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури сайту і web-сторінок

Під час проєктування інформаційної системи «Comfort» першочерговим завданням виступає визначення її цільового призначення як сучасного інтернет-магазину спортивного взуття. Такий ресурс має бути зручним. Основна увага приділяється створенню інтуїтивно зрозумілого інтерфейсу, який забезпечує швидкий перехід від етапу ознайомлення з асортиментом до фінального оформлення покупки. Шляхом аналізу потреб покупців було розроблено карту сайту, що відображає логічну ієрархію розділів та формат подання інформації на кожній сторінці.

В основу візуальної концепції покладено принципи мінімалізму та високої швидкодії, що критично важливо для утримання уваги відвідувача. Дизайн створюється поетапно. Спочатку розробляється макет головної сторінки, яка виступає візитною картою магазину «Comfort», а після узгодження переходять до деталізації внутрішніх розділів. Результатом цього етапу є набір затверджених макетів, що охоплюють усі сценарії взаємодії користувача з системою.

Для наочного представлення логічних зв'язків між компонентами було розроблено структурну схему інтерфейсу магазину «Comfort» Ця схема детально подана на окремому плакаті 2026.КВР.122.423.14.00.00 СС у графічній частині проєкту.

З точки зору відвідувача, сайт складається з набору функціональних сторінок, кожна з яких вирішує конкретну задачу:

1) Головна сторінка. Слугує для залучення уваги та швидкої навігації до основних категорій товарів. На ній розміщуються:

- інтерактивний банер із актуальними акціями для стимулювання продажів;
- блок «Хіти продажів» з популярними моделями взуття;
- сітка категорій (чоловіче, жіноче, дитяче) для швидкого переходу до

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

потрібного розділу;

- інформаційна карта з адресою фізичного магазину для клієнтів, що обирають самовивозу.

2) Сторінка каталогу товарів. Призначена для ефективного пошуку та відбору взуття серед усього асортименту магазину. Елементи інтерфейсу включають:

- панель фільтрації за брендом, розміром, ціною та сезоном, що дозволяє звужити вибір;

- сітку карток товарів із цінами та індикаторами наявності;

- інструменти сортування (за новизною або вартістю) для зручності порівняння.

3) Картка конкретного товару. Надає вичерпну інформацію для прийняття рішення про покупку. Тут обов'язково присутні:

- галерея зображень високої якості для детального огляду взуття з різних ракурсів;

- блок вибору розміру, де відображаються лише доступні на даний момент варіанти;

- текстовий опис матеріалів, технологій та особливостей моделі;

- розділ відгуків інших покупців для формування довіри до продукту.

4) Кошик користувача. Забезпечує підготовку до оплати та керування обраними позиціями. Інтерфейс дозволяє:

- змінювати кількість одиниць кожної моделі або видаляти зайві товари;

- переглядати автоматично розраховану загальну вартість замовлення;

- переходити до фінального етапу оформлення за один клік.

5) Сторінка оформлення замовлення (Checkout). Орієнтована на швидке введення контактної інформації. Вона містить:

- поля для введення прізвища, імені та номера телефону покупця;

- перелік методів доставки (кур'єрська служба, пошта або самовивіз);

- вибір зручного способу оплати для завершення транзакції.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

б) Особистий кабінет клієнта. Дає можливість авторизованим користувачам керувати своїми даними. Тут відображаються:

- історія всіх попередніх замовлень із їх поточними статусами;
- список бажаного (Wishlist) для збереження взуття на майбутнє;
- налаштування профілю для оновлення контактної інформації.

7) Адміністративна панель. Виступає інструментом для повного керування бізнес-процесами магазину. Вона передбачає:

- таблицю керування каталогом для додавання, редагування та видалення товарів;
- модуль обробки замовлень, де менеджер змінює статуси та контролює оплату;
- систему управління обліковими записами клієнтів для технічної підтримки.

Шляхом розробки адаптивної структури забезпечується коректне відображення кожного елемента на персональних комп'ютерах, ноутбуках та мобільних пристроях. Це гарантує зручність користування. Завдяки логічному групуванню функцій відвідувач не відчуває перевантаження інформацією, що позитивно впливає на загальний досвід взаємодії з брендом «Comfort».

2.2 Створення та верстка сторінок сайту

Після завершення етапу проектування логічної структури було виконано безпосередню програмну реалізацію клієнтської частини інтернет-магазину спортивного взуття «Comfort». Основним завданням цього етапу виступало перетворення статичних макетів у динамічні веб-сторінки з високим рівнем інтерактивності. Для розробки використовувалось сучасне середовище Visual Studio Code, що забезпечує підтримку TypeScript та інтеграцію з системами контролю версій.

В основу технологічного стеку покладено фреймворк Next.js 14, [1] який

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

використовує архітектуру App Router для оптимізації маршрутизації та рендерингу сторінок. Швидкість завантаження є пріоритетом. Для стилізації елементів інтерфейсу застосовано бібліотеку TailwindCSS [8], що дозволяє описувати дизайн безпосередньо у вихідному коді за допомогою утилітарних класів. Це значно прискорює процес верстки та забезпечує легкість підтримки коду.

Верстка всіх компонентів виконувалась із використанням семантичних тегів HTML5, таких як `<header>`, `<main>`, `<nav>`, `<section>` та `<footer>`. Такий підхід покращує індексацію сайту пошуковими системами та робить його більш доступним. Усі сторінки побудовані за принципом мобільної адаптивності, що гарантує коректне відображення інтерфейсу на пристроях із різною роздільною здатністю екрана. Побудова виконується дуже ретельно.

Для надання інтерфейсу сучасного вигляду та забезпечення надійної взаємодії було використано набір бібліотек:

- Lucide-react для інтеграції векторних іконок, які зберігають чіткість при масштабуванні;
- Radix UI для реалізації доступних та функціональних модальних вікон, випадаючих меню та табів;
- Zustand [9] для управління глобальним станом застосунку, включаючи збереження даних кошика та списку обраного;
- Tailwind-merge та clsx для динамічного керування класами стилів залежно від стану компонентів.

Технології працюють злагоджено.

Ключовим елементом навігації виступає компонент Header, розташований у файлі `frontend/src/components/Header.tsx`. Він реалізований як «липкий» блок (клас `sticky top-0`), що залишається видимим під час скролінгу сторінки. У структурі компонента використано тег `<nav>` для відображення основних посилань на категорії каталогу (Чоловікам, Жінкам, Дітям). Для візуального виокремлення активних елементів застосовано класи `hover:text-primary` та плавні переходи `transition-colors`. Вигляд навігаційної панелі зображено на рисунку 2.1.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Рисунок 2.1 – Навігаційна панель та головне меню

З правої сторони навігаційної панелі розміщено функціональні кнопки:

– Поле пошуку: реалізовано через тег `<input type="search">` із абсолютно позиціонованою іконкою Search для швидкого знаходження потрібної моделі взуття;

– Кнопка обраного: містить іконку Star та динамічний лічильник, підключений до сховища `useWishlistStore`. При додаванні товару лічильник миттєво оновлюється без перезавантаження сторінки;

– Кнопка кошика: відображає кількість доданих одиниць товару через компонент `ShoppingCart`. Для управління даними кошика використовується `middleware persist` у `Zustand`, що дозволяє зберігати вибір користувача у `LocalStorage`. Всі дії є миттєвими. Логіка роботи кошика наведена у Додатку Д.

Особлива увага приділена формі авторизації, яка реалізована у компоненті `AuthModal.tsx` на основі бібліотеки `Radix UI Dialog`. Форма входу зображена на рисунку 2.2.

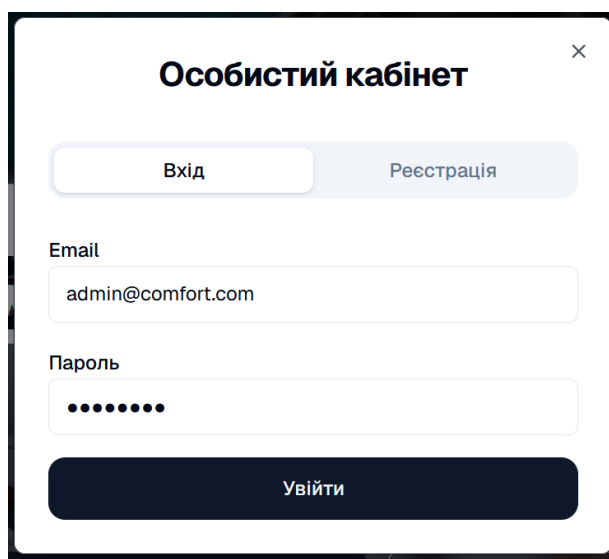


Рисунок 2.2 – Форма входу та авторизації користувача

Для перемикання між режимами входу та реєстрації використано компонент Tabs. Верстка форми включає теги <Label> та <Input>, які мають вбудовану валідацію на стороні клієнта. Обробка даних виконується асинхронно через бібліотеку Axios, що дозволяє користувачу увійти до системи без переривання перегляду каталогу.

Сторінка профілю (frontend/src/app/profile/page.tsx) та сторінка замовлень побудовані як структуровані розділи з використанням сіток (CSS Grid). На сторінці замовлень відображається список попередніх покупок, де кожна позиція представлена у вигляді окремого блоку з детальною інформацією про статус, дату та загальну вартість. Дані завантажуються з сервера та відображаються за допомогою методу .map(), що забезпечує динамічне наповнення інтерфейсу.

Розділ каталогу взуття (frontend/src/app/catalog/page.tsx) зображений на рисунку 2.3 є найскладнішим модулем системи. Його верстка базується на двоколонковій структурі: зліва розташована панель фільтрів, а справа – сітка товарів. Для створення сітки використано класи grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4, що дозволяє автоматично змінювати кількість карток залежно від ширини екрана.

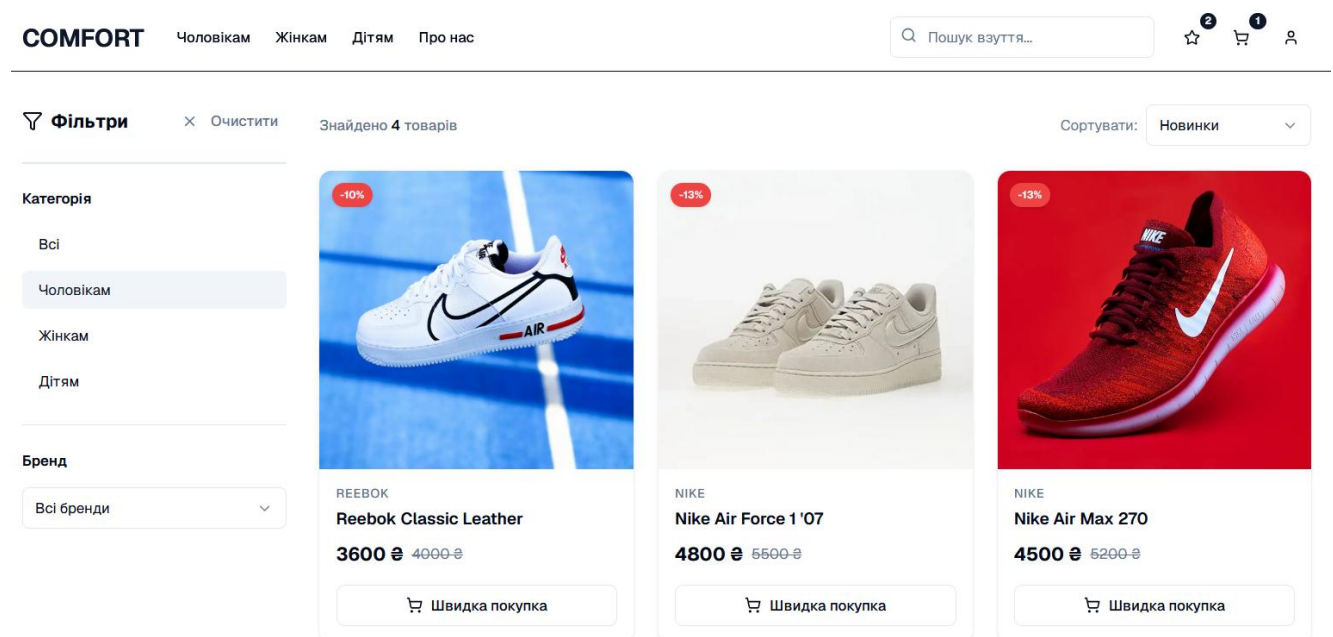


Рисунок 2.3 – Сторінка каталогу з панеллю фільтрації

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

Панель фільтрів реалізована як вертикальний список із використанням елементів <Checkbox> та <Input> для введення діапазону цін. Взаємодія з фільтрами реалізована через оновлення URL-параметрів, що дозволяє користувачам зберігати стан пошуку та ділитися посиланнями на відфільтровані списки взуття. Кожна зміна параметрів викликає повторний рендеринг списку товарів.

Кожна пара взуття у каталозі представлена окремою карткою яка містить:

- тег <Image> від Next.js для оптимізації завантаження та автоматичної конвертації зображень у сучасні формати;
- текстові блоки для назви бренду та моделі з використанням класу line-clamp-1 для запобігання виходу тексту за межі картки;
- кнопку «Швидка покупка» з ефектом active:scale-95 для тактильного відгуку інтерфейсу.

Картка має плавний ефект підйому при наведенні курсора її вигляд зображений на риснку 2.4.

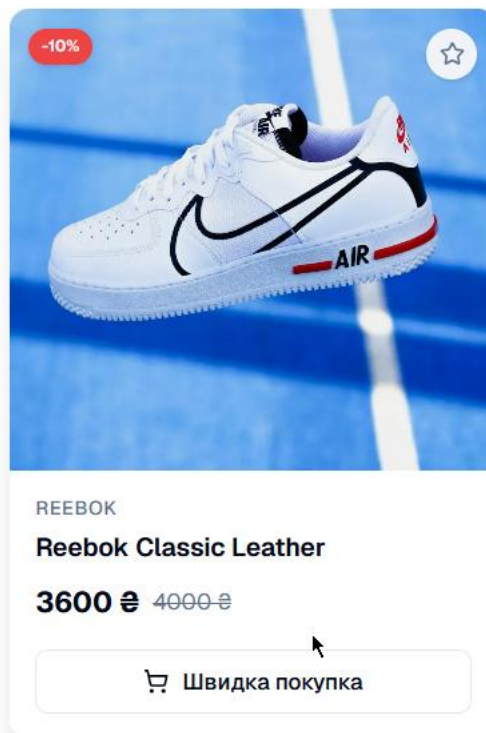


Рисунок 2.4 – Картка взуття та ефекти взаємодії

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

На сторінці конкретного товару (`frontend/src/app/catalog/[id]/page.tsx`) реалізовано розширену логіку вибору розміру. Кожен доступний розмір представлений кнопкою, стан якої (вибрано/недоступно) керується локальним станом `useState`. Для перегляду фотографій взуття зверстано галерею з великим головним зображенням та списком мініатюр знизу. Додавання до кошика супроводжується перевіркою вибору розміру, що мінімізує помилки покупця.

Сторінка оформлення замовлення (`frontend/src/app/checkout/page.tsx`) містить багатокрокову форму, де кожен розділ (контакти, доставка, оплата) виділений у окремий візуальний блок. Для вибору методів доставки використано компонент `RadioGroup`, який забезпечує зручне перемикання між варіантами «Самовивіз» та «Доставка поштою». Сторінка оформлення замовлення зображена на рисунку 2.5. Дані форми збираються в один об'єкт та відправляються на бекенд для фіксації замовлення.

Оформлення замовлення

1 Контактні дані

Ім'я	Прізвище
<input type="text" value="Адмін"/>	<input type="text" value="Магазину"/>
Телефон	Email (опціонально)
<input type="text" value="+380999000000"/>	<input type="text" value="admin@comfort.com"/>

2 Спосіб доставки

<input checked="" type="radio"/> Нова Пошта	<input type="radio"/> Укрпошта	<input type="radio"/> Самовивіз
---	--------------------------------	---------------------------------

Адреса доставки / Відділення

3 Спосіб оплати

<input checked="" type="radio"/> Оплата при отриманні	<input type="radio"/> Карткою онлайн
---	--------------------------------------

Ваше замовлення	
Asics Gel-Kayano 29 (x1)	6200 €
Разом	6200 €
<input type="button" value="Підтвердити замовлення"/>	

Рисунок 2.5 – Сторінка оформлення замовлення

Адміністративна панель (`frontend/src/app/admin/page.tsx`) має власну структуру навігації (Sidebar). Головна сторінка адміністратора (Dashboard) містить картки зі статистикою продажів та графіки активності. Розділ керування взуттям

									Арк.
									34
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.423.14.00.00 ПЗ				

шаблони сторінок, які забезпечують комфортну роботу користувача та адміністратора. Всі компоненти інтегровані з API-інтерфейсами сервера, що дозволяє системі «Comfort» працювати як цілісний програмний продукт. Гнучкість реалізації дозволяє легко масштабувати систему у майбутньому.

2.3 Розробка структури бази даних сайту

Для забезпечення стабільної роботи інтернет-магазину «Comfort» та надійного збереження інформації про асортимент взуття, у межах проєкту виконується ретельне проєктування архітектури бази даних. Вона виступає ядром системи. Шляхом глибокого аналізу предметної області вдається відобразити складну структуру торговельних об'єктів у логічну схему внутрішньої моделі даних, що є критично важливим для швидкодії. Побудова виконується поступово. Процес створення бази даних являє собою послідовність уточнень, які дозволяють визначити оптимальний склад об'єктів, їхні властивості та типи зв'язків для задоволення потреб майбутніх покупців та адміністраторів. Це гарантує надійність.

В основу вибору технологічного рішення покладено нереляційну систему керування базами даних MongoDB [3], яка забезпечує високу гнучкість при роботі з динамічними атрибутами спортивних товарів. Гнучкість схем є перевагою. Завдяки використанню документо-орієнтованої моделі у форматі BSON з'являється можливість зберігати складні вкладені об'єкти, такі як масиви доступних розмірів взуття, без необхідності створення додаткових зв'язних таблиць. Це суттєво прискорює читання. Для реалізації взаємодії між сервером на NestJS та базою даних застосовано бібліотеку об'єктного моделювання Mongoose [7], яка дозволяє суворо контролювати структуру документів на рівні програмного коду.

Під час проєктування бази даних було реалізовано наступні ключові етапи розробки:

- за допомогою аналізу бізнес-процесів визначено основні колекції, які повинна містити база даних для повноцінного функціонування магазину «Comfort»;

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

– шляхом деталізації властивостей кожного об'єкта встановлено необхідний перелік полів, типи даних та обов'язковість їх заповнення для забезпечення цілісності інформації;

– через налаштування унікальних індексів та індивідуальних значень полів організовано швидкий пошук товарів та надійну ідентифікацію користувачів у системі;

– за допомогою встановлення логічних посилань між документами визначено типи зв'язків, що дозволяють відстежувати історію замовлень та формувати списки обраного;

– шляхом розробки спеціалізованих агрегаційних запитів та інструментів адміністрування підготовлено базу до наповнення реальним контентом та виведення аналітичних звітів.

Результат є повністю валідним.

Колекція «Users» призначена для зберігання персональних даних клієнтів та керування правами доступу до адміністративної частини ресурсу. До її складу входять наступні поля:

– email (String): унікальна адреса електронної пошти, що виступає логіном для входу та проходить обов'язкову валідацію на рівні схеми;

– passwordHash (String): зашифрований за допомогою алгоритму bcrypt пароль, доступ до якого обмежено для забезпечення безпеки облікових записів;

– firstName та lastName (String): прізвище та ім'я користувача для персоналізації інтерфейсу та оформлення супровідних документів на доставку;

– phone (String): контактний номер телефону, необхідний для підтвердження замовлень та зв'язку менеджера з покупцем;

– role (Enum): роль користувача у системі, яка може набувати значень «client» або «admin» для розмежування функціональних можливостей;

– wishlist (Array of ObjectIds): масив посилань на документи з колекції товарів, що дозволяє користувачу зберігати моделі взуття для майбутньої покупки.

Дані захищені, структура документа User представлена у Додатку Б.

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

Колекція «Products» містить вичерпну інформацію про кожну пару взуття, що представлена в асортименті магазину «Comfort». Структура документа включає:

- title та brand (String): назва моделі та торговельна марка виробника із увімкненим видаленням зайвих пробілів (trim) для чистоти даних;
- description (String): розгорнутий опис характеристик, матеріалів та особливостей взуття для інформування відвідувачів сайту;
- price та oldPrice (Number): поточна ціна реалізації та попередня вартість для відображення знижок та акційних пропозицій;
- category (Enum): категорія взуття (чоловіче, жіноче, дитяче), що використовується як основний параметр для глобальної фільтрації каталогу;
- sizes (Array of Objects): вкладений масив об'єктів, де кожна позиція містить числове значення розміру та фактичну кількість одиниць у наявності на складі;
- images (Array of Strings): список URL-адрес зображень високої якості, що завантажуються з хмарного сховища Cloudinary для швидкого відображення.

Каталог працює надзвичайно швидко.

Колекція «Orders» забезпечує фіксацію всіх транзакцій та відстеження етапів виконання покупок клієнтами. Опис структури замовлення містить:

- userId (ObjectId): посилання на документ користувача, що є не обов'язковим для гостьових замовлень, але критичним для авторизованих покупців;
- items (Array of Objects): детальний перелік замовленого взуття із зазначенням конкретного ідентифікатора продукту, обраного розміру, кількості та ціни на момент покупки;
- totalAmount (Number): фінальна сума до сплати, що розраховується автоматично на сервері для запобігання підміні даних на клієнтській стороні;
- deliveryMethod та deliveryAddress (String): обраний спосіб отримання товару (самовивіз, пошта) та повна адреса пункту призначення або відділення;

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

- status (Enum): поточний стан замовлення (очікує, в дорозі, готово до видачі, завершено), що оновлюється адміністратором через панель керування;
- customerInfo (Object): вкладений об'єкт із контактними даними покупця для оперативної обробки замовлення менеджером магазину.

Цикл замовлення є повним.

У процесі проектування були чітко визначені типи взаємозв'язків між колекціями для забезпечення цілісності бази даних. Зокрема, встановлено зв'язок «один-до-багатьох» між користувачами та замовленнями, що дозволяє накопичувати історію покупок в одному профілі. Посилання у масиві «wishlist» формують зв'язок «багато-до-багатьох» через програмну обробку на рівні Mongoose, що забезпечує гнучкість вибору. Завдяки використанню вкладених документів для елементів замовлення вдалося уникнути зайвих операцій з'єднання (JOIN) під час виведення детальної інформації про чеки.

Добре спроектована структура бази даних дозволяє системі «Comfort» ефективно обробляти запити навіть при великій кількості одночасних підключень. Це фундамент стабільності проекту. Для наочного представлення логічної структури та взаємозв'язків між основними сутностями було розроблено ER-діаграму бази даних інтернет-магазину.

Результат, отриманий після закінчення робіт етапу, — ER-діаграма та детальний опис структури бази даних сайту. ER-діаграма відображає взаємозв'язки між основними сутностями системи (користувачі, товари, замовлення) і демонструє логічну структуру збереження даних у системі. ER-діаграма представлена на окремому плакаті 2026.КВР.122.423.14.03.00 БД у графічній частині кваліфікаційної роботи.

Шляхом впровадження індексування за полями категорій та брендів вдалося досягти миттєвого відгуку при використанні складної фільтрації у каталозі взуття. Кожна операція запису супроводжується автоматичною валідацією типів, що мінімізує ймовірність появи помилок у базі даних MongoDB Atlas. Така архітектура забезпечує масштабованість.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

2.4 Програмування серверної частини сайту

Реалізація бекенд-складової інтернет-магазину «Comfort» базується на принципах модульної архітектури [4], що забезпечує чіткий розподіл бізнес-логіки, доступу до даних та маршрутизації запитів. Такий підхід робить систему надійною. Завдяки використанню прогресивного фреймворку NestJS [2] вдалося створити масштабований серверний застосунок, де кожна функціональна область (користувачі, товари, замовлення) виділена в окремий програмний модуль. Це значно спрощує тестування. Взаємодія між клієнтом та сервером побудована за стандартом REST API з використанням формату JSON для обміну даними, що гарантує високу сумісність із фронтенд-частиною на Next.js.

В основу архітектури покладено клієнт-серверний механізм, де сервер виступає проміжним шаром між базою даних MongoDB Atlas та інтерфейсом користувача. Загальна схема архітектурної взаємодії компонентів представлена на рисунку 2.7.

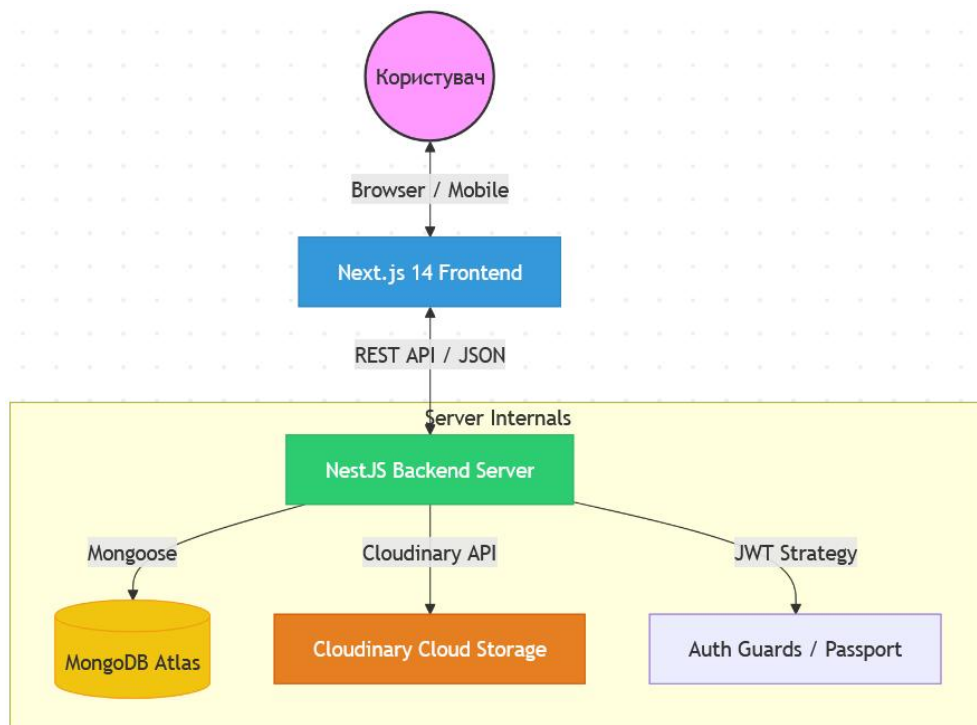


Рисунок 2.7 – Схема архітектури клієнт-серверної взаємодії «Comfort»

Як показано на схемі вище, клієнтська частина на базі Next.js 14 взаємодіє з NestJS сервером через HTTP-протокол. Серверна логіка включає інтеграцію з Mongoose для роботи з базою даних та Cloudinary [11] API для збереження медіафайлів. Окремий рівень безпеки забезпечується через Auth Guards та стратегії Passport. Шляхом використання вбудованого контейнера впровадження залежностей (Dependency Injection) у NestJS досягається гнучке керування сервісами та репозиторіями. Це робить код чистим. Кожен запит від клієнта проходить через систему контролерів, які делегують виконання операцій спеціалізованим сервісам. Така структура запобігає дублюванню логіки та дозволяє легко вносити зміни в алгоритми розрахунку вартості замовлень або правила фільтрації каталогу.

Для реалізації проєкту було задіяно наступний технологічний стек та набір пакетів:

- `@nestjs/core` та `@nestjs/common` – ядро системи для створення модулів, контролерів та провайдерів;
- `@nestjs/mongoose` та `mongoose` – інструменти для об'єктного моделювання та стабільної взаємодії з базою даних MongoDB;
- `passport` [12] та `passport-jwt` – бібліотеки для реалізації стратегій авторизації на основі токенів доступу;
- `bcrypt` – алгоритм для надійного одностороннього хешування паролів перед їх збереженням у базу даних;
- `class-validator` та `class-transformer` – пакети для автоматичної валідації вхідних даних (DTO) за допомогою декораторів;
- `@nestjs/config` – модуль для безпечного керування змінними оточення через файли .env.

Всі пакети працюють злагоджено.

Процес ініціалізації сервера починається з головного файлу `main.ts`, де налаштовується глобальний конфіг та підключаються проміжні обробники. Зокрема, було активовано `ValidationPipe`, який автоматично перевіряє коректність

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

усіх вхідних об'єктів замовлень та профілів. Налаштування сервера NestJS наведено у Додатку А. Для забезпечення безпеки транскордонних запитів налаштовано політику CORS, що дозволяє приймати запити виключно з довіреного домену фронтенду.

Структура серверної частини організована у вигляді ієрархії каталогів, де кожен модуль містить власні DTO, контролери та сервіси. На рисунку 2.8 зображено дерево каталогів бекенду у середовищі розробки.

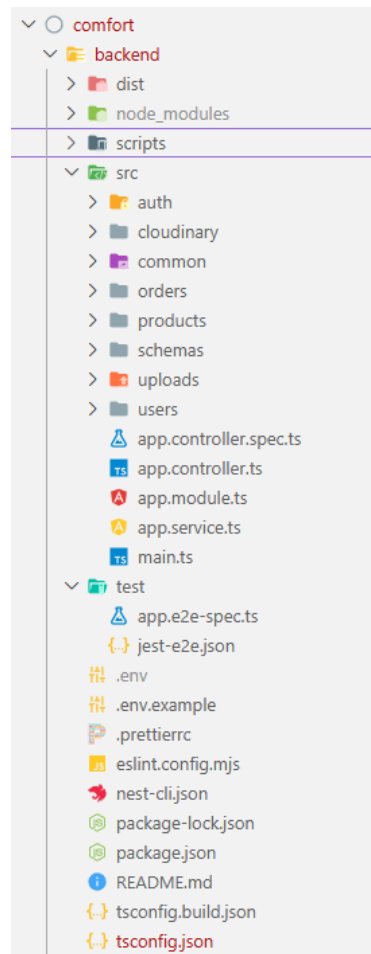


Рисунок 2.8 – Структура модулів серверної частини проєкту

Як видно з діаграми структури модулів, центральним елементом є `AppModule`, який імпортує всі функціональні блоки системи. Модулі `Products` та `Orders` мають власні контролери та сервіси для обробки специфічної бізнес-логіки. Взаємозв'язки між ними мінімізовані для забезпечення автономності

розробки. Модуль авторизації (`AuthModule`) реалізує повний цикл керування сесіями користувачів за допомогою стратегій Passport. Вхідні запити обробляються контролером `auth.controller.ts`. Система підтримує наступні ендпоінти:

- `POST /auth/register` – реєстрація нового клієнта з обов'язковим хешуванням пароля;
- `POST /auth/login` – перевірка облікових даних та встановлення токенів у захищені куки;
- `POST /auth/logout` – анулювання поточного токена оновлення та очищення клієнтських кук;
- `GET /auth/refresh` – отримання нової пари токенів без необхідності повторного введення пароля.

Безпека даних є гарантованою, механізм авторизації та генерації токенів описано у Додатку В.

Шлях проходження запиту під час входу в систему детально відображено на схемі алгоритму процесу авторизації зображеному на рисунку 2.9.

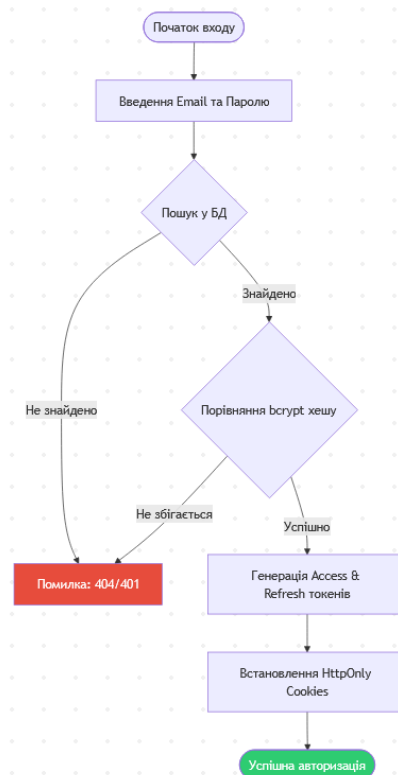


Рисунок 2.9 – Алгоритм процесу авторизації користувача

Згідно з наведеною схемою, після введення даних виконується пошук користувача в базі даних та порівняння bcrypt-хешу. У разі успіху генеруються токени, які встановлюються як HttpOnly Cookies для захисту від XSS-атак. Для надійного захисту приватних ресурсів використовується механізм `AuthGuard('jwt')`, який перевіряє підпис токена у кожному запиті. Додатково впроваджено `RolesGuard`, що дозволяє розмежовувати доступ за рівнем привілеїв.

Обробка інформації про товари здійснюється у `ProductsModule`. Контролер забезпечує публічний доступ до каталогу та закритий доступ до методів редагування. Повний список ендпоінтів модуля:

- `GET /products` – повертає масив моделей із врахуванням фільтрів за брендом та ціною;
- `GET /products/brands` – надає список усіх доступних торговельних марок для панелі фільтрації;
- `GET /products/:id` – детальна інформація про одну пару взуття для динамічної сторінки товару;
- `POST /products` – додавання нової позиції в базу даних (захищено `Roles('admin')`);
- `PATCH /products/:id` – часткове оновлення даних про залишки розмірів або описи;
- `DELETE /products/:id` – повне видалення товару з каталогу магазину «Comfort».

Дані обробляються миттєво.

Логіка управління замовленнями реалізована в `OrdersModule`, який взаємодіє як із клієнтською, так і з адміністративною частинами. Послідовність дій при оформленні покупки представлена у вигляді діаграми послідовності на рисунку 2.10.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

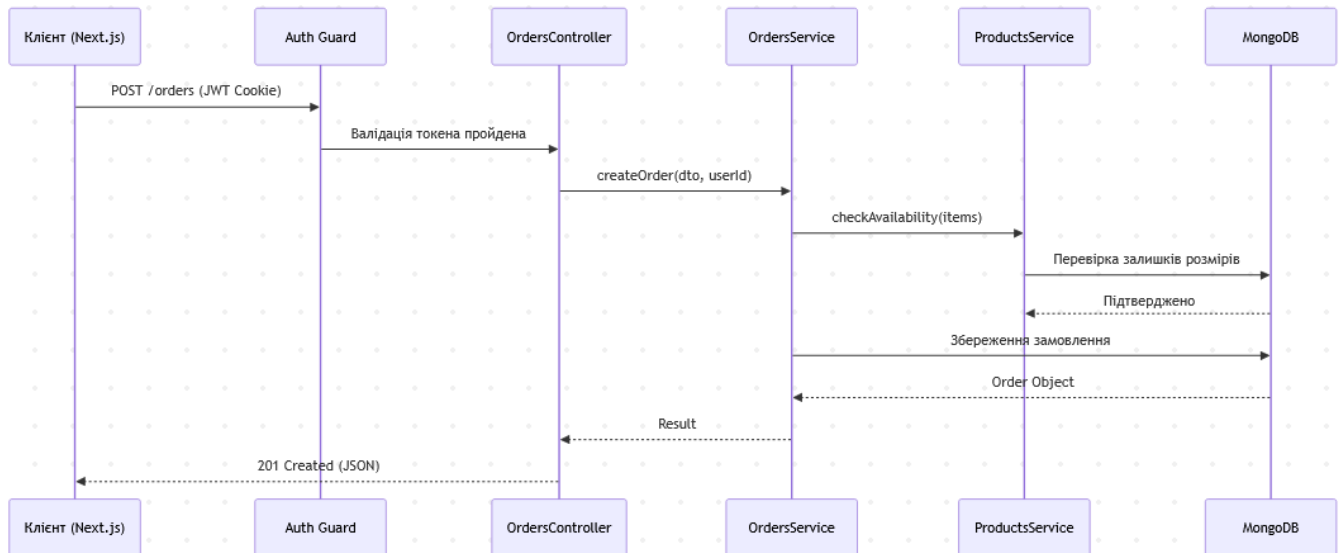


Рисунок 2.10 – Діаграма послідовності процесу оформлення замовлення

Як показано на діаграмі вище, сервіс замовлень координує роботу з модулем товарів для перевірки наявності розмірів перед фіксацією транзакції у базі MongoDB. Це запобігає помилкам. Доступні маршрути модуля замовлень включають:

- `POST /orders` – створення нового замовлення з автоматичним розрахунком вартості;
- `GET /orders/my` – виведення історії покупок поточного авторизованого користувача;
- `GET /orders` – отримання повного списку всіх транзакцій (тільки для адміністратора);
- `GET /orders/stats` – аналітична інформація про продажі для дашборду управління;
- `PATCH /orders/:id/status` – зміна етапів обробки замовлення менеджером магазину.

Весь цикл є прозорим, а бізнес-логіка створення замовлень та списання залишків наведена у Додатку Г.

Завантаження медіафайлів виконується через `UploadsModule` за допомогою інтерцептора `FileInterceptor`. Це спрощує роботу. Процес передбачає

передачу файлу до хмарного сервісу Cloudinary, який повертає оптимізовану URL-адресу. Ендпоінт `POST /uploads` доступний лише для персоналу магазину. Таке рішення дозволяє розвантажити основний сервер та забезпечити швидку видачу зображень користувачам незалежно від їхнього розташування.

У результаті програмування серверної частини було створено відмовостійке API, що об'єднує всі модулі системи в єдиний функціональний організм. Всі компоненти протестовані на предмет коректної взаємодії. Гнучкість архітектури NestJS забезпечує можливість легкого масштабування проєкту «Comfort» у майбутньому без необхідності повної перебудови вихідного коду. Це робить розробку перспективною.

2.5 Тестування web-сайту

Під час завершення активної фази програмування інтернет-магазину «Comfort» обов'язковим етапом стає проведення всебічного тестування для підтвердження стабільності всіх модулів системи. Це гарантує високу якість продукту. Перевірка здійснюється з метою верифікації функціональних можливостей, оцінки швидкодії інтерфейсу та контролю безпеки передачі даних між клієнтом і сервером. Шляхом імітації реальних сценаріїв поведінки покупців вдається виявити та вчасно усунути логічні помилки в коді, що забезпечує надійність торговельних операцій у майбутньому.

Для отримання об'єктивних результатів було реалізовано декілька стратегій перевірки працездатності застосунку:

- функціональне тестування для контролю правильності роботи кошика, пошуку та системи замовлень;
- інтеграційне тестування для перевірки коректності обміну JSON-даними між Next.js фронтендом та NestJS бекендом;
- тестування користувацького інтерфейсу (UI/UX) для оцінки зручності навігації та адаптивності елементів на різних екранах;
- тестування безпеки для підтвердження стійкості механізмів JWT-

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

авторизації та захисту від несанкціонованого доступу до бази MongoDB.

Всі компоненти працюють злагоджено.

На початковому етапі виконано перевірку модуля автентифікації користувачів через компонент AuthModal. Під час введення електронної адреси та пароля система автоматично звертається до сервера для звірки хешу в базі даних. У разі успішного збігу користувач отримує доступ до особистого кабінету та історії своїх покупок. Інтерфейс форми авторизації, що використовувався під час випробувань, наведено на рисунку 2.11.

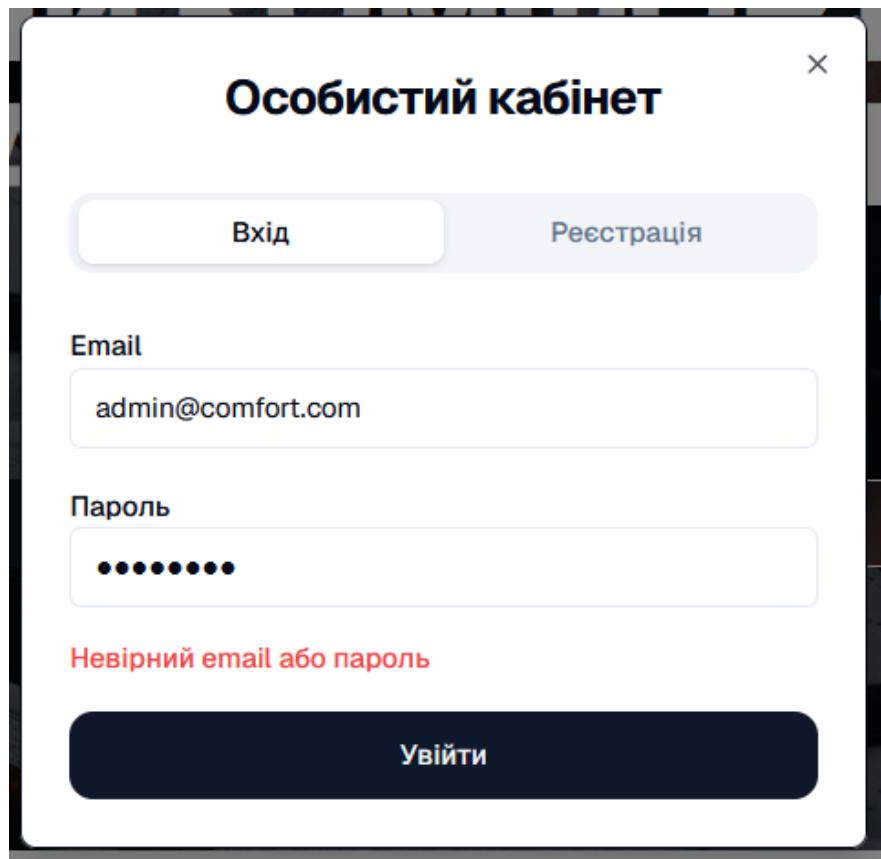


Рисунок 2.11 – Форма входу та перевірка валідації полів

Після входу до системи здійснюється тестування сторінки каталогу з використанням динамічних фільтрів за ціною та брендом. Кожен вибір користувача ініціює новий API-запит, який повинен повертати актуальний масив товарів без затримок. Швидкість рендерингу є високою. Результат коректного відображення відфільтрованого списку взуття представлено на рисунку 2.12.

🔍 Фільтри

✕ Очистити

Знайдено 3 товарів

Категорія

Всі

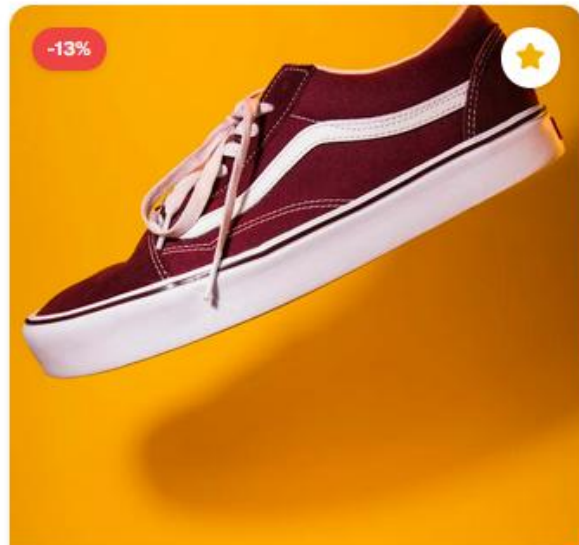
Чоловікам

Жінкам

Дітям

Бренд

Всі бренди



VANS

Vans Old Skool

2800 € ~~3200 €~~

🛒 Швидка покупка

-16%

PUMA

Puma

3200

Рисунок 2.12 – Робота панелі фільтрації та сітки товарів

Особлива увага приділяється процесу додавання товару до кошика та подальшому оформленню замовлення на сторінці Checkout. Перевіряється можливість зміни кількості пар взуття безпосередньо у вікні кошика та правильність розрахунку фінальної вартості.

Для завершення оформлення покупки користувач заповнює контактну форму, вказуючи ім'я, номер телефону та адресу для відправлення. Під час тестування перевіряється робота клієнтської валідації полів, інтеграція зі стейт-менеджером Zustand для збереження стану кошика, а також логіка перерахунку загальної вартості доставки. Після коректного заповнення форма відправляється на NestJS сервер для збереження замовлення у базі даних MongoDB. Процес оформлення замовлення та інтерфейс вибору способів доставки зображено на рисунку 2.13.

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

Оформлення замовлення

1 Контактні дані

Ім'я: Микола
Прізвище: Мірошниченко
Телефон: 0991234567
Email (опціонально):

2 Спосіб доставки

Нова Пошта | Укрпошта | Самовивіз

Адреса магазину: вул. Спортивна, 123, м. Київ

3 Спосіб оплати

Оплата при отриманні | Карткою онлайн

Рисунок 2.13 – Оформлення замовлення та вибір способів доставки

Для підтвердження надійності адміністративної частини було проведено серію тестів із управління контентом через Dashboard. Шляхом створення тестової картки товару та завантаження фотографій через сервіс Cloudinary підтверджено працездатність модуля Uploads. Записи в БД створюються коректно. Вигляд інтерфейсу керування каталогом для менеджера магазину наведено на рисунку 2.14.

Редагувати товар
Заповніть форму нижче, щоб оновити товар.

Назва товару: Nike Air Force 1 '07 | Бренд: Nike
Категорія: Чоловіче | Ціна (€): 4800 | Стара ціна (€): 5500
Опис: Легендарна модель Nike Air Force 1 зберігає класичний стиль. Зносостійкий верх із натуральної шкіри та прихований балон Air забезпечують ідеальний комфорт.
Зображення: [Завантажити фото]
Розміри та кількість: [Додати розмір]

40	2
41	5
42	7
43	4

Рисунок 2.14 – Адміністративна панель для редагування товарів

За допомогою інструменту Postman було виконано стрес-тестування API-ендпоінтів сервера. Зокрема, імітувалися запити з некоректними типами даних для перевірки роботи ValidationPipe у NestJS. Сервер повинен повертати чітку помилку 400 Bad Request при порушенні структури вхідного JSON. Це запобігає збоєм системи. Окрім того, перевірено захист маршрутів: при спробі доступу до /admin без відповідної ролі система видає код 403 Forbidden. Безпека є пріоритетом.

Результати перевірки основних компонентів системи «Comfort» зведені у таблицю 2.1, що виступає підсумковим протоколом тестування.

Таблиця 2.1 – Контрольний список результатів тестування (Checklist)

№	Компонент системи	Критерій успішності	Статус
1	Авторизація клієнта	Коректне встановлення JWT у HttpOnly Cookies	Виконано
2	Пошук та фільтри	Час відповіді сервера на запит не більше 500 мс	Виконано
3	Робота кошика	Збереження стану в LocalStorage після перезавантаження	Виконано
4	Оформлення покупки	Успішна фіксація замовлення у базі даних	Виконано
5	Адмін-панель	Повне виконання CRUD-операцій з товарами	Виконано
6	Завантаження фото	Автоматична оптимізація через Cloudinary	Виконано
7	Адаптивність	Відсутність горизонтального скролу на iPhone/Android	Виконано

У результаті проведених випробувань встановлено, що веб-застосунок магазину «Comfort» функціонує стабільно та повністю відповідає вимогам, закладеним у технічному завданні. Взаємодія між клієнтом на Next.js та серверною логікою забезпечує цілісність даних та комфортну роботу користувача. Проєкт готовий до розгортання.

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розміщення сайту в Інтернеті

Для забезпечення публічного доступу до розробленого інтернет-магазину «Comfort» виконується процедура розгортання програмних компонентів на хмарних хостинг-платформах, що гарантують стабільну роботу та безпеку даних. Це робиться покроково. Шляхом використання спеціалізованих сервісів Render для серверної частини та Vercel для клієнтського інтерфейсу вдається побудувати відмовостійку інфраструктуру з автоматичним оновленням коду. Процес деплою вимагає чіткого налаштування змінних оточення та команд збірки для коректної взаємодії модулів через мережу. Результат має бути стабільним.

Під час підготовки до розміщення серверної логіки на платформі Render необхідно виконати наступну послідовність дій:

- 1) Реєстрація та авторизація на сайті `render.com` за допомогою облікового запису GitHub для синхронізації з репозиторієм проєкту «Comfort».
- 2) Створення нового веб-сервісу через панель керування шляхом вибору пункту «New Web Service» та вказання посилання на backend-директорію.
- 3) Налаштування параметрів збірки, у полі «Build Command» вказується ``npm install && npm run build``, а у полі «Start Command» – команда ``node dist/main``. Сторінка з налаштуванням команд запуску, зображена на рисунку 3.1.

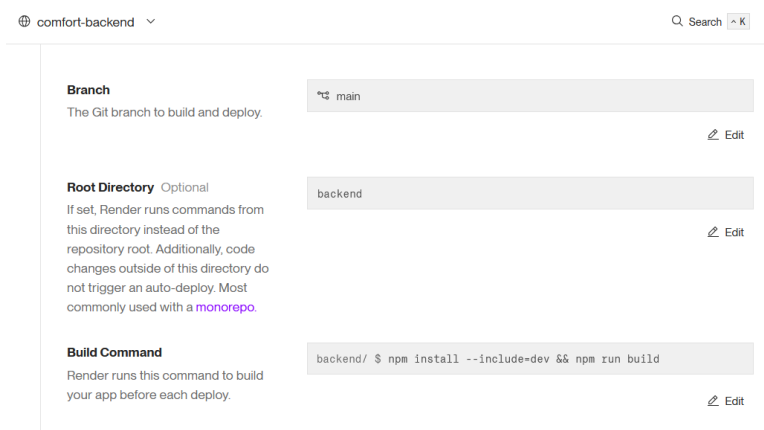


Рисунок 3.1 – Налаштування параметрів збірки Web Service на Render

Важливим етапом виступає конфігурація змінних оточення у розділі «Environment», де прописуються ключі доступу до бази даних та хмарних сховищ. Безпека понад усе. Для роботи сервера необхідно додати параметри `MONGODB_URI`, `JWT_SECRET`, а також `CLOUDINARY_URL` для обробки фотографій взуття. Після збереження налаштувань Render автоматично запустить процес компіляції коду на TypeScript та надасть унікальну URL-адресу для API-ендпоінтів магазину «Comfort». Це дозволяє перевірити доступність бекенду.

Для розгортання клієнтської частини на базі Next.js використовується сервіс Vercel, який забезпечує оптимальну швидкість завантаження через глобальну мережу доставки контенту. Побудова виконується швидко. Послідовність кроків для деплою фронтенду включає:

1) Створення нового проєкту в особистому кабінеті Vercel шляхом імпорту репозиторію з кодом клієнтської частини.

2) Автоматичне визначення фреймворку Next.js системою, що дозволяє уникнути ручного введення складних конфігурацій для статичного рендерингу.

Сторінка налаштувань Render зображена на рисунку 3.2.

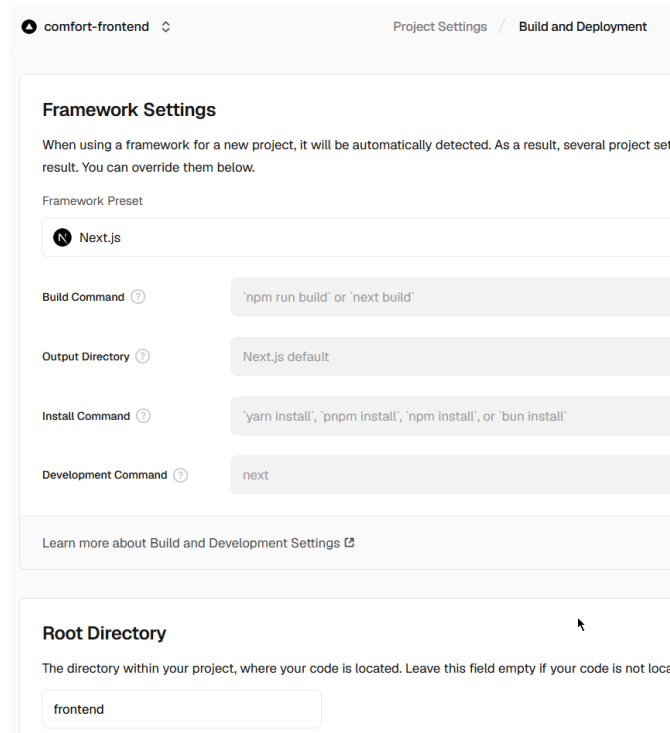


Рисунок 3.2 – Інтерфейс імпорту проєкту та вибору Root Directory у Vercel

Під час фінального налаштування у блоці «Environment Variables» обов'язково додається змінна `NEXT_PUBLIC_API_URL` зі значенням адреси, отриманої раніше від сервісу Render. Саме через цей параметр здійснюється зв'язок фронтенду з базою даних. Шляхом натискання кнопки «Deploy» активується процес збірки, під час якого виконується оптимізація зображень та мінімізація JavaScript-файлів для прискорення роботи сайту. Кінцевий результат відображається миттєво і зображений на рисунку 3.3.

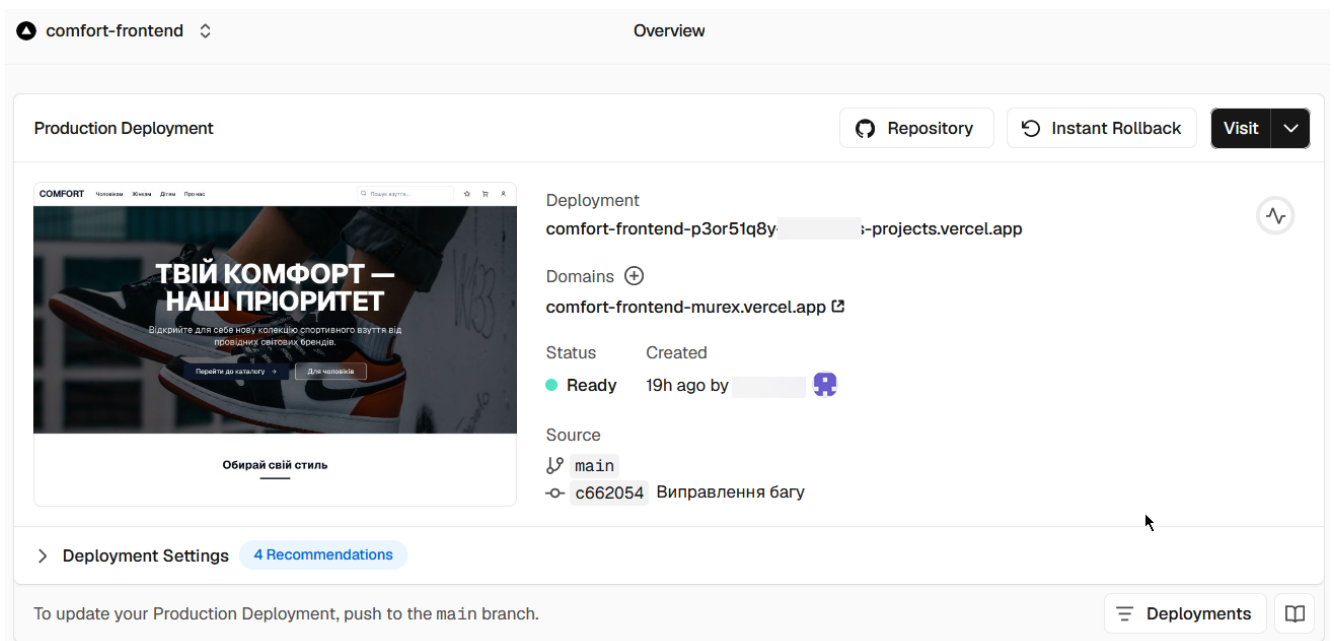


Рисунок 3.3 – Вікно успішного завершення деплою та перегляду Deployment Summary

Після завершення всіх етапів розгортання проводиться фінальна перевірка інтеграції, яка включає тестування авторизації та оформлення замовлення на реальних серверах. Всі посилання мають працювати. Завдяки використанню хмарних технологій інтернет-магазину «Comfort» стає доступним для користувачів з будь-якої точки світу в режимі 24/7, що повністю відповідає цілям комерційного проекту. Стабільність роботи гарантується хостинг-провайдерами.

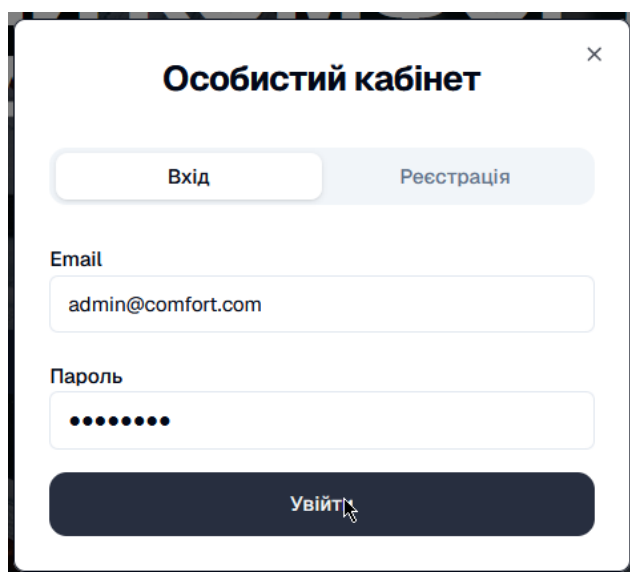
3.2 Інструкція з обслуговування та наповнення сайту

Для підтримки актуальності асортименту та стабільної роботи інтернет-магазину «Comfort» передбачено спеціалізовані інструменти керування контентом через закриту адміністративну панель. Це спрощує роботу персоналу. Шляхом використання інтуїтивно зрозумілого інтерфейсу менеджер може оперативно додавати нові моделі взуття, змінювати ціни та контролювати наявність розмірів на складі. Процес наповнення каталогу є циклічним. Кожна операція супроводжується автоматичною валідацією даних на сервері NestJS для запобігання появі некоректної інформації на сайті. Результат відображається миттєво.

Під час підготовки до додавання нової позиції у каталог необхідно виконати наступну покрокову процедуру:

1) Авторизація у системі під обліковим записом адміністратора для отримання доступу до розділу керування.

За допомогою форми входу зображеній на рисунку 3.4 вводиться логін та пароль адміністратора. Після успішної автентифікації у навігаційному меню з'являється посилання на панель керування.



The image shows a web form titled "Особистий кабінет" (Personal Cabinet) with a close button (X) in the top right corner. At the top, there are two buttons: "Вхід" (Login) and "Реєстрація" (Registration). Below these are two input fields: "Email" with the value "admin@comfort.com" and "Пароль" (Password) with a masked password of seven dots. At the bottom of the form is a large dark blue button labeled "Увійти" (Login).

Рисунок 3.4 – Вікно авторизації адміністратора магазину

2) Перехід до розділу «Каталог товарів» у бічному меню адміністративної панелі для перегляду поточної бази взуття.

Через інтерактивну таблицю зображену на рисунку 3.5 менеджер може оцінити існуючий асортимент та переконатися у відсутності дублікатів перед створенням нової картки. Це важливо для порядку.

Панель адміністратора

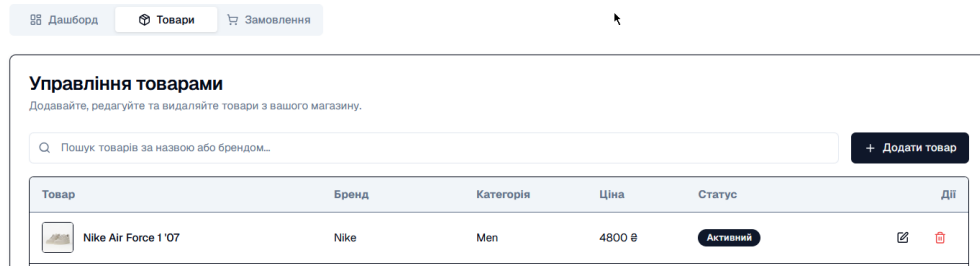


Рисунок 3.5 – Інтерфейс таблиці товарів у Dashboard

3) Активація модального вікна «Додати товар» та заповнення основних текстових полів опису моделі зображено на рисунку 3.6.

У відповідні форми вноситься назва моделі, бренд, детальні характеристики матеріалів та ціна реалізації. Поля мають вбудовану перевірку.

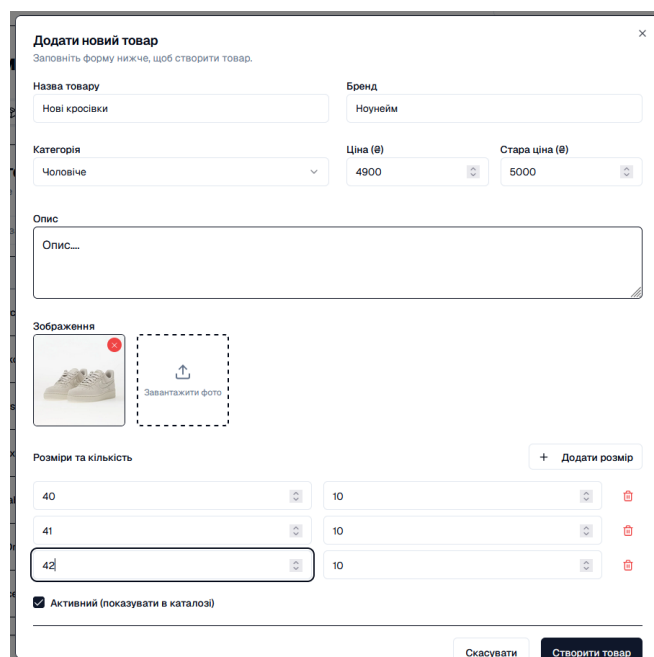


Рисунок 3.6 – Модальне вікно створення нової позиції

4) Налаштування масиву доступних розмірів та вказання фактичної кількості одиниць для кожного значення.

За допомогою динамічних інпутів зображених на рисунку 3.7 додаються актуальні розмірні сітки для чоловічого, жіночого або дитячого взуття відповідно до накладних постачальника. Це забезпечує точність залишків.



Рисунок 3.7 – Блок вибору розмірів у формі додавання

5) Завантаження графічних матеріалів високої якості через інтегрований модуль взаємодії з хмарою Cloudinary.

Шляхом вибору файлів на локальному пристрої ініціюється процес передачі зображень до хмарного сховища для подальшої оптимізації. Фото обробляються дуже швидко, кнопка завантаження зображення зображена на рисунку 3.8.

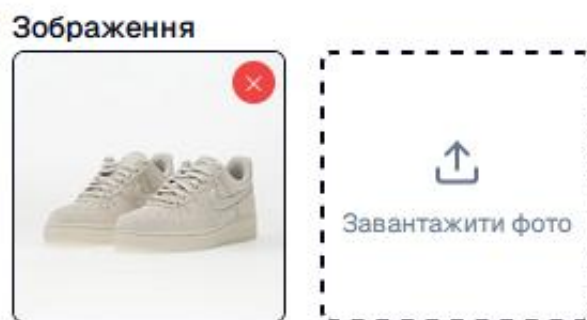


Рисунок 3.8 – Модуль завантаження та перегляду фотографій

б) Фінальне збереження даних та перевірка коректності відображення нової моделі на клієнтській стороні сайту.

Після натискання кнопки «Зберегти» система фіксує новий документ у базі MongoDB Atlas та очищує кеш фронтенду для миттєвого оновлення. Це завершує процес додавання, нова пара взуття з'являється в каталозі, сторінка зображена на рисунку 3.9.

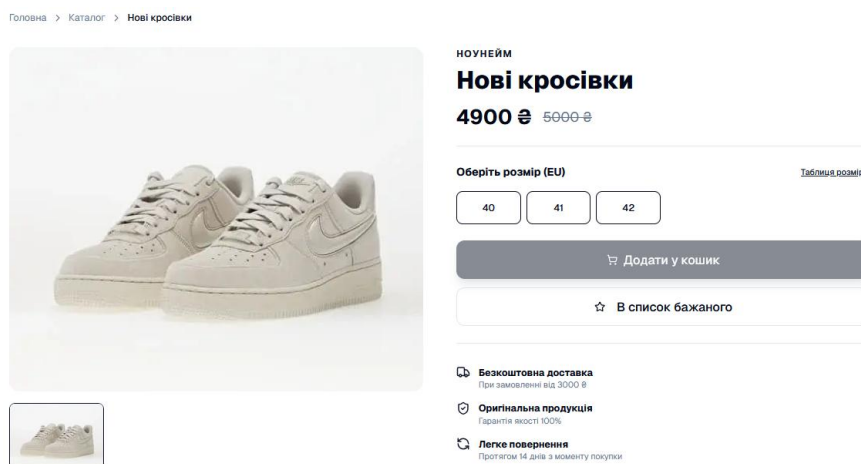


Рисунок 3.9 – Вигляд нової картки товару в основному каталозі

З метою швидкого наповнення бази даних великим обсягом тестової інформації у проєкті передбачено використання спеціалізованого сід-скрипта (seed script). Це економить багато часу. Завдяки використанню команди `npm run seed` у терміналі розробника виконується автоматична генерація масиву об'єктів товарів із випадковими, але реалістичними характеристиками. Такий підхід дозволяє перевірити роботу пагінації та фільтрів на великих обсягах даних ще до моменту завантаження реального асортименту магазину «Comfort».

Шляхом виконання сід-скрипта база даних MongoDB Atlas заповнюється структурованими документами, що містять посилання на демонстраційні зображення та повні розмірні сітки. Робота виконується програмно. Це забезпечує можливість швидкого розгортання копії системи для тестування нових функцій або проведення демонстрацій без необхідності ручного введення інформації. Всі операції є безпечними. Подальше обслуговування системи полягає у регулярному моніторингу логів сервера та своєчасному оновленні складських залишків через описаний вище інтерфейс адміністратора.

										Арк.
										57
Зм.	Арк.	№ докум.	Підпис	Дата	2026.КВР.122.423.14.00.00 ПЗ					

3.3 Інструкція з популяризації та підтримки сайту

У межах стратегії залучення клієнтів для магазину «Comfort» першочерговим етапом виступає технічна оптимізація сторінок під вимоги сучасних пошукових систем. Це виступає основою успіху. Завдяки впровадженню серверного рендерингу (SSR) у Next.js 14 забезпечується повна індексація динамічного контенту каталогу взуття роботами Google без додаткових затримок. Швидкість завантаження стає критичною. Шляхом використання Metadata API для кожної картки товару автоматично генеруються унікальні заголовки та описи, що дозволяє ресурсу займати високі позиції у пошуковій видачі. Пріоритетом тут є семантика.

Для просування інтернет-магазину в органічному пошуку передбачено виконання наступних кроків:

- за допомогою інструментів Google Search Console виконується реєстрація карти сайту sitemap.xml для прискорення обходу нових сторінок каталогу взуття;
- шляхом налаштування канонічних посилань у тегах сторінок запобігається поява дублікатів контенту під час використання різних фільтрів або параметрів сортування;
- через оптимізацію зображень у форматі WebP досягається високий показник LCP (Largest Contentful Paint), що позитивно впливає на загальний рейтинг домену.

Щодо соціальних мереж, активна присутність бренду «Comfort» у Instagram та TikTok дозволяє залучити найбільш активну частину аудиторії через візуальний контент. Візуал має велике значення. Для автоматизації збору аналітики в код фронтенду інтегрується Meta Pixel, який відстежує дії відвідувачів від перегляду кросівок до фіксації покупки. Дані збираються миттєво. Завдяки налаштуванню API Conversion забезпечується передача подій безпосередньо з сервера NestJS, що дозволяє обходити обмеження блокувальників реклами та точніше налаштувати ретаргетинг.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

З метою швидкого отримання перших замовлень доцільно використовувати платні інструменти просування у рекламній мережі Google Ads. Конверсія зростає дуже швидко. Шляхом запуску товарних оголошень (Shopping Ads) вдається продемонструвати конкретні моделі взуття з цінами безпосередньо у результатах пошуку потенційним покупцям. Витрати бюджету контролюються. Для підвищення ефективності кампаній застосовується динамічний ремаркетинг, який нагадує користувачам про переглянуті раніше моделі спортивного взуття через банери на сторонніх ресурсах.

Під час експлуатації системи особлива увага приділяється регламентним роботам із технічної підтримки для гарантування безперебійного доступу до сервісу. Надійність є головним пріоритетом. Процес обслуговування включає наступний перелік обов'язкових операцій:

- за допомогою вбудованих інструментів MongoDB Atlas виконується щоденне автоматичне створення резервних копій бази даних для запобігання втраті інформації про замовлення;
- шляхом моніторингу логів сервера через консоль Render виявляються та оперативно усуваються помилки виконання API-запитів або затримки у з'єднанні;
- через регулярне оновлення версій пакетів Node.js та виправлення вразливостей у бібліотеках підтримується високий рівень безпеки персональних даних клієнтів.

Інфраструктура залишається повністю захищеною.

Для подальшого розвитку проєкту передбачено розширення функціональних можливостей адміністративної панелі за рахунок інтеграції складніших аналітичних дашбордів. Аналітика допомагає бізнесу рости. Побудова системи підтримки користувачів через Telegram-ботів дозволить автоматизувати відповіді на типові питання щодо доставки взуття та наявності розмірів. Це розвантажує менеджерів. Завдяки масштабованій архітектурі NestJS впровадження нових модулів відбувається без зупинки роботи основного сайту, що підкреслює комерційну цінність розробленого програмного продукту «Comfort».

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини даного дипломного проєкту є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки веб-сайту магазину спортивного взуття «Comfort», прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є веб-сайт магазину спортивного взуття «Comfort».

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 вебсайту.

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки «Comfort». Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 4.1.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						60
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	8
2	Розробка технічного завдання	Керівник проєкту	6
		Бекенд розробник	4
3	Дизайн інтерфейсу	Дизайнер	26
4	Розробка функціоналу	Бекенд розробник	30
		Фронтенд розробник	30
5	Тестування та відладка	Бекенд розробник	8
		Фронтенд розробник	8
6	Документування	Керівник проєкту	4
7	Розгортання та підтримка	Бекенд розробник	6
8	Управління проєктом	Керівник проєкту	10
Разом			140

Сумарний час виконання операцій технологічного процесу становить 140 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 «Comfort».

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:

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

де: T_c – тарифна ставка, грн (приймаємо для керівника проєкту – 450 гривень/годину, бекенд розробника – 350 гривень/годину, фронтенд розробника – 300 гривень/годину, дизайнера – 250 гривень/годину); K_r – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

Керівника проєкту $Z_{\text{осн.}} = 28 * 450 = 12\,600$ грн.

Бекенд розробника $Z_{\text{осн.}} = 48 * 350 = 16\,800$ грн.

Фронтенд розробника $Z_{\text{осн.}} = 38 * 300 = 11\,400$ грн.

Дизайнера $Z_{\text{осн.}} = 26 * 250 = 6\,500$ грн.

Сумарна основна заробітна плата становить

$$Z_{\text{осн.}} = 12\,600 + 16\,800 + 11\,400 + 6\,500 = 47\,300 \text{ грн.}$$

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

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

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

Отже додаткова заробітна плата по категоріях працівників становить:

Керівника проєкту $Z_{\text{дод.}} = 12\,600 * 0,1 = 1\,260$ грн.

Бекенд розробника $Z_{\text{дод.}} = 16\,800 * 0,1 = 1\,680$ грн.

Фронтенд розробника $Z_{\text{дод.}} = 11\,400 * 0,1 = 1\,140$ грн.

Дизайнера $Z_{\text{дод.}} = 6\,500 * 0,1 = 650$ грн.

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод.}} = 1\,260 + 1\,680 + 1\,140 + 650 = 4\,730 \text{ грн.}$$

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

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

$$V_{\text{о.п.}} = 47\,300 + 4\,730 = 52\,030 \text{ грн.}$$

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

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

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

$$V_{\text{есв}} = 52\,030 * 0,22 = 11\,447 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п / п	Категорія працівникі в	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЕСВ, грн.	Всього витрати на оплату праці, грн. 6 = 3+4+5
		Тарифна ставка, грн.	К- сть годи н	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Керівник проекту	450	28	12 600	1 260	3 049	16 909
2	Бекенд розробник	350	48	16 800	1 680	4 066	22 546
3	Фронтенд розробник	300	38	11 400	1 140	2 759	15 299
4	Дизайнер	250	26	6 500	650	1 573	8 723
Разом				47 300	4 730	11 447	63 477

Отже, загальні витрати на оплату праці становлять 63 477 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

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

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.

$$Z_{\text{ек}} = 0,3 * 140 * 15,94 = 669 \text{ грн.}$$

Витрати на електроенергію становлять 669 грн.

4.4 Розрахунок суми амортизаційних відрахувань вебсайту

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:

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

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

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

$$A = \frac{20\,000,00 * 0,04}{150} * 140 = 747 \text{ грн.}$$

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

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.

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

де: H_B – накладні витрати.

$$H_B = 52\,030 * 0,4 = 20\,812 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості вебсайту

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

Таблиця 4.4 - Кошторис витрат «Comfort».

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	63 477	74
2.	Витрати на електроенергію	669	1
3.	Амортизаційні відрахування	747	1
4.	Накладні витрати	20 812	24
5.	Собівартість	85 705	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 $C_B = 85\,705$ грн

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

4.7 Розрахунок ціни вебсайту

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

Ціну робіт можна визначити за формулою:

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

де: $C_{\text{в}}$ – собівартість; $P_{\text{рен}}$ – рівень рентабельності; ПДВ – ставка податку на додану вартість.

$$Ц = 85\,705 * (1 + 0,3) * (1 + 0,2) = 133\,700 \text{ грн.}$$

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).

$$\text{ЧТВ} = -C_{\text{в}} + \sum_{i=1}^t \frac{\Gamma_{\text{п}}}{(1+i)^i}, \quad (4.11)$$

де: $C_{\text{в}}$ – собівартість розробки; $\Gamma_{\text{п}}$ – грошовий потік за t – ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$\text{ЧТВ} = -85\,705 + \frac{47994}{(1+0,1)^1} + \frac{47994}{(1+0,1)^2} + \frac{44\,994}{(1+0,1)^3} = 33\,349 \text{ грн}$$

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

Термін окупності визначається за формулою:

$$T_{\text{ок}} = T_{\text{пв}} + \frac{N_{\text{в}}}{\Gamma_{\text{пр}}} \quad (4.12)$$

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

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

$$T_{\text{ок}} = 2 + \frac{2\,735}{47\,994} = 2,1 \text{ р.}$$

Всі дані внесемо в зведену таблицю 4.5.

Таблиця 4.5 – Техніко-економічні показники «Comfort».

№ п/п	Показник	Значення
1.	Собівартість, грн.	85 705
2.	Плановий прибуток або грошовий потік, грн.	47 994
3.	Ціна, грн.	133 700
4.	Чиста теперішня вартість, грн.	33 349
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 вебсайту «Comfort».є доцільною з економічної точки зору.

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Професійні захворювання та травми.

Охорона праці є важливою складовою забезпечення безпечних умов діяльності на будь-якому сучасному підприємстві. Вивчення причин виробничого травматизму та виникнення професійних захворювань дозволяє розробити ефективні заходи для їхнього запобігання [16]. Виробнича травма визначається як раптове погіршення стану здоров'я працівника внаслідок дії небезпечного виробничого фактора під час виконання трудових обов'язків. Професійне захворювання виникає в результаті тривалого та систематичного впливу шкідливих умов праці на організм працюючого [16].

Для успішної профілактики негативних наслідків необхідно детально вивчати чинники робочого середовища. Основними небезпечними та шкідливими факторами, що призводять до травм та захворювань, є такі джерела:

- Шкідливі хімічні речовини спричиняють гострі або хронічні отруєння працівників;
- Фізичні чинники у вигляді підвищеного шуму та вібрації призводять до зниження слуху;
- Біологічні фактори викликають інфекційні захворювання у медичного та лабораторного персоналу;
- Психофізіологічні перевантаження стають причиною швидкого зниження концентрації уваги працівника;

Аналіз причин виникнення небезпечних ситуацій свідчить про необхідність постійного контролю за станом робочих місць. Для систематизації даних про виробничий травматизм застосовують класифікацію за різними ознаками [16]. Це дозволяє чітко визначити основні джерела небезпеки та впровадити відповідні інженерні чи організаційні рішення для захисту персоналу підприємства.

Для наочного представлення основних категорій професійних патологій та

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

умов їхнього виникнення доцільно використовувати узагальнену структуровану інформацію. Співвідношення між чинниками та наслідками наведено у таблиці 5.1.

Таблиця 5.1 - Класифікація факторів та наслідків

Група факторів	Основні наслідки для здоров'я	Приклади професій
Фізичні чинники	Розвиток вібраційної хвороби та професійної глухоти	Оператори верстатів
Хімічні чинники	Хронічні токсичні ураження органів дихання	Хімічні технологи
Ергономічні чинники	Захворювання опорно-рухового апарату та хребта	Офісні працівники

Дані наведені у таблиці вказують на різноманітність факторів ризику на робочих місцях. Своєчасне проведення медичних оглядів допомагає виявити перші ознаки професійних патологій [16]. Це мінімізує негативні наслідки для здоров'я працівників та знижує економічні втрати підприємства від непрацездатності персоналу.

5.2 Види електричних травм

Електричний струм становить підвищену небезпеку через відсутність видимих ознак загрози до моменту безпосереднього контакту людини зі струмоведучими частинами [2]. Електричні травми поділяють на два основні види, якими є місцеві електричні травми та загальні електричні травми або електричні удари. Місцеві травми характеризуються чітко вираженим локальним порушенням цілісності тканин організму під дією струму або електричної дуги [17].

Місцеві пошкодження мають різний характер залежно від умов виникнення аварійної ситуації та параметрів електричної мережі. Серед них виділяють наступні види локальних уражень:

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

- Електричні опіки виникають внаслідок теплової дії струму на шкіру та глибокі тканини людини;
- Електричні знаки являють собою чітко окреслені плями сірого або жовтого кольору на поверхні шкіри;
- Металізація шкіри характеризується проникненням найдрібніших частинок розплавленого дугою металу у верхні шари шкірного покриву;
- Електроофтальмія проявляється як запалення зовнішніх оболонок очей через потужне випромінювання дуги;
- Механічні пошкодження виникають внаслідок судомного скорочення м'язів під дією струму;

Кожен із зазначених видів місцевих пошкоджень вимагає специфічного підходу до надання першої медичної допомоги та подальшого лікування. Проте найбільшу небезпеку для життя людини становлять загальні електричні травми [17]. Електричний удар порушує нормальну діяльність життєво важливих органів та систем організму, що може призвести до летальних наслідків.

За ступенем важкості та клінічними проявами електричні удари поділяють на кілька категорій. Детальний опис цих станів наведено у таблиці 5.2.

Таблиця 5.2 - Ступені електричних ударів

Ступінь удару	Характерні клінічні прояви	Вплив на життєдіяльність
Перший ступінь	Судомне скорочення м'язів без втрати свідомості	Організм зберігає основні функції
Другий ступінь	Судомне скорочення м'язів із втратою свідомості	Потрібен постійний нагляд за постраждалим
Третій ступінь	Втрата свідомості та порушення серцевої діяльності	Виникає пряма загроза для життя людини
Четвертий ступінь	Клінічна смерть внаслідок зупинки дихання	Необхідне негайне проведення реанімаційних заходів

Наведена класифікація ступенів електричних ударів демонструє стрімке зростання небезпеки при збільшенні часу дії струму. Для запобігання таким важким наслідкам на підприємствах впроваджують системи захисного заземлення, занулення та автоматичного відключення живлення [17]. Інструктаж персоналу та використання засобів індивідуального захисту є обов'язковими умовами безпечної роботи з електроустановками.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

ВИСНОВКИ

У результаті виконання представленої кваліфікаційної роботи було успішно спроектовано та реалізовано комплексну інформаційну систему «Comfort» для повної автоматизації процесів реалізації спортивного взуття. Цей результат повністю відповідає вимогам ТЗ. Під час розробки було охоплено повний життєвий цикл програмного виробу, що включав етапи від глибокого аналізу предметної області до фінального розгортання в мережі Інтернет. Це робить проєкт завершеним.

Для досягнення поставленої мети було проведено ґрунтовне дослідження ринку електронної комерції та сформовано перелік функціональних вимог для забезпечення конкурентних переваг магазину. Саме модульною є архітектура системи тепер. Зокрема, шляхом побудови чіткої карти сайту та інтерактивних макетів інтерфейсу вдалося реалізувати зручну навігацію, яка адаптується під будь-які типи сучасних мобільних пристроїв. Всі елементи працюють злагоджено.

В основу збереження даних покладено нереляційну модель MongoDB Atlas, яка дозволяє гнучко керувати об'ємними масивами інформації про товари, замовлення та профілі зареєстрованих клієнтів. За допомогою програмних засобів забезпечується цілісність даних. Шляхом впровадження динамічних схем та налаштування логічних зв'язків між колекціями було створено стійку структуру, що підтримує оперативне оновлення складських залишків у реальному часі. База працює надзвичайно швидко.

Під час безпосереднього програмування було задіяно сучасний технологічний стек на базі фреймворків NestJS та Next.js 14 для створення високопродуктивної клієнт-серверної архітектури. Повноцінна взаємодія компонентів відбувається через API. Через використання механізму JWT-авторизації та захищених httpOnly cookies забезпечено високий рівень безпеки персональних даних, тоді як бібліотека TailwindCSS дозволила реалізувати лаконічний дизайн. Головним пріоритетом виступала швидкість відгуку.

Отримані результати підтверджують готовність інформаційної системи до

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

промислової експлуатації в реальних умовах роздрібної торгівлі та її здатність витримувати пікові навантаження під час масових продажів. Великий потенціал для розширення має даний проєкт. У подальшому розвиток продукту «Comfort» може передбачати впровадження інтелектуальних систем рекомендацій на основі вподобань покупців та підключення нових платіжних сервісів для розширення географії збуту. Максимально гнучким залишається обране рішення завжди.

Важливим практичним наслідком роботи стало набуття професійних навичок у сфері fullstack-розробки та вдосконалення вміння приймати складні архітектурні рішення для вирішення реальних бізнес-завдань. Набутий досвід є надзвичайно корисним сьогодні. Крім технічних аспектів, було значно покращено здатність до системного аналізу проблем та формування технічної документації відповідно до державних стандартів, що є критичним для сучасного інженера.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
						73
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Next.js Documentation : вебсайт. URL: <https://nextjs.org/docs> (дата звернення: 14.05.2024).
- 2) NestJS – A progressive Node.js framework : вебсайт. URL: <https://docs.nestjs.com> (дата звернення: 14.05.2024).
- 3) MongoDB Manual : вебсайт. URL: <https://www.mongodb.com/docs/manual> (дата звернення: 15.05.2024).
- 4) Чин Т. Node.js. Розробка серверних застосунків. Київ : Діалектика, 2022. 320 с.
- 5) TypeScript Handbook : вебсайт. URL: <https://www.typescriptlang.org/docs/handbook> (дата звернення: 16.05.2024).
- 6) Фрімен А. Pro Next.js. London : Apress, 2023. 450 р.
- 7) Mongoose v8.0.0 Documentation : вебсайт. URL: <https://mongoosejs.com/docs> (дата звернення: 17.05.2024).
- 8) Tailwind CSS Docs : вебсайт. URL: <https://tailwindcss.com/docs> (дата звернення: 18.05.2024).
- 9) Zustand Library : вебсайт. URL: <https://docs.pmnd.rs/zustand> (дата звернення: 19.05.2024).
- 10) Мартін Р. Чистий код. Львів : Фабула, 2019. 448 с.
- 11) Cloudinary API Reference : вебсайт. URL: <https://cloudinary.com/documentation> (дата звернення: 20.05.2024).
- 12) Passport.js – Simple, unobtrusive authentication for Node.js : вебсайт. URL: <https://www.passportjs.org/docs> (дата звернення: 21.05.2024).
- 13) JSON Web Tokens (JWT) : вебсайт. URL: <https://jwt.io/introduction> (дата звернення: 22.05.2024).
- 14) Node.js v20.x API Documentation : вебсайт. URL: <https://nodejs.org/api> (дата звернення: 23.05.2024).
- 15) Веб-технології та розробка веб-застосунків / О.В. Шевченко, М.А. Іванов, С.П. Кравченко та Петренко О.М. Київ : Політехніка, 2023. 412 с.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

16) Курепін В.М. Основи охорони праці : навчальний посібник. Миколаїв : МНАУ, 2022. 347 с.

17) 2. Левченко О.Г., Денисюк С.П., Каштанов С.Ф. Електробезпека в енергетиці : навчальний посібник. Київ : Видавець ФО-П Піча Ю.В., 2022. 280 с.

					2026.КВР.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

ДОДАТКИ

Додаток А. Текст головного модуля сервера «main.ts»

```
import { NestFactory } from '@nestjs/core';
import { ValidationPipe } from '@nestjs/common';
import { ConfigService } from '@nestjs/config';
import * as dns from 'node:dns';
import cookieParser from 'cookie-parser';
import { AppModule } from './app.module';

dns.setServers(['8.8.8.8', '8.8.4.4']);

async function bootstrap() {
  const app = await NestFactory.create(AppModule);
  const configService = app.get(ConfigService);

  app.setGlobalPrefix('api');
  app.useGlobalPipes(
    new ValidationPipe({
      whitelist: true,
      forbidNonWhitelisted: true,
      transform: true,
    }),
  );
  app.use(cookieParser());
  app.enableCors({
    origin: configService.get<string>('FRONTEND_URL'),
    credentials: true,
  });

  const port = configService.get<number>('PORT') || 5000;
  await app.listen(port);
}bootstrap();
```

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		76

Додаток Б. Схема моделі даних користувача «user.schema.ts»

```
import { Prop, Schema, SchemaFactory } from '@nestjs/mongoose';
import { Document, Types } from 'mongoose';

export type UserDocument = User & Document;

@Schema({ timestamps: true })
export class User {
  @Prop({ required: true, unique: true, lowercase: true, trim: true })
  email: string;

  @Prop({ required: true, select: false })
  passwordHash: string;

  @Prop({ required: true })
  firstName: string;

  @Prop({ required: true })
  lastName: string;

  @Prop({ required: true })
  phone: string;

  @Prop({ default: 'client', enum: ['client', 'admin'] })
  role: string;

  @Prop({ type: [{ type: Types.ObjectId, ref: 'Product' }], default: [] })
  wishlist: Types.ObjectId[];
  @Prop({ select: false })
  refreshToken: string;
}

export const UserSchema = SchemaFactory.createForClass(User);
```

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		77

Додаток В. Логіка сервісу авторизації «service.ts»

```
@Injectable()
export class AuthService {
  constructor(
    private usersService: UsersService,
    private jwtService: JwtService,
    private configService: ConfigService, ) {}
  async login(loginDto: LoginDto) {
    const user = await this.usersService.findByEmail(loginDto.email);
    if (!user) {
      throw new UnauthorizedException('Невірний email або пароль'); }
    const passwordMatches = await bcrypt.compare(loginDto.password,
user.passwordHash);
    if (!passwordMatches) {
      throw new UnauthorizedException('Невірний email або пароль'); }
    const tokens = await this.getTokens(user._id.toString(), user.email,
user.role);
    await this.usersService.updateRefreshToken(user._id.toString(),
tokens.refreshToken);
    return tokens; }
  async getTokens(userId: string, email: string, role: string) {
    const payload = { sub: userId, email, role };
    const accessToken = await this.jwtService.signAsync(payload, {
      secret: this.configService.get('JWT_ACCESS_SECRET'),
      expiresIn: '15m', });
    const refreshToken = await this.jwtService.signAsync(payload, {
      secret: this.configService.get('JWT_REFRESH_SECRET'),
      expiresIn: '7d', });
    return { accessToken, refreshToken }; }
}
```

									Арк.
									78
Зм.	Арк.	№ докум.	Підпис	Дата	2026.КВР.122.423.14.00.00 ПЗ				

Додаток Г. Сервіс обробки замовлень «orders.service.ts»

```
@Injectable()
export class OrdersService {
  constructor(
    @InjectModel(Order.name) private orderModel: Model<OrderDocument>,
    @InjectModel(Product.name) private productModel:
Model<ProductDocument>,
  ) {}

  async create(createOrderDto: CreateOrderDto, userId?: string):
Promise<OrderDocument> {
    for (const item of createOrderDto.items) {
      const product = await this.productModel.findById(item.productId);
      if (!product) throw new NotFoundException('Товар не знайдено');

      const sizeInfo = product.sizes.find((s) => s.size === item.size);
      if (!sizeInfo || sizeInfo.count < item.quantity) {
        throw new BadRequestException('Недостатньо товару на складі');
      }

      await this.productModel.updateOne(
        { _id: item.productId, 'sizes.size': item.size },
        { $inc: { 'sizes.$.count': -item.quantity } },
      );
    }
    const newOrder = new this.orderModel({ ...createOrderDto, userId: userId
|| null });
    return newOrder.save();
  }
}
```

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

Додаток Д. Сховище управління станом кошика «useCartStore.ts»

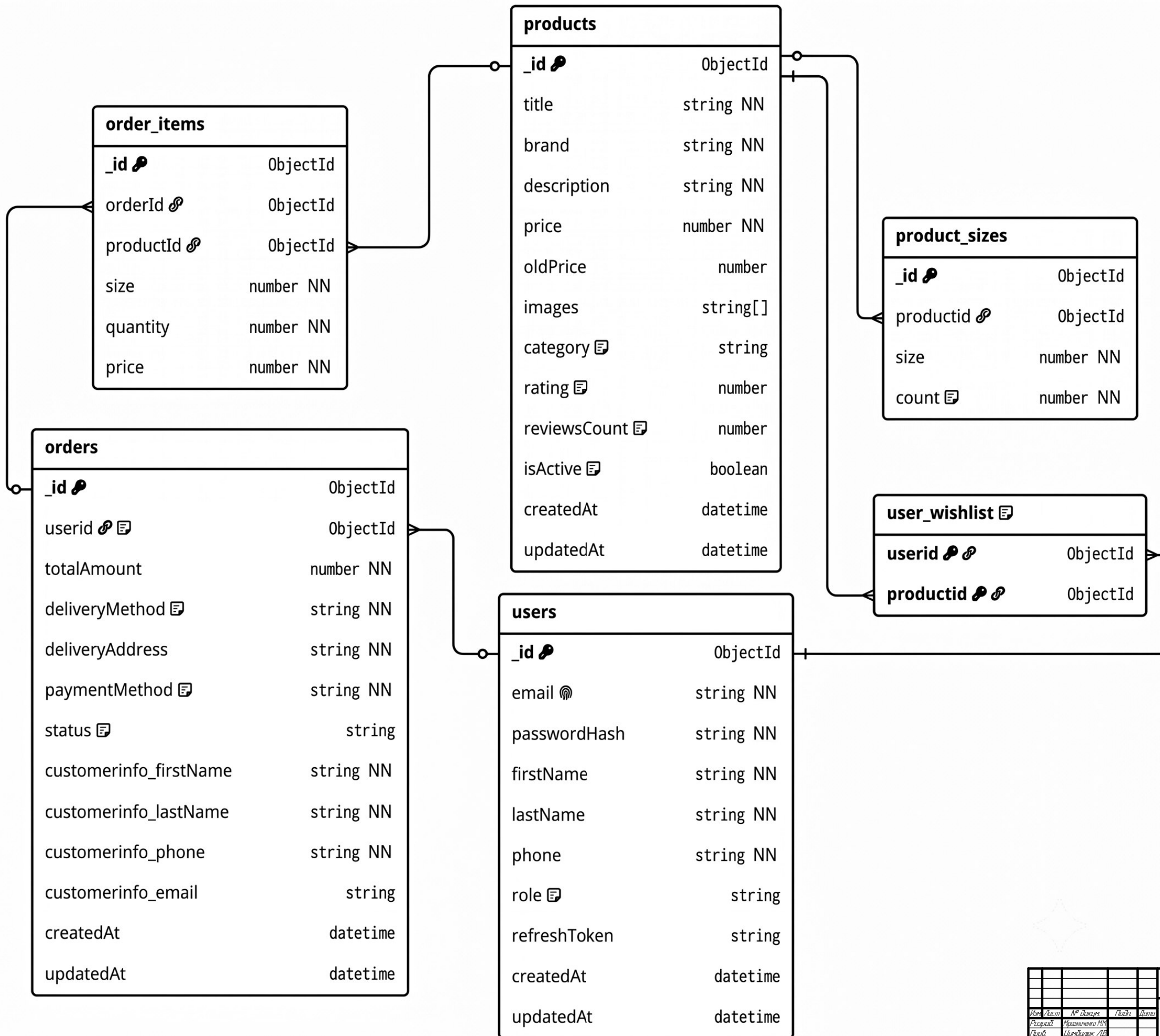
```
export const useCartStore = create<CartState>()(
  persist(
    (set, get) => ({
      items: [],
      addItem: (newItem) => {
        const currentItems = get().items;
        const existingItemIndex = currentItems.findIndex(
          (item) => item.productId === newItem.productId && item.size ===
newItem.size
        );

        if (existingItemIndex > -1) {
          const updatedItems = [...currentItems];
          updatedItems[existingItemIndex].quantity += newItem.quantity;
          set({ items: updatedItems });
        } else {
          set({ items: [...currentItems, newItem] });
        }
      },
      clearCart: () => set({ items: [] }),
      getTotalPrice: () => {
        return get().items.reduce((total, item) => total + item.price *
item.quantity, 0);
      },
    })),
    { name: 'comfort-cart' }
  )
);
```

					2026.KBP.122.423.14.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		80

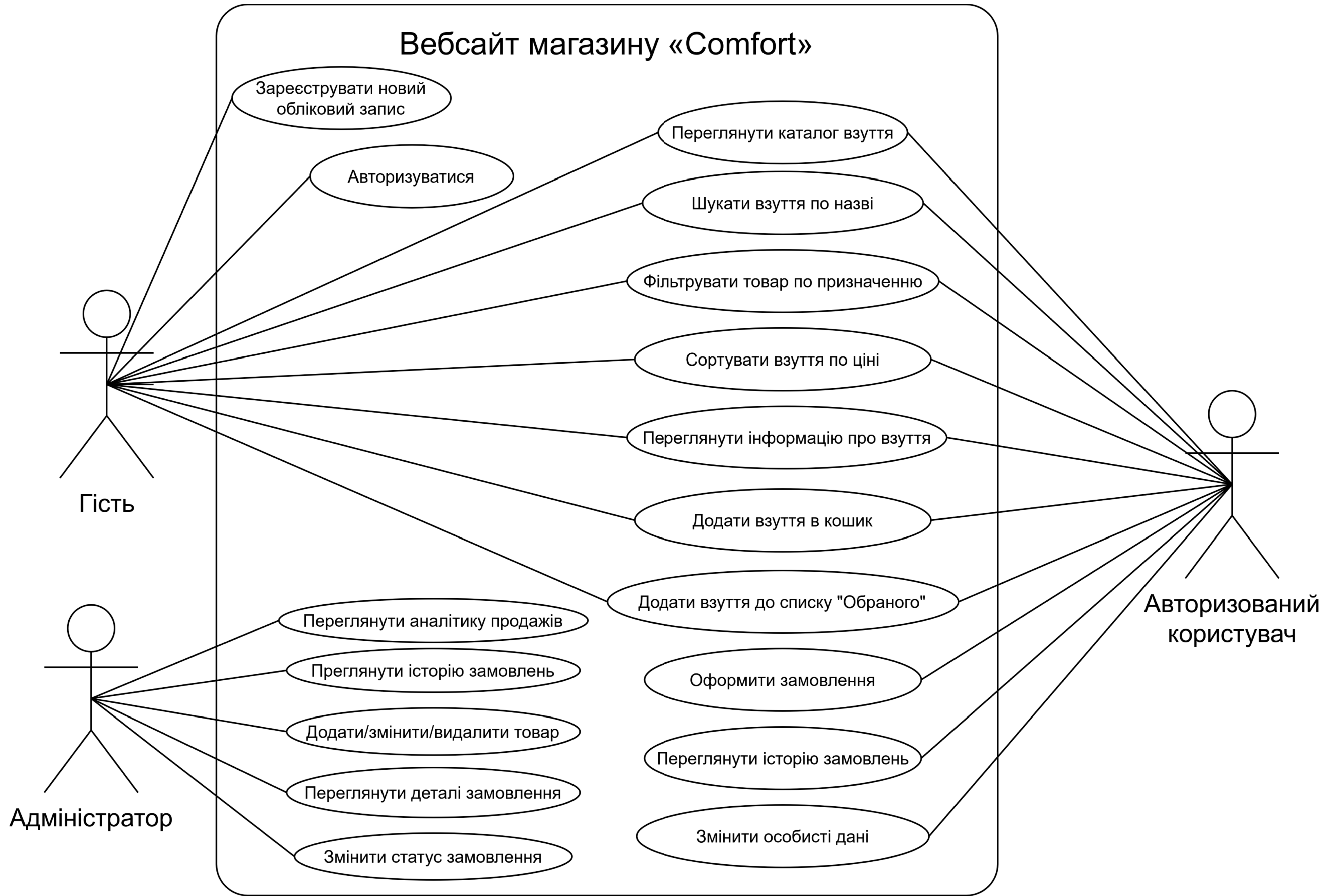
ER-діаграма бази даних

2026.KBP.122.4.23.14.00.00 БД



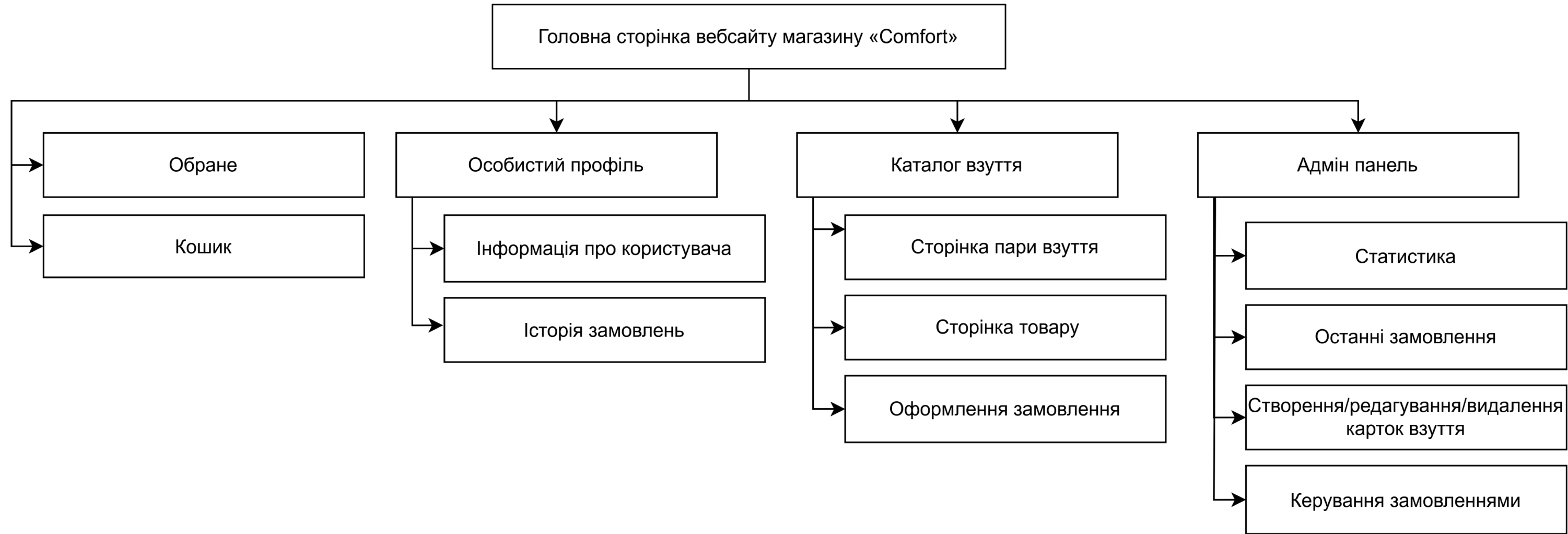
Діаграма варіантів використання

2026.KBP.122.4.23.14.00.00 ДВ



2026.KBP.122.4.23.14.00.00 ДВ				Лист	Масштаб
Изм.	Лист	№ докум.	Дата	Лист	Масштаб
Разраб.	Машинченко ММ				
Проб.	Цимбалюк ДВ				
Текстпр.					
Рецензент					
Исполн.	Приймак В.А.				
Чит.					
Разработка вебсайта магазина спортивного взуття «Comfort»				Лист	Листов 1
Діаграма варіантів використання				ВСП ТФК ТНТУ	КН-4.23
				м. Тернопіль	
Копірава				Формат А1	

Схема структурна вебсайту магазину



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

Таблиця техніко-економічних показників

№	Показник	Одиниця вимірювання	Значення
1	Кількість основних API-ендпоінтів (маршрутів)	одиниця	18
2	Фреймворк серверної частини	–	NestJS
3	Фреймворк клієнтської частини	–	Next.js 16
4	Загальний обсяг вихідного коду	рядок	6 038
5	Кількість файлів у структурі Frontend	одиниць	45
6	Кількість файлів у структурі Backend	одиниць	45
7	Кількість сутностей у базі даних	штук	3
8	Собівартість	грн	85 705
9	Плановий грошовий потік	грн	47 994
10	Ціна	грн	133 700
11	Чиста теперішня вартість	грн	33 394
12	Термін окупності	рік	2,1

				2026.KBP.122.4.23.14.00.00 ТБ			
Изм.	Лист	№ вказ.	Лист	Листа	Розробка вебсайту магазину спортивного взуття «Світлог»	Лист	Масштаб
Розроб.	Машиничко МВ				Таблиця техніко-економічних показників	Лист	Листа 1
Проб.	Цимбалюк ЛВ					ВСП ТФК ТНТУ	КН-4.23
Іконтр.						м. Тернопіль	
Рецензент							
Іконтр.	Пріймак В.А.						
Чиб.							