

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка інтернет-магазину косметики Lorena засобами
Next.js, Stripe, Tailwind CSS та MongoDB

Виконав: студент IV курсу, групи СН-43

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Тарас Х.І.

(прізвище та ініціали)

Керівник

(підпис)

Дмитроца Л.П.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Гащин Н.Б.

(прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 27 » січня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Тарас Христина Іванівна
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтернет-магазину косметики Lorena засобами Next.js, Stripe, Tailwind CSS та MongoDB

Керівник роботи Дмитроца Леся Павлівна, к.т.н., доцент кафедри комп'ютерних наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 07 » травня 2025 року № 4/7-444

2. Термін подання студентом завершеної роботи 27 червня 2025р.

3. Вихідні дані до роботи Наукові публікації, інтернет-джерела та технічна документація щодо розробки веб-застосунків з використанням фреймворку Next.js, CSS-фреймворку Tailwind CSS, платіжної системи Stripe, бази даних MongoDB, а також матеріали щодо створення та адміністрування електронної комерції (e-commerce) та реалізації функціоналу адміністративної панелі в сучасних веб-додатках.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз предметної області та постановка завдання на розробку інтернет-магазину косметики Lorena. 2. Проектування та реалізація інтернет-магазину Lorena. 3. Практична реалізація та тестування інтернет-магазину Lorena. 4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Тема та актуальність дослідження. 2. Мета та завдання кваліфікаційної роботи. 3. Аналіз ринку інтернет-магазинів косметики. 4. Функціональні вимоги до сучасних e-commerce рішень. 5. Огляд конкурентів: сильні та слабкі сторони. 6. Постановка задачі створення інтернет-магазину Lorena. 7. Обґрунтування вибору технологій. 8. Архітектура інтернет-магазину Lorena. 9. Реалізація клієнтської частини сайту. 10. Реалізація адміністративної панелі керування. 11. Особливості дизайну та адаптивності інтерфейсу. 12. Інтеграція платіжної системи Stripe та автентифікація через Clerk. 13. Тестування, валідація та розгортання проекту. 14. Практична цінність розробки та перспективи впровадження. 15. Висновки та результати виконаної роботи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С., кандидат технічних наук, доцент кафедри МТ		

7. Дата видачі завдання 27 січня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	27.01.2025	
2.	Підбір та опрацювання літературних джерел по темі кваліфікаційної роботи	08.05.2025-15.05.2025	
3.	Виконання дослідження щодо сучасного стану та тенденцій розвитку ринку онлайн-косметики, аналізу конкурентів та предметної області, формулювання завдання та визначення функціональних та нефункціональних вимог до системи Розроблення архітектури інтернет-магазину, структури баз даних, користувацького інтерфейсу та серверної частини відповідно до визначених функціональних вимог	16.05.2025-22.05.2025	
4.	Оформлення розділу «Аналіз предметної області та постановка завдання на розробку інтернет-магазину косметики Lorena»	23.05.2025-01.06.2025	
5.	Оформлення розділу «Проектування та реалізація інтернет-магазину Lorena»	23.05.2025-01.06.2025	
6.	Оформлення розділу «Практична реалізація та тестування інтернет-магазину Lorena»	23.05.2025-01.06.2025	
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	06.06.2025-07.06.2025	
8.	Виконання завдання до підрозділу «Основи охорони праці»	08.06.2025-09.06.2025	
9.	Оформлення кваліфікаційної роботи	06.06.2025-10.06.2025	
10.	Нормоконтроль	11.06.2025-13.06.2025	
11.	Перевірка на плагіат	16.06.2025	
12.	Попередній захист кваліфікаційної роботи	18.06.2025	
13.	Захист кваліфікаційної роботи	28.06.2025	

Студент

(підпис)

Тарас Х.І.

(прізвище та ініціали)

Керівник роботи

(підпис)

Дмитроца Л.П.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка інтернет-магазину косметики Lorena засобами Next.js, Stripe, Tailwind CSS та MongoDB // Кваліфікаційна робота освітнього рівня «Бакалавр» // Тарас Христина Іванівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-43 // Тернопіль, 2025 // С. 82, рис. – 31 , табл. – 2 , додат. – 2 , бібліогр. – 52 .

Ключові слова: інтернет-магазин, веброзробка, електронна комерція, next.js, tailwind css, mongoDB, stripe, clerk.

Кваліфікаційна робота присвячена дослідженню процесу створення сучасного інтернет-магазину косметики з використанням технологій веброзробки.

У першому розділі кваліфікаційної роботи описано актуальні тенденції розвитку електронної комерції в Україні та світі. Висвітлено ключові вимоги до функціоналу сучасних онлайн-магазинів. Розглянуто аналоги та конкурентні рішення. Проаналізовано вибір технологій, необхідних для реалізації проєкту.

У другому розділі кваліфікаційної роботи описано архітектуру застосунку, структуру даних, обрані технології та бібліотеки. Досліджено особливості реалізації клієнтської та адміністративної частин. Подано опис функціоналу, що забезпечує повноцінну роботу магазину.

У третьому розділі кваліфікаційної роботи описано процес розгортання вебзастосунку, тестування інтерфейсу та перевірку продуктивності. Проаналізовано результати, отримані за допомогою інструментів оптимізації. Проведено перевірку на відповідність сучасним вимогам користувацького досвіду.

Об'єкт дослідження: процес розробки вебзастосунку для електронної комерції.

Предмет дослідження: технології створення функціонального інтернет-магазину з використанням Next.js, Tailwind CSS, MongoDB, Stripe та Clerk.

ANNOTATION

Development of the Lorena Cosmetics Online Store Using Next.js, Stripe, Tailwind CSS, and MongoDB // Qualification work of the educational level "Bachelor" // Taras Khrystyna Ivanivna // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-43 // Ternopil, 2025 // P. 82 , fig. - 31, tabl. - 2, annexes. – 2, references - 52.

Keywords: online store, web development, e-commerce, next.js, tailwind css, mongoDB, stripe, clerk.

The qualification work is devoted to the study of the process of creating a modern online cosmetics store using web development technologies.

The first chapter of the qualification work describes the current trends in the development of e-commerce in Ukraine and the world. The key requirements for the functionality of modern online stores are highlighted. Analogues and competitive solutions are considered. The choice of technologies required for project implementation is analyzed.

The second section of the qualification work describes the application architecture, data structure, selected technologies, and libraries. The features of the client and administrative parts are investigated. A description of the functionality that ensures the full operation of the store is provided.

The third chapter of the qualification work describes the process of deploying the web application, testing the interface, and checking its performance. The results obtained with the help of optimization tools are analyzed. The results obtained with the help of optimization tools are analyzed.

Object of research: the process of developing a web application for e-commerce.

The subject of research: technologies for creating a functional online store using Next.js, Tailwind CSS, MongoDB, Stripe, and Clerk.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

2FA (англ. Two-Factor Authentication) – двофакторна автентифікація.

API (англ. Application Programming Interface) – програмний інтерфейс прикладного програмування.

CI/CD (англ. Continuous Integration / Continuous Deployment) – безперервна інтеграція та безперервне розгортання.

CLS (англ. Cumulative Layout Shift) – кумулятивне зсування макета.

CRUD (англ. Create, Read, Update, Delete) – базові операції з даними.

CSS (англ. Cascading Style Sheets) – каскадні таблиці стилів.

CSR (англ. Client-Side Rendering) – клієнтський рендеринг сторінки.

ER (англ. Entity-Relationship) – модель "сутність-зв'язок".

HTML (англ. HyperText Markup Language) – мова гіпертекстової розмітки.

HTTP (англ. HyperText Transfer Protocol) – протокол передавання гіпертексту.

JSON (англ. JavaScript Object Notation) – формат обміну даними.

KPI (англ. Key Performance Indicators) – ключові показники ефективності.

ORM (англ. Object-Relational Mapping) – об'єктно-реляційне відображення.

SDK (англ. Software Development Kit) – набір інструментів для розробника.

SEO (англ. Search Engine Optimization) – оптимізація для пошукових систем.

SSG (англ. Static Site Generation) – генерація статичних сторінок.

SSR (англ. Server-Side Rendering) – серверний рендеринг сторінки.

TTI (англ. Time to Interactive) – час до повної інтерактивності сторінки.

МСБ – малий і середній бізнес.

ПК – персональний комп'ютер.

UI/UX – інтерфейс користувача / досвід користувача.

ШІ – штучний інтелект.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ІНТЕРНЕТ-МАГАЗИНУ КОСМЕТИКИ LORENA	11
1.1 Аналіз ринку інтернет-магазинів косметики та їх функціональних вимог	11
1.1.1 Сучасні тенденції у сфері електронної комерції косметики	11
1.1.2 Функціональні вимоги до сучасних e-commerce рішень	12
1.1.3 Роль адміністративної панелі в управлінні інтернет-магазином	14
1.2 Аналітичний огляд існуючих інтернет-магазинів косметики	15
1.3 Постановка завдання розробки інтернет-магазину косметики з адміністративною панеллю.....	18
1.3.1 Область застосування та призначення розробки інтернет- магазину Lorena.....	18
1.3.2 Вимоги до функціоналу сайту інтернет-магазину Lorena	19
1.3.3 Стадії та етапи розробки інтернет-магазину	20
1.4 Вибір та обґрунтування використаних технологій	22
1.4.1 Обґрунтування вибору фреймворку Next.js	22
1.4.2 Використання Tailwind CSS для UI.....	23
1.4.3 Вибір MongoDB в якості бази даних для інтернет-магазину	24
1.4.4 Інтеграція з платіжною системою Stripe.....	25
1.4.5 Обґрунтування використання Clerk для автентифікації	26
1.5 Висновок до першого розділу	27
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ LORENA	29
2.1 Технологічні та архітектурні основи побудови вебзастосунку Lorena	29
2.1.1 Загальний опис архітектури інтернет-магазину	29

2.1.2 Структура файлової системи проекту.....	30
2.1.3 Бібліотеки, використані для розробки клієнтської частини проекту	33
2.1.4 Бібліотеки, використані для розробки адміністративної частини проекту	34
2.2 Моделювання бази даних	36
2.3 Реалізація функціональності клієнтської частини інтернет-магазину	38
2.3.1 Реєстрація користувачів та їх авторизація у системі.....	38
2.3.2 Структура головної сторінки та навігаційного меню	42
2.3.3 Додавання товарів у список бажань та перегляд сторінки продукту	44
2.3.4 Додавання товарів у кошик, оформлення та оплата замовлення.	47
2.4 Огляд функціоналу адміністративної частини інтернет-магазину	50
2.4.1 Реєстрація користувачів та їх авторизація у системі.....	50
2.4.2 Інформаційна панель адміністратора.....	51
2.4.3 Управління колекціями та додавання нових товарів	52
2.4.4 Управління замовленнями та клієнтами.....	55
2.5 Огляд дизайну інтернет-магазину Lorena	57
2.5.1 Кольорова схема інтернет-магазину та логотипи.....	57
2.5.2 Розробка адаптивного дизайну за допомогою Tailwind CSS	59
2.6 Висновок до другого розділу	61
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ LORENA	62
3.1 Розгортання інтернет-магазину в Інтернеті.....	62
3.2 Валідація й тестування функціональності інтернет-магазину	63
3.2.1 Валідація розмітки інтерфейсу та форм	63
3.2.2 Перевірка інтернет-магазину на кросбраузерність та тестування його швидкості	64
3.3 Підтримка та подальший розвиток інтернет-магазину Lorena.....	66
3.4 Висновок до третього розділу	66

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	68
4.1 Психологічні чинники небезпеки	68
4.2 Загальні вимоги безпеки з охорони праці для користувачів ПК	70
4.3 Висновок до четвертого розділу	72
ВИСНОВКИ.....	73
ПЕРЕЛІК ДЖЕРЕЛ	75
ДОДАТКИ	

ВСТУП

Актуальність теми. Внаслідок глобалізації та цифровізації суспільства онлайн-торгівля набула стрімкого розвитку. Особливо динамічно розвивається сегмент продажу косметичної продукції, де важливими факторами успіху є зручність користування, швидкість завантаження, адаптивний дизайн та безпечна обробка платежів. У зв'язку з цим створення сучасного, функціонального та ефективного інтернет-магазину косметики є актуальним напрямком сучасних досліджень в галузі інформаційних технологій та веброзробки.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка інтернет-магазину косметики Lorena із застосуванням сучасних вебтехнологій для забезпечення високої якості обслуговування користувачів та ефективного адміністрування. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- проаналізувати стан сучасного ринку електронної комерції у сфері косметики;
- визначити функціональні вимоги до інтернет-магазину;
- обґрунтувати вибір технологічного стеку для реалізації;
- реалізувати клієнтську та адміністративну частину застосунку;
- провести тестування продуктивності та функціональності.

Практичне значення одержаних результатів.

Результатом роботи є повноцінний інтернет-магазин, придатний до використання в реальному середовищі. Реалізовані рішення можуть бути використані як основа для запуску малого чи середнього бізнесу у сфері продажу косметики, а також адаптовані під інші ніші електронної комерції. Окрім цього, проєкт демонструє приклад ефективного поєднання сучасних технологій у веброзробці з фокусом на зручність, продуктивність та безпеку.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ІНТЕРНЕТ-МАГАЗИНУ КОСМЕТИКИ LORENA

1.1 Аналіз ринку інтернет-магазинів косметики та їх функціональних вимог

1.1.1 Сучасні тенденції у сфері електронної комерції косметики

У 2025 році електронна комерція в косметичній галузі продовжує демонструвати стійке зростання, зумовлене зміною споживчих уподобань, цифровими інноваціями та глобальними економічними тенденціями.

За даними аналітичної компанії Mordor Intelligence, обсяг світового ринку онлайн-продажів косметики у 2025 році оцінюється в 16,23 мільярда доларів США, з прогнозованим зростанням до 24,37 мільярда доларів США до 2030 року при середньорічному темпі зростання (CAGR) 8,47% [1]. Це свідчить про стабільне зростання попиту на косметичні продукти через інтернет.

У першій половині 2024 року 41% всіх продажів косметичних та персональних товарів здійснювалися через електронні платформи, що є значним зростанням порівняно з попередніми роками [2]. Такий перехід до онлайн-покупок зумовлений зручністю, широким вибором продукції та можливістю порівняння цін.

Цифрова трансформація є ключовим фактором підвищення конкурентоспроможності малого та середнього бізнесу (МСБ) в Україні, особливо у контексті європейської інтеграції. Як зазначено у статті «Особливості цифрового розвитку малого і середнього бізнесу України, країн Європи та G7» [3], ефективна цифровізація сприяє зниженню витрат, автоматизації процесів та виходу бізнесу на міжнародні ринки, що є особливо важливим для українських компаній в умовах сучасних викликів.

Сучасні споживачі очікують персоналізованого досвіду покупок. Бренди активно впроваджують аналітику даних та штучний інтелект для створення індивідуальних рекомендацій, що підвищує рівень задоволеності клієнтів та сприяє зростанню продажів.

Платформи соціальних мереж, такі як TikTok та Instagram, стали ключовими каналами для просування косметичних продуктів. Наприклад, TikTok Shop продає косметичний продукт кожні дві секунди, що підкреслює важливість соціальної комерції в сучасному маркетингу [4].

Споживачі все більше звертають увагу на етичні аспекти виробництва косметики, включаючи використання натуральних інгредієнтів, відсутність тестування на тваринах та екологічну упаковку. Це стимулює бренди до впровадження сталих практик у своїй діяльності.

Моделі підписки на косметичні продукти стають все більш популярними, оскільки забезпечують зручність для споживачів та стабільний дохід для компаній. Це також сприяє підвищенню лояльності клієнтів та зменшенню рівня відтоку.

1.1.2 Функціональні вимоги до сучасних e-commerce рішень

Інноваційні e-commerce рішення мають забезпечувати можливість виходу на міжнародний ринок без фізичної присутності, що стало можливим завдяки цифровізації. У наукових публікаціях наголошується, що навіть малий бізнес або так звані «мікро-транснаціональні корпорації» здатні продавати свою продукцію в усьому світі, використовуючи лише онлайн-платформи [5].

В провідних e-commerce рішеннях автоматизація процесів управління замовленнями стала ключовою умовою ефективності та конкурентоспроможності. Автоматизовані системи дозволяють мінімізувати людський фактор, зменшити витрати часу та ресурсів, а також підвищити точність і прозорість усіх етапів обробки замовлень [6].

Сучасні вебплатформи електронної комерції повинні відповідати широкому спектру функціональних вимог, що впливають як з технічного прогресу, так і з очікувань користувачів. У контексті інтернет-магазинів косметики, як-от Lorena, особливо важливими є можливості ефективного управління товарами, забезпечення зручного пошуку та фільтрації продукції, а також високий рівень взаємодії з користувачем.

Ключовою вимогою є наявність повноцінного каталогу продукції, який дозволяє адміністратору додавати, редагувати та видаляти товари, вказувати категорії, опис, характеристики, наявність, ціну та знижки. Важливу роль відіграє інтерфейс пошуку, який має забезпечувати швидкий доступ до потрібних товарів із можливістю застосування різноманітних фільтрів, зокрема за брендом, ціновим діапазоном, типом шкіри, рейтингом та іншими параметрами [7].

Система автентифікації користувачів повинна підтримувати як традиційну авторизацію за email та паролем, так і соціальну авторизацію (наприклад, за допомогою Google або GitHub). При цьому передбачена рольова модель користувачів, яка включає як звичайних покупців, так і адміністраторів чи менеджерів системи.

Невід'ємною функціональністю є можливість додавання товарів до кошика з подальшим оформленням замовлення. При цьому система має забезпечувати динамічне оновлення вартості замовлення, коректну обробку кількості одиниць товару, а також просте та зрозуміле оформлення покупки. Після підтвердження замовлення клієнт має отримувати інформацію про нього на електронну пошту.

Питання оплати є критично важливим. Тому e-commerce рішення має інтегруватися з сучасними платіжними сервісами (наприклад, Stripe, PayPal), підтримувати оплату банківськими картками та забезпечувати безпечне збереження та передавання платіжної інформації згідно з міжнародними стандартами безпеки [8].

Крім того, необхідною є підтримка функціоналу доставки: користувач повинен мати можливість вибору типу доставки (безкоштовна, швидка) та отримувати трекінг-номер для відстеження свого замовлення. Усі ці дані зберігаються та відображаються у його особистому кабінеті.

Ще однією важливою складовою є адміністративна панель, яка дозволяє керувати товарами, замовленнями, користувачами, а також переглядати статистику, формувати аналітичні звіти та відстежувати динаміку продажів [9]. Для кращої взаємодії з продуктом бажано, щоб інтерфейс був адаптований до мобільних пристроїв і мав сучасний дизайн, який відповідає принципам UX/UI.

Окрім основних функцій, до сучасних вимог належать можливість формування списку бажань, рекомендації товарів на основі поведінки користувача, підтримка швидкого пошуку з підказками, використання аналітики для персоналізації контенту та інтеграція з зовнішніми сервісами для завантаження зображень.

Не менш важливим є питання безпеки. Усі запити до сервера повинні бути захищеними, а персональні дані – зашифрованими. Важливою функцією для адміністраторів є двофакторна автентифікація, яка додає додатковий рівень захисту [10].

Усі перелічені вимоги будуть враховані при створенні інтернет-магазину Lorena, що розроблятиметься із застосуванням сучасного технологічного стеку, який забезпечить масштабованість, продуктивність і високий рівень зручності як для користувачів, так і для адміністраторів.

1.1.3 Роль адміністративної панелі в управлінні інтернет-магазином

Адміністративна панель є ключовим елементом управління сучасним інтернет-магазином, виконуючи роль центрального хабу для контролю та оптимізації всіх бізнес-процесів [11]. Вона забезпечує інтегроване середовище для управління товарами, замовленнями, клієнтами, аналітикою та іншими критично важливими аспектами онлайн-торгівлі.

Зокрема, адмінпанель дозволяє ефективно керувати асортиментом продукції: додавати нові товари, редагувати їхні характеристики, відстежувати залишки на складі та оновлювати ціни. Це особливо важливо в галузі косметики, де асортимент часто змінюється, а точність інформації про продукти має вирішальне значення для покупців.

Крім того, через адміністративну панель здійснюється обробка замовлень: перегляд статусів, підтвердження оплат, організація доставки та обробка повернень. Вона також надає можливість аналізувати поведінку клієнтів, відстежувати ключові показники ефективності (KPI) та приймати обґрунтовані рішення щодо маркетингових стратегій і розвитку бізнесу.

Безпека є ще одним критичним аспектом адміністративної панелі. Вона забезпечує контроль доступу до конфіденційної інформації, дозволяючи лише авторизованим користувачам виконувати певні дії. Сучасні системи адміністрування підтримують багаторівневу систему ролей, що дозволяє розмежовувати права доступу відповідно до посадових обов'язків співробітників [12].

1.2 Аналітичний огляд існуючих інтернет-магазинів косметики

У контексті розробки інтернет-магазину косметики Lorena доцільно проаналізувати низку українських онлайн-магазинів косметики, які, хоча й не є гігантами ринку, демонструють ефективні стратегії ведення бізнесу та можуть слугувати джерелом натхнення для подібних проєктів.

Було проаналізовано такі інтернет-магазини косметики:

1. Kosmetychka.if.ua. Цей інтернет-магазин пропонує широкий асортимент косметичних засобів, включаючи макіяж, догляд за волоссям та тілом, а також аксесуари. Важливою особливістю є співпраця з перевіреними виробниками, що гарантує якість продукції. Крім того, сайт надає детальні описи товарів та якісні фотографії, що полегшує вибір для покупців.

2. **Marie Fresh Cosmetics.** Цей бренд спеціалізується на виробництві натуральної косметики з високою концентрацією активних компонентів. Продукція створена на основі європейських інгредієнтів, сертифікованих за стандартами Ecocert і Cosmos. Особливістю є розробка засобів лікарем-косметологом, що враховує потреби різних типів шкіри.

3. **TI'TUEL.** Український виробник косметики, який пропонує засоби для догляду за обличчям, тілом та нігтями. Продукція не позиціонується як професійна або органічна, але орієнтована на якісний догляд для різних типів шкіри. Сайт надає можливість придбати косметику безпосередньо від виробника, що може бути привабливим для споживачів, які шукають локальні бренди.

4. **PureShop.** Інтернет-магазин, що спеціалізується на продажу професійної косметики преміум-класу. Асортимент включає оригінальні засоби з високоякісними компонентами, що дозволяє досягти швидких результатів у догляді за шкірою. Магазин також пропонує консультації для підбору продукції відповідно до індивідуальних потреб клієнтів.

Порівняльний аналіз функціоналу та дизайну згаданих вище інтернет-магазинів косметики наведено у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз функціональних рішень та дизайну інтернет-магазинів косметики

Показник/Сайт	Kosmetychka.if.ua	Marie Fresh Cosmetics	TI'TUEL	PureShop
Каталог продукції	Широкий, багато категорій	Обмежений, лише власна продукція	Власна продукція середнього асортименту	Великий, преміум-сегмент
Система фільтрів	Базова (категорії, бренди)	Мінімальна	Базова	Розширена (тип шкіри, компоненти тощо)
Пошук по сайту	Так	Так	Так	Так
Особистий кабінет	Так	Так	Так	Так
Відгуки користувачів	Присутні	Присутні	Відсутні	Присутні

Продовження таблиці 1.1

Мобільна адаптація	Так, базова	Так, сучасна	Так, базова	Так, сучасна
Оплата онлайн	Так (банківська карта, Приват24)	Так (Wayforpay, LiqPay)	Так	Так
Інтеграція з соцмережами	Присутня (Facebook, Instagram)	Активна інтеграція	Присутня	Присутня
Дизайн сайту	Мінімалістичний, простий	Стильний, фірмовий	Простий	Сучасний, преміум-естетика
Мова сайту	Українська	Українська, англійська	Українська	Українська
Форма зв'язку / зворотного зв'язку	Так (форма + телефони)	Так (чат, форма, соцмережі)	Так (форма)	Так (форма, чат, месенджери)
Наявність блогу / порад	Відсутній	Присутній (блог, поради)	Відсутній	Частково (новини, опис продукції)

Аналіз наявних рішень на українському ринку косметичних інтернет-магазинів виявив як сильні сторони конкурентів, так і сфери для потенційного вдосконалення. Високим рівнем візуальної подачі та зручності користування відзначаються сайти Marie Fresh Cosmetics та PureShop, що мають сучасний дизайн, адаптивність під мобільні пристрої та чітку логіку взаємодії з клієнтом. Наявність блогу, фільтрації за інгредієнтами, типом шкіри, підтримка онлайн-оплати та якісна презентація продукції формують довіру до бренду.

Водночас, менш масштабні проєкти на кшталт Kosmetychka.if.ua та TITUEL мають спрощений функціонал і скромніший інтерфейс. Вони часто не використовують повний потенціал інтерактивних інструментів (відгуки, рекомендаційні системи, інтеграцію з соцмережами), а також не забезпечують поглибленої фільтрації або персоналізованого підходу до клієнта.

З точки зору UX/UI, переваги конкурентів полягають у послідовному стилі, візуальній ієрархії контенту та доступності інформації. Недоліки ж проявляються у відсутності багатомовності, обмеженому функціоналі особистого кабінету, слабкій взаємодії з користувачем після оформлення

замовлення, а також нестачі маркетингових інструментів, таких як e-mail розсилка, бонусна система або знижки для постійних клієнтів.

Отже, при створенні інтернет-магазину Lorena доцільно зосередити увагу не лише на основному функціоналі, а й на сучасному дизайні, мобільній адаптивності, розширених фільтрах, активному блог-розділі та персоналізованій взаємодії з клієнтом, щоб сформувати конкурентну перевагу навіть на фоні більш ресурсних проєктів.

1.3 Постановка завдання розробки інтернет-магазину косметики з адміністративною панеллю

1.3.1 Область застосування та призначення розробки інтернет-магазину Lorena

Інтернет-магазин косметики Lorena призначений для користувачів, які цінують якісну доглядову та декоративну косметику та бажають зручно здійснювати покупки онлайн. Цільова аудиторія проєкту – це переважно жінки віком від 18 до 45 років, які активно користуються інтернетом, віддають перевагу сучасному дизайну інтерфейсу, швидкому пошуку продуктів, доступності інформації про склад і рекомендації щодо використання, а також можливості безпечної та зручної оплати. Додатково платформа буде корисною для чоловіків, які шукають косметичні подарунки, та для молоді, яка цікавиться трендами у сфері догляду за собою.

Мета розробки полягає у створенні повноцінного e-commerce рішення, яке забезпечить ефективну комунікацію між брендом та покупцем, автоматизацію процесів замовлення, оплати та обробки товарів, а також дозволить адміністрації зручно керувати контентом, товарами, замовленнями та користувачами через адміністративну панель.

Інтеграція адміністративної панелі є критично важливою частиною проєкту, адже вона дозволяє власникам бізнесу оперативно реагувати на зміни

в асортименті, відстежувати продажі, аналізувати поведінку користувачів, оновлювати зображення та описи товарів, а також підтримувати зворотний зв'язок із клієнтами. Це забезпечує не лише ефективне функціонування платформи, але й масштабованість у майбутньому.

1.3.2 Вимоги до функціоналу сайту інтернет-магазину Lorena

Інтернет-магазин косметики Lorena має відповідати сучасним вимогам до зручності, безпеки та ефективності взаємодії з користувачами. Основні функціональні вимоги поділяються на дві категорії: фронтенд-функціональність для користувачів та адміністративна функціональність для керування платформою.

З боку відвідувачів і покупців сайт повинен забезпечувати такі можливості:

- перегляд товарів з деталізацією (назва, опис, ціна, зображення, відгуки);
- фільтрація і сортування продукції за категоріями, брендами або ціною;
- реєстрація та авторизація користувачів через безпечну систему (Clerk);
- можливість додавання товарів у кошик та його редагування;
- проведення безпечної онлайн-оплати через Stripe;
- перегляд історії замовлень, відстеження статусу доставки;
- можливість додавання товарів до списку бажань (wishlist);
- підтримка адаптивного дизайну для зручного перегляду з мобільних пристроїв.

З боку адміністратора сайту важливо забезпечити функціональність для ефективного управління контентом і замовленнями:

- авторизація адміністратора та доступ до захищеної адмін-панелі;

- додавання, редагування та видалення товарів і колекцій через зручні форми з валідацією;
- управління замовленнями, перегляд деталей покупок, зміна статусів;
- аналітика продажів, кількості клієнтів, найбільш популярних товарів (Recharts);
- керування зображеннями товарів через інтеграцію з Cloudinary;
- пошук по товарах, замовленнях та клієнтах;
- створення і редагування колекцій для групування продукції.

Всі ці функції мають бути реалізовані з використанням сучасних підходів до розробки вебзастосунків, що забезпечить як комфорт користувача, так і зручність адміністрування, а також дозволить масштабувати платформу в майбутньому.

1.3.3 Стадії та етапи розробки інтернет-магазину

Розробка інтернет-магазину косметики Lorena здійснювалася із дотриманням поетапного підходу, заснованого на каскадній моделі життєвого циклу програмного забезпечення (Waterfall model). Цей підхід передбачає чітке послідовне виконання етапів, де завершення одного є обов'язковою передумовою для переходу до наступного [13]. Така модель була обрана з огляду на чітко визначені вимоги до системи, структурованість процесу розробки, а також зручність контролю над етапами створення функціонального продукту.

Першим кроком є аналіз предметної області та постановка завдання. У ході цього етапу було зібрано вимоги до майбутнього застосунку, окреслено цільову аудиторію, а також визначено функціональні потреби як клієнтів магазину, так і адміністраторів. Окрему увагу приділено вивченню ринку косметичних e-commerce рішень в Україні та закордоном, що дозволило сформулювати обґрунтоване технічне завдання.

На основі зібраних вимог було здійснено вибір технологій, які найбільш ефективно відповідали цілям проєкту. Архітектура інтернет-магазину будувалася з використанням Next.js 14 для сучасного веб-інтерфейсу, MongoDB як гнучкої NoSQL бази даних, Stripe для онлайн-платежів, Clerk для автентифікації та керування користувачами, а також Tailwind CSS і Shadcn UI для побудови адаптивного інтерфейсу. Паралельно проєктувалася структура бази даних із урахуванням зв'язків між сутностями, таких як товари, колекції, користувачі, замовлення, кошик, тощо.

На етапі реалізації було створено інтерфейс користувача з усіма ключовими можливостями: реєстрація та авторизація, перегляд колекцій та товарів, додавання до кошика, здійснення оплати, перегляд історії замовлень. Для цього застосовувалися як серверні компоненти Next.js, так і клієнтські взаємодії через React-Hook-Form. Паралельно з цим тривала розробка адміністративної панелі, яка включає керування контентом, перегляд замовлень, додавання нових продуктів і колекцій, аналітичні графіки та систему пошуку.

Наступним етапом стала інтеграція зовнішніх сервісів, включаючи Cloudinary для обробки зображень, Stripe Webhooks для обробки змін статусів замовлень у реальному часі, а також Clerk як центральний інструмент для забезпечення безпечного входу в систему та керування ролями користувачів.

Після цього проєкт перейшов до стадії тестування, яка охоплювала перевірку форми на коректність валідації, поведінку кошика, обробку платежів, а також адаптивність інтерфейсу на різних пристроях. Також було перевірено стійкість системи до навмисно некоректних запитів.

Фінальною фазою стало розгортання проєкту на платформі Vercel з підключенням до MongoDB Atlas, налаштуванням доменного імені та конфігурацією змінних середовища для Stripe і Clerk. Цей етап включав оптимізацію продуктивності й підготовку до майбутньої підтримки.

Таким чином, поетапна реалізація за класичною каскадною моделлю забезпечила прозорість, передбачуваність і контроль над усіма аспектами

створення інтернет-магазину. Завдяки такій структурі вдалося досягти високого рівня якості продукту й забезпечити його відповідність сучасним очікуванням користувачів та бізнес-цілям.

1.4 Вибір та обґрунтування використаних технологій

1.4.1 Обґрунтування вибору фреймворку Next.js

Фреймворк Next.js було обрано як основну технологічну платформу для реалізації інтернет-магазину Lorena з огляду на його гнучкість, продуктивність і сучасний підхід до створення повноцінних вебдодатків. На момент розробки використовувалась версія Next.js 14, яка включає App Router — інноваційний підхід до маршрутизації, що дозволяє більш ефективно організовувати компонування сторінок, розділення логіки серверних і клієнтських частин, а також повторне використання інтерфейсних елементів [14].

Однією з головних переваг Next.js є гібридна рендеринг-модель, яка дозволяє комбінувати серверну генерацію сторінок (Server-Side Rendering, SSR) з попередньою генерацією на етапі збірки (Static Site Generation, SSG) [15]. Завдяки цьому інтернет-магазин може забезпечити як високу швидкість завантаження сторінок, так і актуальність даних, особливо для товарів, які часто змінюються або залежать від користувача. Це критично важливо для e-commerce-проектів, де час відгуку сайту безпосередньо впливає на конверсію.

Ще одним вагомим фактором є повна інтеграція з React — сучасною бібліотекою для побудови інтерфейсів користувача. Оскільки Next.js є надбудовою над React, це дає змогу застосовувати весь інструментарій екосистеми React, включаючи хуки, контекст, кастомні компоненти та формову логіку. Крім того, Next.js підтримує TypeScript "з коробки", що сприяє створенню більш надійного коду з чітко визначеними типами [16].

Також важливою є зручна інтеграція з хмарними платформами (зокрема, Vercel, розробником якої є сама команда Next.js), що значно спрощує процес

розгортання та CI/CD (Continuous Integration / Continuous Delivery) [17]. Це дозволяє зосередитися на розробці функціоналу, не витрачаючи зайвих ресурсів на інфраструктурні задачі. Методологія CI/CD дозволяє забезпечити якість коду, автоматизоване тестування і швидке постачання оновлень користувачам. Ця методологія включає безперервну інтеграцію змін у репозиторій із одночасним тестуванням, а також автоматизоване постачання стабільних версій програмного продукту в продакшн-середовище [18].

Завдяки архітектурі App Router, яка базується на файловій структурі з підтримкою розділення за layout-компонентами, забезпечується краща модульність коду. Це полегшує підтримку й масштабування проєкту – як у частині користувацького інтерфейсу, так і в адміністративній панелі. Крім того, наявність middleware дає змогу реалізовувати логіку контролю доступу до сторінок на серверному рівні, що особливо актуально для захищених маршрутів у адмінці [19].

У результаті, Next.js 14 став оптимальним вибором для розробки сучасного, масштабованого, продуктивного інтернет-магазину з інтегрованою панеллю керування, відповідаючи всім вимогам до сучасного e-commerce-рішення як з боку розробника, так і з боку користувача.

1.4.2 Використання Tailwind CSS для UI

Перше враження надзвичайно важливе – близько 94% користувачів формують його на основі зовнішнього вигляду сайту. Іншими словами, те, як візуально оформлений вебресурс, здебільшого визначає, яким буде перше враження від нього. Йдеться не лише про естетичну привабливість. Для 46,1% відвідувачів саме дизайн є основним критерієм оцінки надійності компанії чи бренду, що стоїть за сайтом [20].

У процесі розробки інтерфейсу інтернет-магазину Lorena було обрано Tailwind CSS як основний інструмент для стилізації користувацького інтерфейсу. Tailwind CSS є утилітарним CSS-фреймворком, який надає набір

готових класів для безпосереднього застосування до HTML-елементів, що дозволяє створювати адаптивні, модульні та інтуїтивно зрозумілі інтерфейси без необхідності написання великої кількості власного CSS-коду [21].

Однією з переваг Tailwind CSS є підтримка мобільної адаптивності та можливість створення сучасного дизайну, дотримуючись найкращих практик UI/UX. Інформація щодо аналізу дизайнерських компетенцій програміста у контексті використання Tailwind CSS представлена у тезах про відповідне дослідження [22], що були опубліковані у матеріалах XIII Міжнародної студентської науково-технічної конференції «Природничі та гуманітарні науки. Актуальні питання» та подані в додатку А кваліфікаційної роботи.

У поєднанні з офіційними плагінами, такими як `@tailwindcss/forms` та `@tailwindcss/typography`, значно спрощується робота з формами та типографікою, забезпечуючи швидку та ефективну розробку інтерфейсу.

Tailwind CSS також відомий своєю високою продуктивністю. Завдяки механізму `tree-shaking` та оптимізації через PostCSS і PurgeCSS, у фінальну збірку потрапляє лише використаний у проєкті CSS, що суттєво зменшує розмір файлів і покращує швидкість завантаження сторінок [23].

Таким чином, використання Tailwind CSS забезпечило швидку розробку інтерфейсу користувача, відповідність сучасним стандартам дизайну, зручну підтримку адаптивності та оптимальну продуктивність застосунку.

1.4.3 Вибір MongoDB в якості бази даних для інтернет-магазину

Для зберігання даних у проєкті інтернет-магазину косметики Lorena було обрано MongoDB – документно-орієнтовану нереляційну базу даних (NoSQL), яка ідеально підходить для гнучких, динамічних та масштабованих вебдодатків. Однією з основних причин вибору MongoDB стала її здатність працювати зі структурованими документами у форматі JSON-подібних BSON-об'єктів [24], що добре узгоджується зі структурою об'єктів у JavaScript/TypeScript, який використовується на фронтенді та бекенді.

MongoDB дозволяє створювати складні вкладені структури даних, що особливо зручно для e-commerce-додатків, у яких є товари з характеристиками, зображеннями, категоріями, відгуками тощо. Документно-орієнтована модель забезпечує гнучке зберігання інформації без жорстких схем, що дає змогу швидко змінювати або доповнювати структуру бази в процесі розробки [25].

Ще однією перевагою MongoDB є її масштабованість та висока продуктивність при читанні й записі даних. Це дозволяє обслуговувати одночасно велику кількість користувачів, що є критичним для інтернет-магазинів із високим трафіком, особливо під час пікових періодів (акцій, сезонних розпродажів тощо).

MongoDB також підтримує побудову агрегованих запитів, що дає змогу обробляти статистику замовлень, переглядів, користувацьких дій без необхідності додаткової логіки на стороні сервера. У поєднанні з ORM-бібліотекою Mongoose або з Prisma (через MongoDB adapter), розробник отримує потужний інструмент для ефективної взаємодії з базою з перевіркою типів, валідацією, middleware та хуками [26].

Крім того, MongoDB має хмарну платформу MongoDB Atlas, яка дозволяє швидко розгорнути продуктивну базу даних, забезпечити резервне копіювання, реплікацію, моніторинг та автоматичне масштабування – усе це значно спрощує адміністрування та підвищує надійність системи [27].

Таким чином, MongoDB – це гнучке, масштабоване й сучасне рішення, яке забезпечує швидкий розвиток проєкту, високу продуктивність і зручність у розробці, що робить його ідеальним вибором для реалізації інтернет-магазину з адміністративною панеллю.

1.4.4 Інтеграція з платіжною системою Stripe

Для реалізації платіжної системи в інтернет-магазині Lorena було обрано сервіс Stripe, який є одним із найпопулярніших рішень для онлайн-оплати у світі. Stripe забезпечує розробників потужними API для інтеграції платежів,

підтримує різноманітні валюти та методи оплати, включаючи банківські картки, Google Pay, Apple Pay та інші локальні рішення [28].

Однією з ключових причин вибору Stripe є його простота інтеграції з фреймворком Next.js, підтримка Webhooks для обробки подій, пов'язаних з транзакціями, і сумісність з серверними функціями, реалізованими на платформі Vercel. Stripe також пропонує захищене та відповідне стандартам PCI DSS зберігання платіжних даних, що дозволяє зосередитися на бізнес-логіці без необхідності вирішувати складні завдання з безпеки [29].

Система також дозволяє реалізовувати динамічні ціни, кошики, промокоди, а також інтегрувати Stripe Dashboard для моніторингу транзакцій та управління підписками чи одноразовими покупками. Stripe чудово масштабується – як з технічної, так і з бізнесової точки зору – що є важливим для інтернет-магазинів, які планують ріст та розвиток.

У рамках проєкту Lorena Stripe було інтегровано на етапі оформлення замовлення, після чого здійснюється перенаправлення користувача на сторінку безпечної оплати. Після завершення транзакції система отримує повідомлення через Webhook і створює запис про замовлення в базі даних MongoDB.

1.4.5 Обґрунтування використання Clerk для автентифікації

Для реалізації системи автентифікації в інтернет-магазині Lorena було обрано сервіс Clerk, який є сучасним рішенням для керування користувачами, що спеціалізується на інтеграції з JavaScript-фреймворками, зокрема з Next.js. Clerk дозволяє швидко впровадити реєстрацію, вхід, керування сесіями, авторизацію, а також функції профілю користувача без необхідності розробки таких компонентів з нуля.

Основною перевагою використання Clerk є його повна сумісність з новим App Router API Next.js 13/14, що дозволяє зручно організувати захищені маршрути для адміністративної панелі або обмежити доступ до особистих даних користувачів. Clerk також підтримує соціальну автентифікацію (через

Google, GitHub тощо), а також двофакторну аутентифікацію (2FA), що підвищує рівень безпеки застосунку.

Крім зручного API, сервіс пропонує готові компоненти UI (наприклад, SignIn, SignUp, UserButton), які легко стилізуються і інтегруються у вже наявний інтерфейс з використанням Tailwind CSS. Ще однією важливою функціональністю є Clerk Dashboard, який дає змогу адміністратору або розробнику керувати користувачами, перевіряти сесії, налаштовувати політики доступу та сповіщення.

Таким чином, Clerk значно спрощує реалізацію системи автентифікації у порівнянні з власноручним налаштуванням рішень на базі JWT або OAuth, знижуючи ризики помилок у безпеці та прискорюючи розробку.

1.5 Висновок до першого розділу

У першому розділі було проведено всебічний аналіз предметної області розробки інтернет-магазину косметики Lorena. Зокрема, досліджено сучасні тенденції у сфері електронної комерції косметичних товарів, що свідчать про активне зростання цього сегмента ринку, зокрема в Україні. Проаналізовано функціональні вимоги до сучасних e-commerce рішень, серед яких ключовими є зручність користування, безпека, мобільна адаптивність, персоналізація контенту, інтеграція з платіжними системами та ефективне управління товарними позиціями.

Окрема увага була приділена адміністративній панелі, яка є критично важливою складовою для повноцінного функціонування будь-якого інтернет-магазину, оскільки забезпечує гнучке управління асортиментом, обробку замовлень, аналітику продажів і моніторинг активності користувачів.

У межах аналітичного огляду було розглянуто низку актуальних рішень на ринку української косметики в інтернеті, проведено їх порівняння за критеріями функціоналу та дизайну. Це дозволило виявити характерні риси

конкурентного середовища, сильні та слабкі сторони існуючих рішень і сформулювати власні вимоги до проєкту Lorena.

На основі поставлених задач визначено призначення майбутнього застосунку та охарактеризовано функціональні вимоги до нього. Також описано модель життєвого циклу проєкту, що передбачає етапи планування, проєктування, реалізації, тестування та впровадження.

На завершення розділу обґрунтовано вибір технологій, що будуть застосовуватися в розробці. Зокрема, фреймворк Next.js було обрано за його універсальність і підтримку SSR/CSR, Tailwind CSS – за швидкість розробки інтерфейсів, MongoDB – як гнучке документно-орієнтоване сховище, Stripe – як надійне платіжне рішення, а Clerk – як зручний та безпечний сервіс автентифікації.

Таким чином, проведений аналіз заклав основу для реалізації ефективного, сучасного та зручного інтернет-магазину косметики, здатного відповідати вимогам ринку та очікуванням користувачів.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ LORENA

2.1 Технологічні та архітектурні основи побудови вебзастосунку Lorena

2.1.1 Загальний опис архітектури інтернет-магазину

Інтернет-магазин Lorena створено за принципами клієнт-серверної архітектури, де чітко розділено обов'язки між клієнтською та серверною частинами. Такий підхід дає змогу забезпечити масштабованість, гнучкість та ефективне розгортання нових функцій. Клієнтська частина працює у браузері користувача і відповідає за інтерфейс та взаємодію з користувачем, тоді як серверна частина обробляє бізнес-логіку, взаємодіє з базою даних і зовнішніми сервісами [30].

Фреймворк Next.js 14, що лежить в основі платформи Lorena, дозволяє поєднувати клієнтські і серверні можливості в одному середовищі. Завдяки App Router у Next.js реалізовано підтримку серверного рендерингу, що сприяє швидкому завантаженню сторінок і кращій індексації в пошукових системах [31]. Це особливо важливо для інтернет-магазину, де важливу роль відіграє SEO-оптимізація.

На клієнтській стороні користувач взаємодіє з інтерфейсом, який розроблений із застосуванням Tailwind CSS та бібліотеки Shadcn UI. Всі дії користувача, такі як додавання товарів до кошика, реєстрація, або оформлення замовлення, ініціюють запити до API, які обробляються на сервері. Запити надсилаються через HTTP-з'єднання у форматі JSON, і сервер у відповідь надає лише необхідні дані, зменшуючи навантаження на мережу.

На серверній стороні обробляються усі запити користувачів, здійснюється доступ до бази даних MongoDB, де зберігається інформація про товари, категорії, замовлення та клієнтів. Сервер також взаємодіє з платіжною

системою Stripe для обробки транзакцій та з Cloudinary для завантаження й оптимізації зображень. Таким чином, сервер відіграє роль посередника між користувачем і всіма критичними зовнішніми сервісами та базами даних.

Адміністративна панель реалізована як частина вебзастосунку, з обмеженим доступом на основі авторизації через платформу Clerk. Вона дозволяє адміністратору додавати нові товари, переглядати замовлення, оновлювати інформацію про клієнтів та аналізувати основні показники діяльності магазину. Цей функціонал повністю взаємодіє з тим самим сервером і базою даних, що і публічна частина магазину, завдяки чому забезпечується цілісність системи.

Весь застосунок розгортається на хмарній платформі Vercel, що забезпечує автоматичне масштабування, безперервну інтеграцію та швидке оновлення змін. Така архітектура дозволяє команді розробників швидко вносити поліпшення, тестувати нові функції та підтримувати високу якість обслуговування користувачів.

У підсумку, архітектура Lorena є прикладом сучасного підходу до побудови вебзастосунків: вона поєднує зручний клієнтський інтерфейс із потужною серверною логікою, забезпечуючи ефективну і стабільну роботу інтернет-магазину як з боку покупців, так і з боку адміністрації.

2.1.2 Структура файлової системи проекту

Інтернет-магазин Lorena складається з двох незалежних частин: клієнтської (користувачької) та адміністративної. Вони розміщені у двох окремих папках – `lorena_store` та `lorena_admin`. Така сегментація дозволяє чітко розмежувати функціонал для кінцевих користувачів і для адміністраторів платформи, забезпечуючи кращу безпеку, масштабованість та зручність розробки.

У структурі папки `lorena_store` зосереджено весь функціонал, що доступний покупцям. Вона містить директорію `app`, яка побудована на основі

App Router фреймворку Next.js і організована за логікою route-based структури. Всередині неї присутні окремі папки для авторизації ((auth)), корзини (cart), перегляду колекцій та товарів (collections, products), оформлення замовлень (orders), пошуку (search) та списку бажаного (wishlist). Крім цього, в app/api реалізовані серверні API-обробники, наприклад, для взаємодії зі списком бажаного.

Допоміжні бібліотеки та логіка винесені у папку lib, де знаходяться підкаталоги actions, hooks, models і providers, які відповідають за обробку даних, повторно використовувані гачки, опис моделей і глобальні провайдери стану або контекстів. Компонентна структура реалізована в окремій директорії components, що полегшує повторне використання UI-елементів. Папка public містить ресурси, доступні напряму через URL, включно з favicon та іншими файлами.

Папка .next містить скомпільовані файли після компіляції, а також кешовані ресурси, необхідні для швидкого запуску і рендерингу. Вона автоматично генерується під час розгортання або розробки, і, як правило, не редагується вручну.

На рисунку 2.1 наведено діаграму, що візуалізує файлову структуру клієнтської частини проекту інтернет-магазину Lorena.

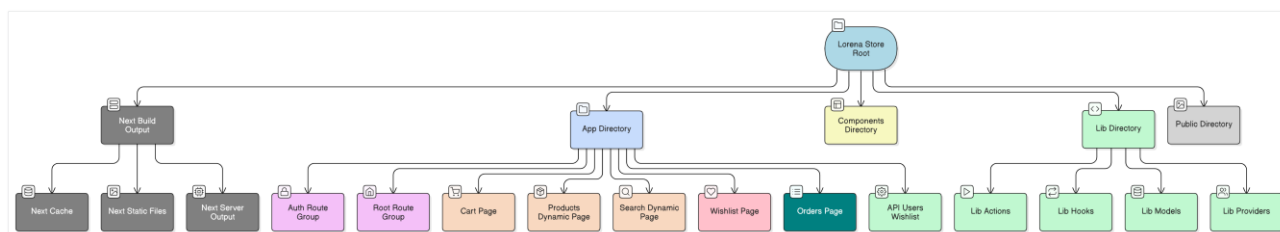


Рисунок 2.1 – Файлова структура клієнтської частини проекту

Окрема папка lorena_admin містить адміністративну панель, яка також реалізована на основі Next.js. У директорії app розташовані маршрути, що стосуються панелі керування замовленнями, клієнтами, товарами та колекціями. Кожен із CRUD-функціоналів [32] організовано в окремих

вкладених шляхах, таких як `products/new`, `collections/[collectionId]` або `orders/[orderId]`. Тут також реалізовані окремі API-обробники, зокрема для роботи з товарами, клієнтами, пошуком, оформленням замовлень та `webhook`-інтеграцією.

Компоненти в `lorena_admin` згруповані за типами функціональності: для відображення товарів, замовлень, клієнтів та колекцій. Також у структурі є спеціальні папки `layout` та `custom ui`, що відповідають за базове оформлення інтерфейсу і кастомізовані елементи управління.

Крім того, в адміністративній частині у папці `public` зберігаються медіафайли для товарів і колекцій, зокрема фотографії косметичних засобів, які систематизовано за типами: `lipstick`, `spf`, `handcream` тощо. Це дозволяє організовано керувати мультимедійними ресурсами та забезпечує швидкий доступ до зображень через CDN.

На рисунку 2.2 наведено діаграму, що візуалізує файлову структуру адміністративної частини проекту інтернет-магазину Lorena.

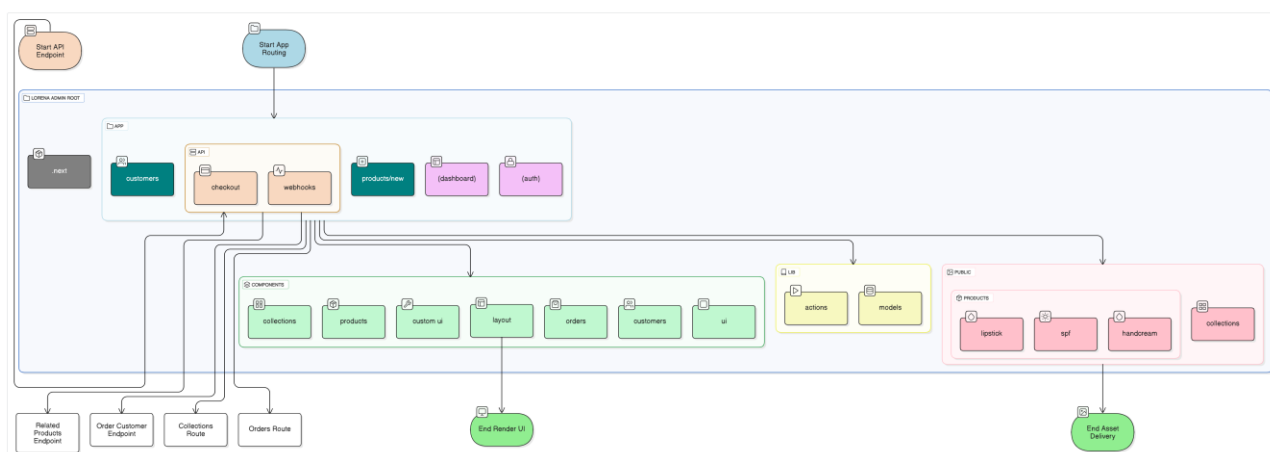


Рисунок 2.2 – Файлова структура адміністративної частини проекту

Таким чином, структура файлової системи інтернет-магазину Lorena є чітко організованою, логічно поділеною між клієнтським і адміністративним інтерфейсами, і повністю відповідає принципам масштабованої та підтримуваної архітектури сучасних вебзастосунків.

2.1.3 Бібліотеки, використані для розробки клієнтської частини проєкту

У процесі створення веб-додатків важливу роль відіграють сторонні бібліотеки готові модулі коду, які дозволяють розробникам не створювати функціональність «з нуля», а використовувати перевірені рішення для типових задач: від рендерингу інтерфейсу та стилізації до керування станом і взаємодії з API [33]. Це значно пришвидшує процес розробки, підвищує стабільність системи та полегшує підтримку коду.

Усі підключені бібліотеки зберігаються у спеціальному конфігураційному файлі `package.json`, що є складовою кожного проєкту на базі Node.js. Цей файл містить як список робочих залежностей (`dependencies`), необхідних для функціонування застосунку в продакшені, так і перелік інструментів для розробки (`devDependencies`) (див. рисунок 2.3).

```

11  "dependencies": {
12    "@clerk/nextjs": "^4.29.7",
13    "@stripe/stripe-js": "^3.0.6",
14    "lucide-react": "^0.341.0",
15    "mongoose": "^8.2.0",
16    "next": "14.1.0",
17    "react": "^18",
18    "react-dom": "^18",
19    "react-hot-toast": "^2.4.1",
20    "stripe": "^14.18.0",
21    "tailwind-scrollbar-hide": "^1.1.7",
22    "zustand": "^4.5.1"
23  },
24  "devDependencies": {
25    "@types/node": "^20",
26    "@types/react": "^18",
27    "@types/react-dom": "^18",
28    "autoprefixer": "^10.0.1",
29    "postcss": "^8",
30    "tailwindcss": "^3.3.0",
31    "typescript": "^5"
32  }
33 }

```

Рисунок 2.3 – Фрагмент файлу `package.json` з переліком бібліотек, використаних у клієнтській частині проєкту

Клієнтська частина інтернет-магазину Lorena реалізована з використанням фреймворку Next.js (версія 14.1.0), що забезпечує серверний рендеринг і зручну маршрутизацію. Основою інтерфейсу слугує React (v18)

одна з найпопулярніших бібліотек для побудови компонентного UI. Для гнучкої стилізації використано Tailwind CSS, доповнений плагіном `tailwind-scrollbar-hide` для приховування смуг прокрутки.

Інтеграція з платіжною системою Stripe здійснюється за допомогою бібліотек `@stripe/stripe-js` і `stripe`, які дають змогу обробляти платежі безпечно і зручно. Для аутентифікації користувачів використано платформу Clerk, підключену через офіційний SDK `@clerk/nextjs`.

Глобальний стан додатка, зокрема функціонал кошика, реалізовано за допомогою Zustand – простої та ефективної бібліотеки для керування станом [34]. Робота з базою даних MongoDB здійснюється через бібліотеку Mongoose (v8.2.0), що забезпечує зручну роботу з моделями та схемами.

Візуальне оформлення інтерфейсу доповнюється за допомогою SVG-ікон із бібліотеки `lucide-react`, а система сповіщень реалізована через `react-hot-toast`, яка дозволяє легко відображати інформативні повідомлення.

Під час розробки застосовувалась мова TypeScript (v5), яка забезпечує типізацію коду й запобігає поширеним помилкам. Для налаштування стилізації також використовувались PostCSS і Autoprefixer, які автоматично додають необхідні вендорні префікси до CSS-коду. Для типізації середовища були додані відповідні пакети `@types` для React і Node.js.

Завдяки такому комплексу бібліотек клієнтська частина проєкту є швидкою, масштабованою, візуально привабливою та має повний набір функцій, необхідних для повноцінного інтернет-магазину.

2.1.4 Бібліотеки, використані для розробки адміністративної частини проєкту

Адміністративна частина проєкту Lorena розроблена з використанням сучасного стеку інструментів, що забезпечують зручність створення інтерфейсу, обробки форм, валідації, відображення статистики та взаємодії з базою даних і сервісами. Основу UI складають бібліотеки Radix UI та Tailwind

CSS, а для керування формами використано react-hook-form у зв'язці з zod для валідації. Виведення графіків реалізовано за допомогою recharts, а для роботи з Cloudinary бібліотека next-cloudinary.

Інші інструменти забезпечують гнучке стилізування (clsx, tailwind-merge), керування таблицями (@tanstack/react-table), обробку дат (date-fns) та сповіщення (react-hot-toast). Інтеграція зі Stripe та Clerk також залишається ключовою частиною архітектури.

Фрагмент файлу package.json, що містить повний перелік підключених бібліотек, зображено на рисунку 2.4.

```

11  "dependencies": {
12    "@clerk/nextjs": "^4.29.7",
13    "@hookform/resolvers": "^3.3.4",
14    "@radix-ui/react-alert-dialog": "^1.0.5",
15    "@radix-ui/react-dialog": "^1.0.5",
16    "@radix-ui/react-label": "^2.0.2",
17    "@radix-ui/react-separator": "^1.0.3",
18    "@radix-ui/react-slot": "^1.0.2",
19    "@tanstack/react-table": "^8.12.0",
20    "class-variance-authority": "^0.7.0",
21    "clsx": "^2.1.0",
22    "cmdk": "^0.2.1",
23    "date-fns": "^3.3.1",
24    "lucide-react": "^0.335.0",
25    "mongoose": "^8.1.3",
26    "next": "14.1.0",
27    "next-cloudinary": "^5.20.0",
28    "react": "^18",
29    "react-dom": "^18",
30    "react-hook-form": "^7.50.1",
31    "react-hot-toast": "^2.4.1",
32    "recharts": "^2.12.2",
33    "stripe": "^14.19.0",
34    "tailwind-merge": "^2.2.1",
35    "tailwindcss-animate": "^1.0.7",
36    "zod": "^3.22.4"
37  },
38  "devDependencies": {
39    "@types/node": "^20",
40    "@types/react": "^18",
41    "@types/react-dom": "^18",
42    "autoprefixer": "^10.0.1",
43    "eslint": "^8",
44    "eslint-config-next": "14.1.0",
45    "postcss": "^8",
46    "tailwindcss": "^3.3.0",
47    "typescript": "^5"
48  }

```

Рисунок 2.4 – Фрагмент файлу package.json з переліком бібліотек, використаних у адміністративній частині проєкту

Уся ця інфраструктура забезпечує зручну і масштабовану адміністративну панель для керування товарами, замовленнями та користувачами.

2.2 Моделювання бази даних

Для візуалізації логічної структури бази даних, а також відображення основних сутностей і зв'язків між ними, була побудована ER-діаграма (Diagram Entity-Relationship). Така діаграма дозволяє наочно представити, які об'єкти містяться в базі даних, які поля кожна сутність зберігає, та як саме ці сутності взаємодіють між собою [35]. ER-діаграму розробленої БД інтернет-магазину Lorena наведено на рисунку 2.5.

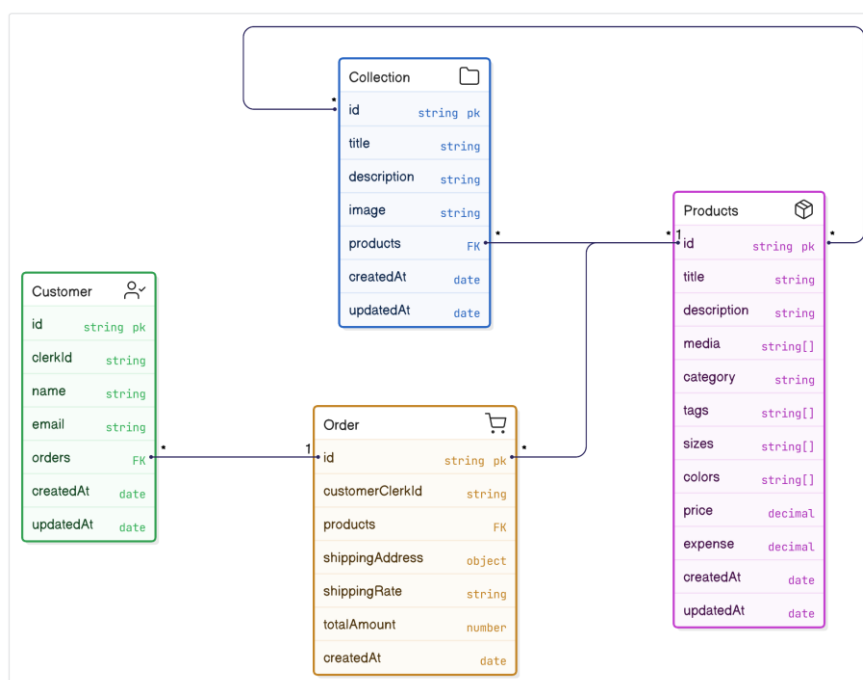


Рисунок 2.5 – ER-діаграма БД інтернет-магазину Lorena

Оскільки інтернет-магазин складається із з двох незалежних частин: клієнтської (користувачької) та адміністративної, тому було створено 2 бази даних.

У базі даних lorena_admin зосереджено роботу з даними, що стосуються товарного каталогу, замовлень і клієнтів. Основу структури становлять моделі товару, добірки, замовлення та клієнта. Товари містять усі необхідні атрибути (назва, опис, категорія, ціна, зображення тощо) та можуть бути включені до однієї або кількох добірок, що полегшує управління асортиментом. Замовлення

пов'язані з користувачами завдяки зовнішньому ідентифікатору, отриманому через сервіс авторизації, і містять детальну інформацію про вибрані товари, адресу доставки та суму. Кожен клієнт, у свою чергу, має доступ до списку замовлень, що автоматично формуються при покупці.

У базі даних `lorena_store` реалізовано збереження інформації про користувачів магазину. Колекція користувачів містить унікальний ідентифікатор (`clerkId`), список бажаних товарів (`wishlist`) та часові мітки створення й оновлення запису.

Побудована ER-діаграма значно спрощує розуміння внутрішньої логіки даних, полегшує підтримку та розвиток проєкту, а також стала основою для перевірки коректності реалізованих моделей.

Для зберігання даних у інтернет-магазині "Lorena" використано документоорієнтовану базу даних MongoDB у поєднанні з бібліотекою Mongoose, яка забезпечує зручну роботу з моделями, схемами та взаємозв'язками даних. У межах проєкту створено дві окремі бази даних: `lorena_store` для клієнтської частини та `lorena_admin` для адміністративної.

Взаємозв'язки між сутностями реалізовано за допомогою ідентифікаторів типу `ObjectId`, що дозволяє Mongoose підвантажувати пов'язані документи за допомогою функції `populate()`. Це створює гнучку й масштабовану структуру даних, що добре підходить для складного багатокomпонентного застосунку, як-от Lorena.

Модель користувача `User`, яка зберігається у базі даних `lorena_store`, представлена у Додатку Б, лістинг Б.1. Моделі `Collection`, `Customer`, `Order` і `Product`, які зберігаються у базі даних `lorena_admin`, наведено відповідно у лістингах Б.2–Б.5.

Вибір Mongoose як обгортки над MongoDB зумовлений її зручністю для роботи з TypeScript/JavaScript-проєктами, можливістю декларативного визначення схем, валідації даних і гнучким механізмом встановлення зв'язків між моделями [36]. Mongoose значно полегшує роботу з базою даних, спрощує синтаксис CRUD-операцій і зменшує кількість помилок при взаємодії з даними.

Завдяки цьому розробка серверної частини стала більш структурованою та передбачуваною.

2.3 Реалізація функціональності клієнтської частини інтернет-магазину

2.3.1 Реєстрація користувачів та їх авторизація у системі

У клієнтській частині інтернет-магазину Lorena реалізовано функціонал реєстрації та авторизації користувачів за допомогою зовнішнього сервісу автентифікації – Clerk. Цей сервіс забезпечує безпечне управління користувачами, спрощує реалізацію входу, реєстрації, відновлення пароля та інших пов'язаних функцій.

Clerk інтегровано у проєкт за допомогою бібліотеки `@clerk/nextjs`. Вона забезпечує готові компоненти та хук-інтерфейси, які легко вбудовуються у застосунок, створений на базі фреймворку Next.js. Для зберігання додаткової інформації про користувачів (наприклад, список бажаних товарів) у базі даних MongoDB використовується окрема колекція User, створена за допомогою бібліотеки Mongoose.

Після реєстрації користувача Clerk генерує унікальний ідентифікатор `clerkId`, який зберігається у колекції User та використовується для пов'язування даних у системі.

Таким чином, реєстрація та вхід у систему реалізовані без необхідності розробки власного механізму автентифікації, що дозволяє зосередитися на бізнес-логіці інтернет-магазину, не жертвуючи при цьому безпекою або зручністю користувача.

Доступ до головної сторінки інтернет-магазину надається всім користувачам, незалежно від того, увійшли вони до системи чи ні. Проте у разі відсутності автентифікації функціональні можливості обмежені: користувач може лише переглядати наявні колекції та товари. Доступ до персоналізованих

функцій, зокрема кошика та списку бажань, надається лише після успішного входу в обліковий запис.

Детальніше розглянемо інтерфейс та функціональні можливості сторінки реєстрації нового користувача, що наведений на рисунку 2.6. Для створення облікового запису необхідно заповнити форму, яка включає обов’язкові поля: ім’я, прізвище, адресу електронної пошти та пароль. Окрім цього, на сторінці передбачено можливість авторизації через обліковий запис Google. У такому випадку основні дані користувача автоматично імпортуються із відповідного профілю.

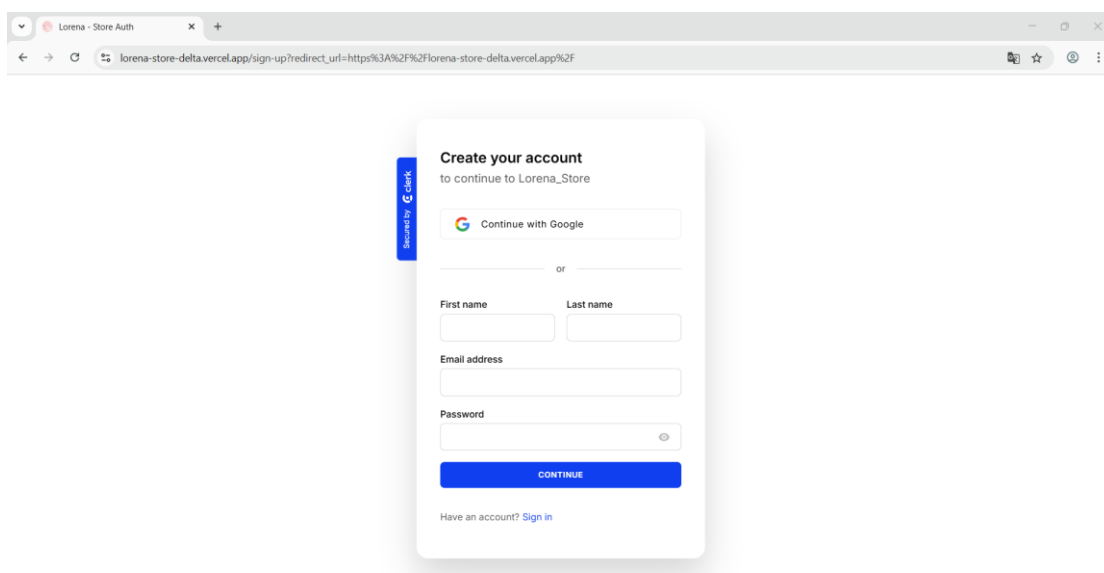


Рисунок 2.6 – Інтерфейс сторінки реєстрації нового користувача

Усі поля форми реєстрації супроводжуються валідацією введених даних, що забезпечує коректність та безпеку інформації, яку надає користувач. Механізми валідації конфігуруються через адміністративну панель сервісу Clerk, що дозволяє гнучко змінювати правила валідації відповідно до потреб проєкту.

У разі, якщо користувач вже має зареєстрований обліковий запис, він може перейти до сторінки входу, натиснувши на кнопку Sign in, розташовану в нижній частині форми реєстрації. Авторизація здійснюється або за допомогою

облікового запису Google, або шляхом введення електронної пошти, пов'язаної з обліковим записом (див. рисунок 2.7).

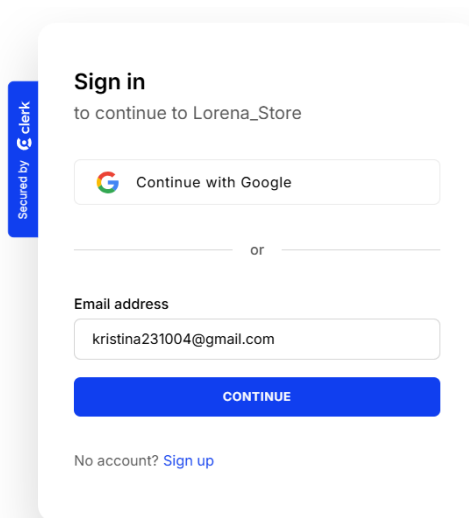


Рисунок 2.7 – Інтерфейс сторінки входу у існуючий обліковий запис

У випадку авторизації за допомогою електронної пошти, після її введення та натискання кнопки Continue, на вказану адресу надсилається одноразовий код підтвердження. Після успішного введення цього коду користувач автоматично перенаправляється на головну сторінку інтернет-магазину, яку можна побачити на рисунку 2.8.

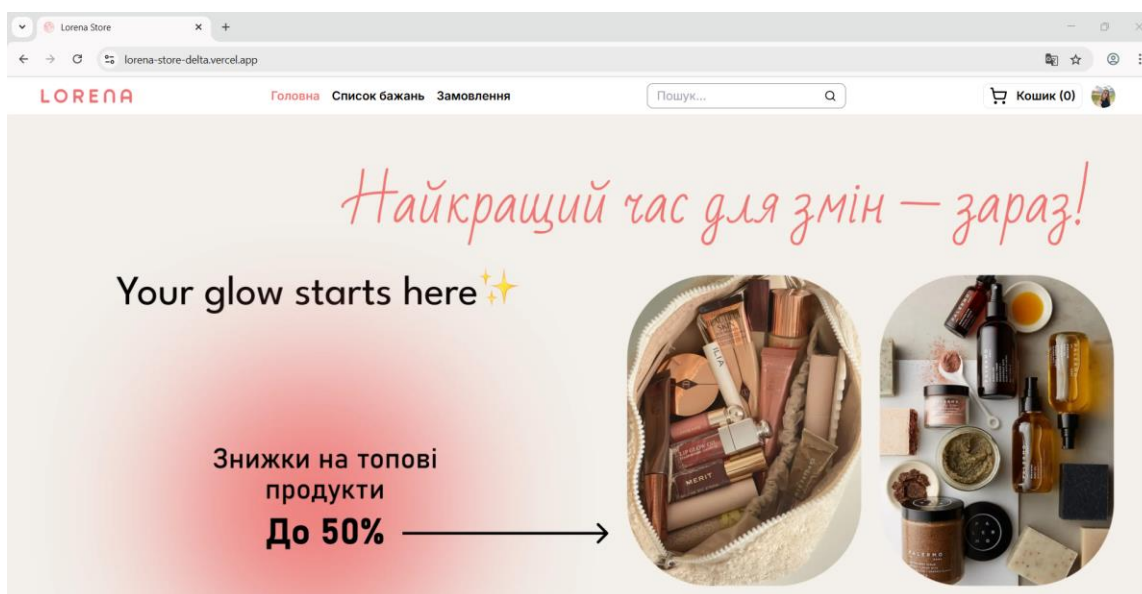


Рисунок 2.8 – Вигляд головної сторінки інтернет-магазину після авторизації

Успішний вхід користувача до системи відображається у зміні інтерфейсу навігаційної панелі (Navbar): у правому верхньому куті з'являється оновлена іконка профілю. Цей елемент представлений у вигляді компонента UserButton, що автоматично генерується службою Clerk. При натисканні на нього користувач отримує доступ до перегляду свого профілю, редагування параметрів облікового запису і можливості виходу з системи (див. рисунок 2.9) [37].

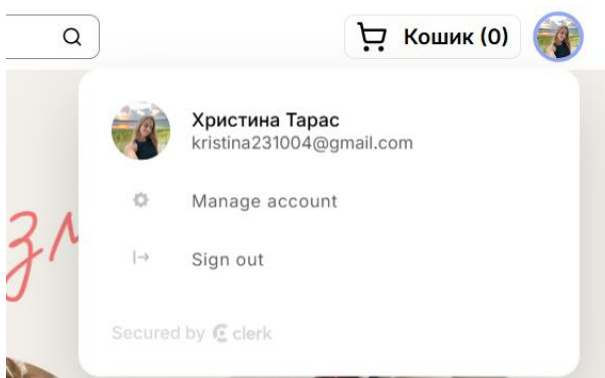


Рисунок 2.9 – Вигляд контекстного меню компонента UserButton

Окрім базових функцій автентифікації, адміністративна панель Clerk також надає інструменти аналітики: адміністратор може переглядати список зареєстрованих користувачів, їхні основні дані, а також час останньої активності в системі (див. рисунок 2.10).

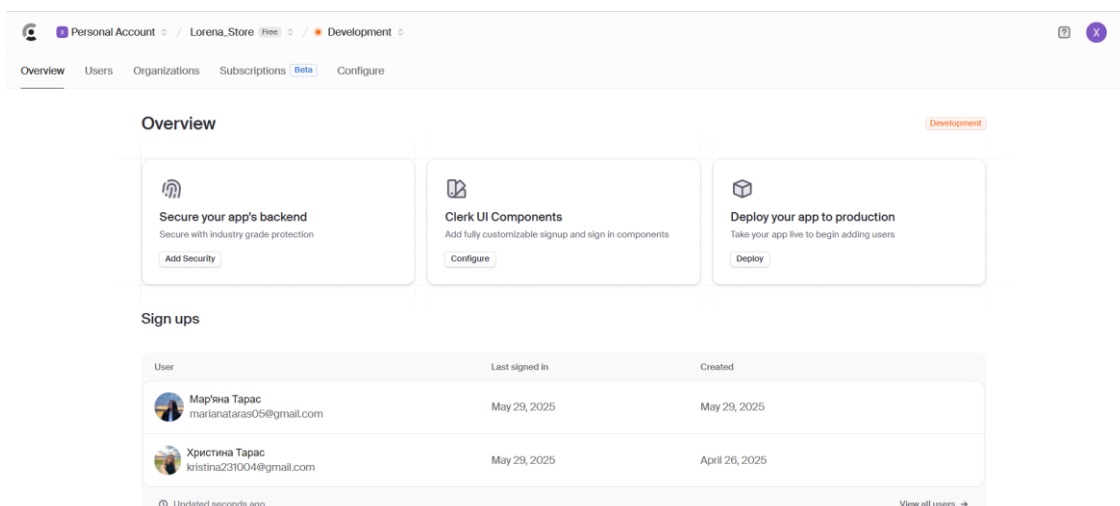


Рисунок 2.10 – Адміністративна панель сервісу Clerk

Наявність аналітичних інструментів у панелі адміністратора Clerk позитивно впливає на керування користувацькою базою. Це дозволяє оперативно виявляти неактивних користувачів, аналізувати динаміку залучення нових акаунтів, оцінювати ефективність маркетингових кампаній або змін у функціоналі платформи. Крім того, доступ до актуальної інформації про користувачів підвищує рівень безпеки системи, даючи змогу своєчасно реагувати на підозрілу активність.

2.3.2 Структура головної сторінки та навігаційного меню

Навігаційна панель інтернет-магазину розташована у верхній частині інтерфейсу та забезпечує швидкий доступ до ключових розділів платформи (див. рисунок 2.11).

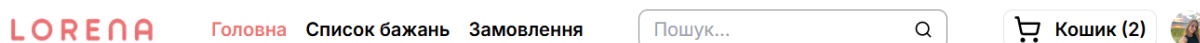


Рисунок 2.11 – Навігаційна панель

Ліворуч розміщено логотип інтернет-магазину, який виконує функцію посилання на головну сторінку. У центральній частині навігаційної панелі знаходяться три основні вкладки: «Головна», «Список бажань» та «Замовлення». Вкладка «Головна» дає змогу повернутися на стартову сторінку інтернет-магазину з будь-якої іншої сторінки. Посилання «Список бажань» перенаправляє користувача на сторінку з переліком товарів, які він попередньо позначив як бажані. Вкладка «Замовлення» відкриває сторінку, де відображено історію замовлень користувача, включаючи дані про товари, дату оформлення та іншу супровідну інформацію.

Головна сторінка інтернет-магазину Lorena виконує функцію вітрини та початкової точки взаємодії користувача із сайтом. Її вигляд показано на рисунку 2.12.

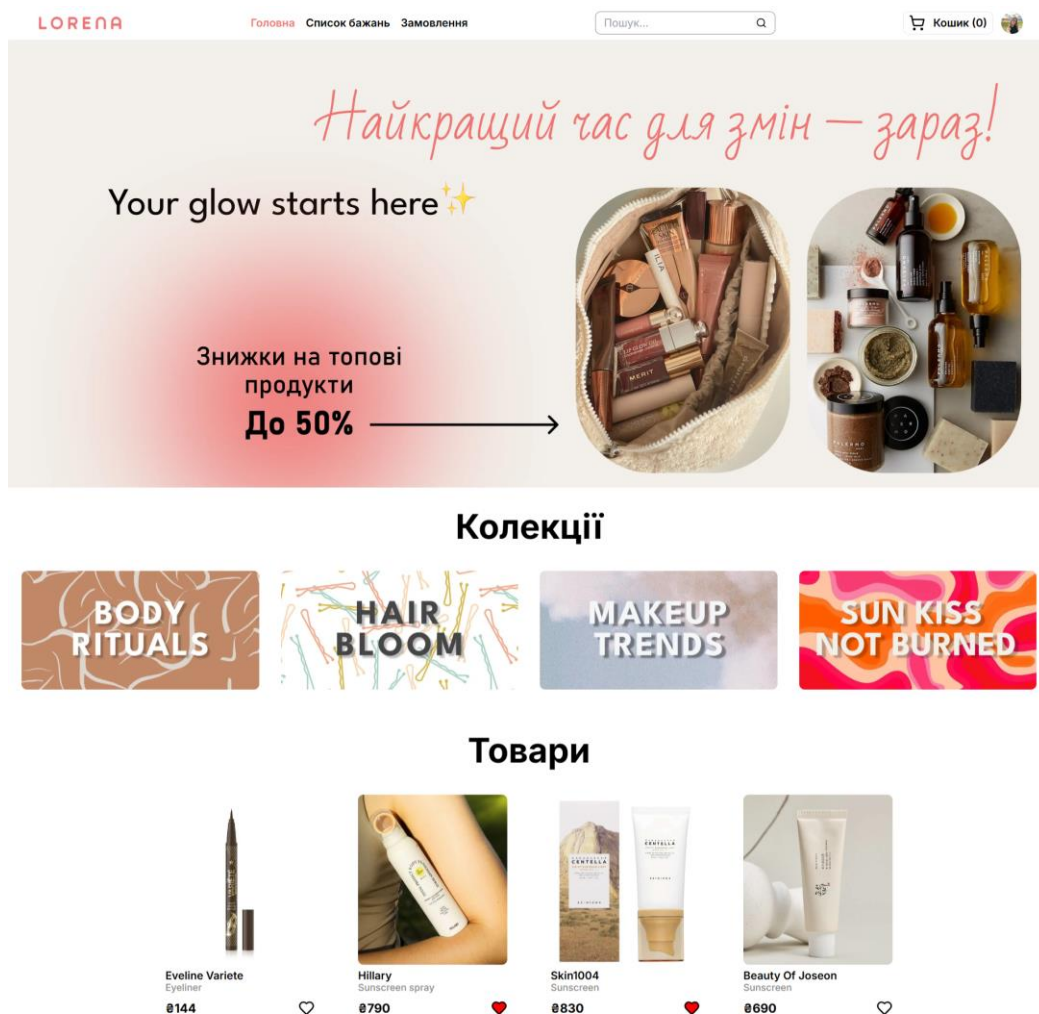


Рисунок 2.12 – Головна сторінка інтернет-магазину Lorena

Вона оформлена у візуально привабливому стилі з акцентом на жіночність, легкість та естетичну привабливість продукції.

Центральним елементом сторінки є банер, що виконує роль промоційного блоку. Він містить рекламний слоган українською та англійською мовами, ілюстрації з прикладами косметичної продукції, а також текстову позначку про знижки на популярні товари. Такий підхід поєднує естетичне оформлення з функціональним повідомленням маркетингового характеру, стимулюючи зацікавленість користувача.

Нижче банера розташовано секцію колекцій, де представлено тематично згруповані категорії товарів: Body Rituals, Hair Bloom, Makeup Trends та Sun Kiss Not Burned. Кожна колекція оформлена в унікальному візуальному стилі,

що підсилює відчуття брендової ідентичності та дозволяє користувачу швидко орієнтуватися у продуктовому асортименті відповідно до власних уподобань.

Завершує головну сторінку галерея товарів, де представлені окремі косметичні продукти із зазначенням назви, типу, ціни та інтерактивних елементів: кнопки додавання до списку бажань (позначена іконкою серця) і можливість перегляду детальної інформації про товар. Кожен товар супроводжується якісною фотографією, що дозволяє користувачу швидко оцінити зовнішній вигляд продукції.

Загалом головна сторінка демонструє логічну та зручну структуру, поєднуючи елементи навігації, візуального маркетингу й каталогу продукції, що сприяє комфортному та ефективному користувацькому досвіду.

2.3.3 Додавання товарів у список бажань та перегляд сторінки продукту

Авторизовані користувачі можуть додавати товари до списку бажань із головної або будь-якої сторінки каталогу. Біля кожного товару розміщено іконку-серце (див. рис. 2.13), натискання на яку додає товар до персонального списку, що зберігається в базі даних із прив'язкою до ідентифікатора користувача.

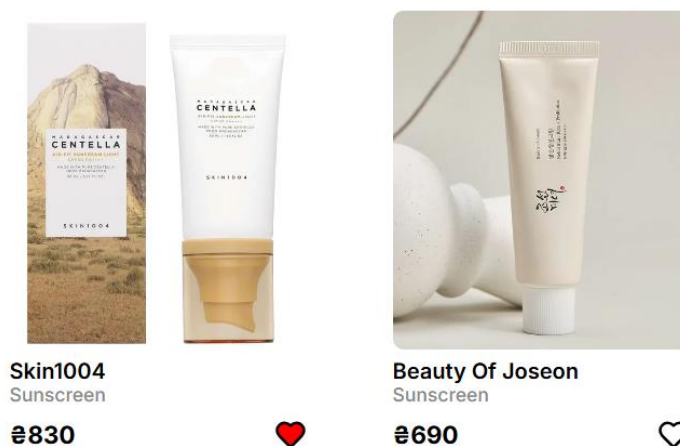


Рисунок 2.13 – Додавання товару до списку бажань за допомогою іконки-серця

Цей функціонал реалізовано за допомогою перевірки стану автентифікації. У разі, якщо неавторизований користувач намагається додати товар до списку, система перенаправляє його на сторінку входу або реєстрації. Такий підхід забезпечує цілісність даних і дозволяє зберігати персоналізовану інформацію лише для зареєстрованих користувачів.

Доступ до сторінки «Список бажань» здійснюється виключно через навігаційне меню інтернет-магазину. Інтерфейс сторінки є мінімалістичним: на ній відображається перелік товарів, які користувач попередньо додав до списку (див. рисунок 2.14).

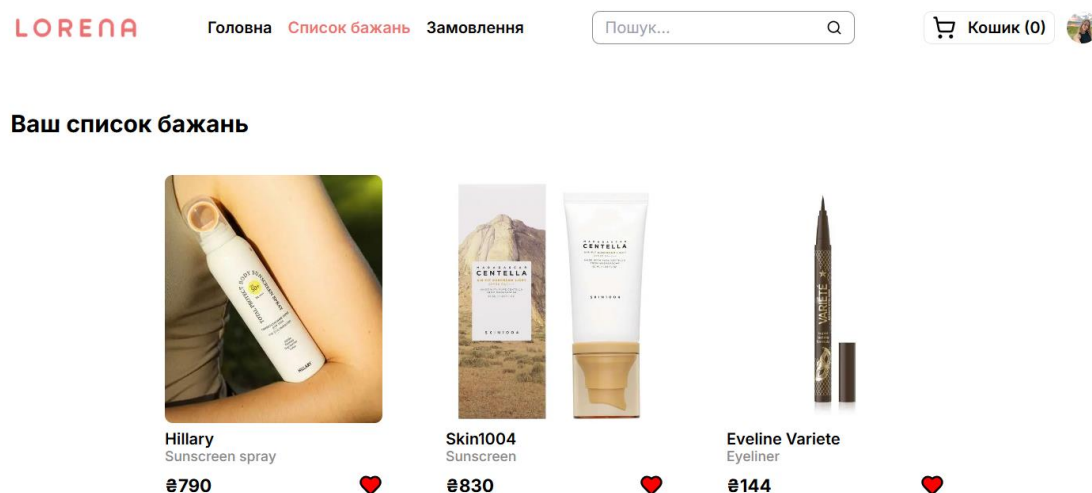


Рисунок 2.14 – Сторінка «Список бажань»

Крім цього, кожен товар містить посилання на сторінку з детальним описом, яку можна відкрити, натиснувши на зображення або назву продукту (див. рисунок 2.15).

На сторінці окремого товару відображається велике зображення продукту, його назва, категорія, ціна, а також докладний опис, який містить інформацію про призначення засобу, основні інгредієнти та ефект від його використання. Окрім цього, користувач має можливість обрати бажану варіацію, розмір/об'єм, вказати кількість одиниць та додати обраний товар до кошика за допомогою відповідної кнопки.

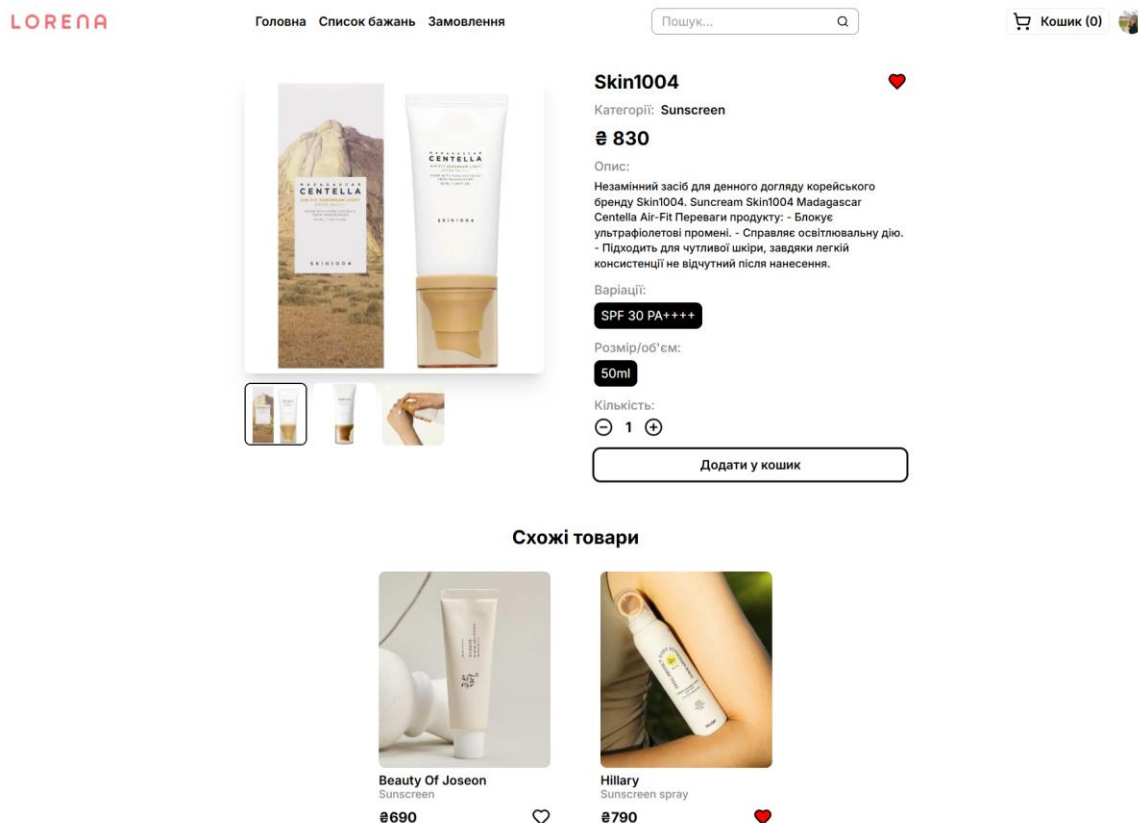


Рисунок 2.15 – Сторінка продукту

Під основною інформацією про товар на сторінці розміщено блок «Схожі товари», який відображає продукти, що потенційно можуть зацікавити користувача. Виведення цього блоку реалізовано на основі динамічного запиту до API, що відбувається за допомогою функції `getRelatedProducts(productId: string)`. Ця функція надсилає HTTP-запит на маршрут та отримує у відповідь масив пов'язаних товарів у форматі JSON (див. лістинг).

```
${process.env.NEXT_PUBLIC_API_URL}/products/${productId}/related
```

Подальше відображення кожного з елементів масиву здійснюється через компонент `ProductCard`. Вибірка схожих товарів виконується на сервері. Як правило, алгоритм формування таких рекомендацій базується на збігу категорій, схожості характеристик (категорія, теги, бренд тощо) або популярності серед користувачів, які переглядали даний товар. Це дозволяє

персоналізувати взаємодію користувача з сайтом, стимулюючи подальший перегляд каталогу та підвищуючи ймовірність покупки.

2.3.4 Додавання товарів у кошик, оформлення та оплата замовлення

Функціональність додавання товарів до кошика є однією з ключових для будь-якого інтернет-магазину. У клієнтській частині розробленого застосунку Lorena ця опція реалізована за допомогою інтерактивної кнопки «Додати у кошик», яка розміщується на сторінці кожного товару. Натискання на цю кнопку ініціює функцію `cart.addItem`, яка передає об'єкт із параметрами вибраного товару: інформацію про сам товар (`productInfo`), кількість (`quantity`), обраний колір (`selectedColor`) та розмір (`selectedSize`). Ця взаємодія реалізована у вигляді обробника події `onClick`, прикріпленого до кнопки (див. лістинг 2.1).

Лістинг 2.1 – Додавання товару до кошика

```
<button
  className="outline text-base-bold py-3 rounded-lg hover:bg-
black hover:text-white"
  onClick={() => {
    cart.addItem({
      item: productInfo,
      quantity,
      color: selectedColor,
      size: selectedSize,
    });
  }}
>
  Додати у кошик
</button>
```

Кнопка має стилізацію, яка відповідає загальному дизайну застосунку, зокрема зміну кольору фону та тексту при наведенні курсора. Завдяки модульному підходу, логіка додавання до кошика є інкапсульованою, що забезпечує гнучкість та масштабованість функціоналу. Це дозволяє зберігати вибрані користувачем параметри товару безпосередньо у клієнтському сховищі або керованому глобальному стані додатку.

Користувач може переглянути вміст кошика за допомогою відповідної іконки, розміщеної у верхній панелі навігації. Перехід до сторінки кошика надає змогу переглянути обрані товари, змінити кількість одиниць, обрати розмір або колір, а також видалити непотрібні позиції (див. рисунок 2.16).

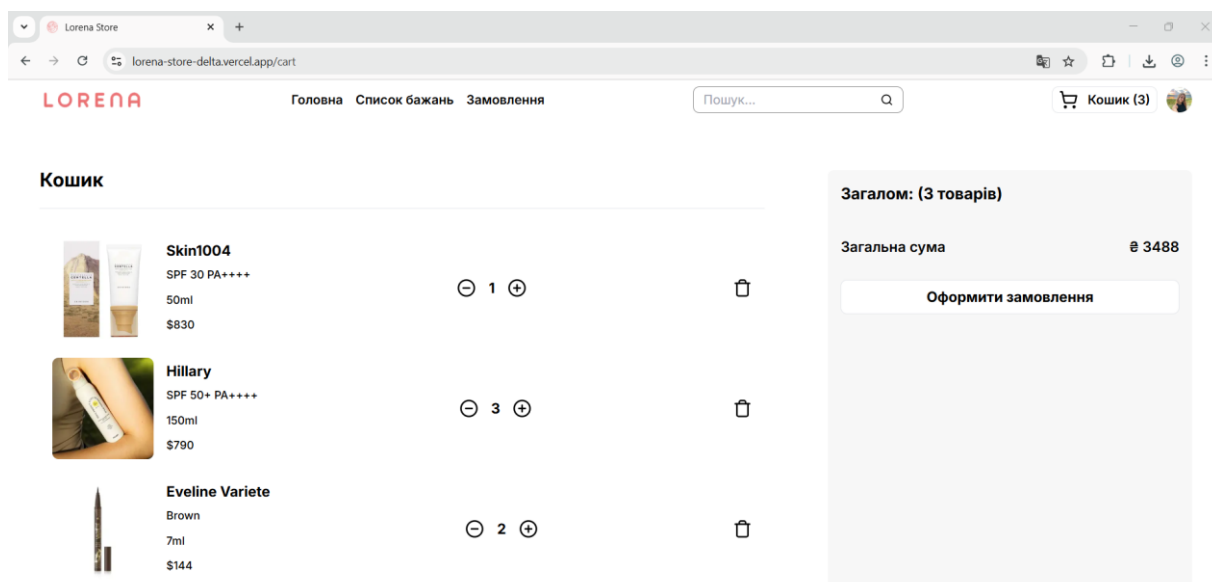


Рисунок 2.16 – Вигляд сторінки кошика

У правій частині інтерфейсу сторінки кошика розміщується зведена інформація про поточне замовлення. Зокрема, відображається загальна кількість доданих товарів, сума до сплати з урахуванням обраних параметрів, а також інтерактивна кнопка з написом «Оформити замовлення». Цей елемент інтерфейсу виконує функцію фінального підтвердження покупки, після чого користувача перенаправляється до процесу введення платіжних даних і здійснення оплати. Такий підхід дозволяє забезпечити зручність користування для покупця та логічну послідовність дій при формуванні замовлення.

Оформлення замовлення (checkout) передбачає заповнення форми з персональними та адресними даними користувача. На цьому етапі також відображається фінальна сума замовлення з урахуванням кількості обраних товарів та обраного способу доставки. Користувач має змогу вибрати один із

запропонованих методів доставки, зокрема — безкоштовну стандартну доставку (5–7 днів) або швидку доставку (1–3 дні) за додаткову плату.

← Lorena

Sandbox

Pay Lorena

UAH 2,897.00

Skin1004

Qty 1

UAH 830.00

Hillary

Qty 2

UAH 1,580.00

Eveline Variete

Qty 2

UAH 288.00

Subtotal

UAH 2,698.00

Shipping

Швидка доставка (1-3 days)

UAH 199.00

Total due

UAH 2,897.00

Pay with link

Or

Shipping information

Email

kristina231004@gmail.com

Shipping address

Khrystyna Taras

Ukraine

Halyska Street, 7D

Address line 2

Ternopil

Тернопільська область — Ternopil's'ka oblast

46002

Shipping method

☐

Безкоштовна доставка

5-7 days

Free

☒

Швидка доставка

1-3 days

UAH 199.00

Payment details

Card information

4242 4242 4242 4242

08 / 27

158

VISA

☒

Billing info is same as shipping

☐

Securely save my information for 1-click checkout

Pay faster on Lorena and everywhere Link is accepted.

Pay

Powered by stripe

Terms Privacy

Рисунок 2.17 – Сторінка оформлення замовлення

Користувач може завершити оплату, ввівши платіжні реквізити або скориставшись інтегрованим сервісом швидкої оплати "Link". Після успішної транзакції інформація про замовлення зберігається в базі даних, і користувач отримує підтвердження (див. рисунку 2.18).

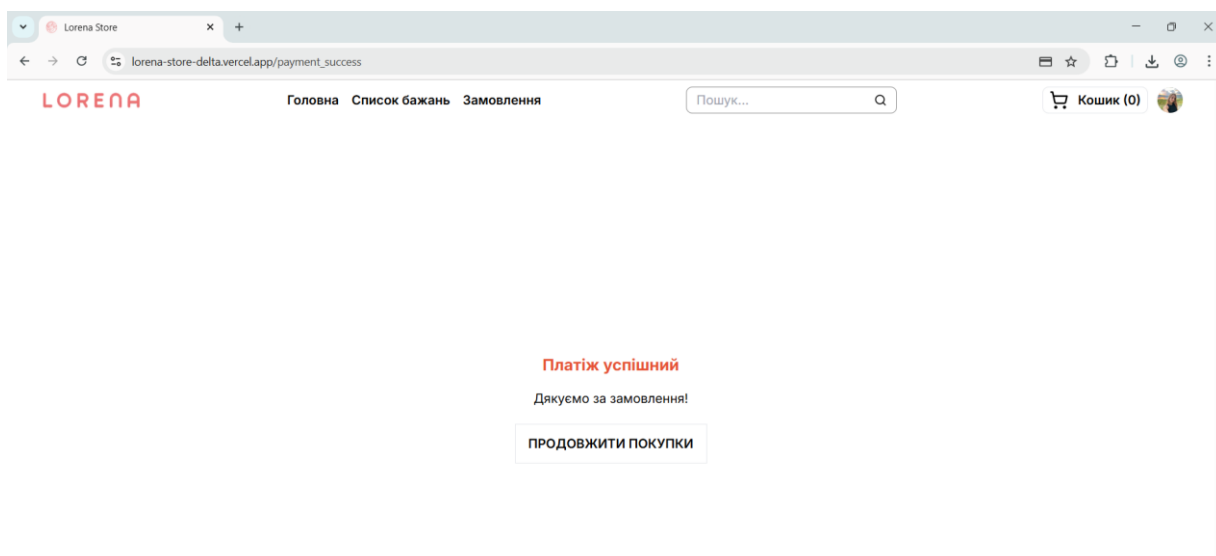


Рисунок 2.18 – Підтвердження успішності платежу

Після успішного завершення транзакції оформлене замовлення автоматично зберігається в системі та стає доступним для перегляду у вкладці «Замовлення», що розташована у верхній навігаційній панелі інтерфейсу користувача. Детальний огляд розділу «Замовлення» буде наведено під час огляду адміністративної частини інтернет-магазину Lorena.

2.4 Огляд функціоналу адміністративної частини інтернет-магазину

2.4.1 Реєстрація користувачів та їх авторизація у системі

Доступ до адміністративної частини інтернет-магазину Lorena реалізовано за допомогою зовнішнього сервісу аутентифікації Clerk, який забезпечує як процес реєстрації нових користувачів, так і їх подальшу авторизацію в системі. Посилання на адміністративний інтерфейс не є загальнодоступним і надається виключно користувачам із відповідними правами доступу — адміністраторам, власникам магазину або іншим уповноваженим особам.

Під час реєстрації за допомогою Clerk користувачеві необхідно вказати базові дані (електронну пошту, пароль тощо), після чого система створює

обліковий запис. Надалі для входу в адміністративний інтерфейс необхідна авторизація, яка також здійснюється через Clerk із використанням захищеного механізму аутентифікації. Процеси реєстрації та авторизації є ідентичними, як у клієнтській частині сайту, включаючи вигляд сторінок.

Таким чином, рішення із використанням Clerk не лише спрощує інтеграцію реєстраційно-авторизаційного механізму, але й гарантує високий рівень безпеки, розмежування ролей та контроль за доступом до конфіденційного функціоналу магазину.

2.4.2 Інформаційна панель адміністратора

Після успішної авторизації адміністратор потрапляє на головну сторінку адміністративної частини інтернет-магазину Lorena інформаційну панель (див. рисунок 2.19).

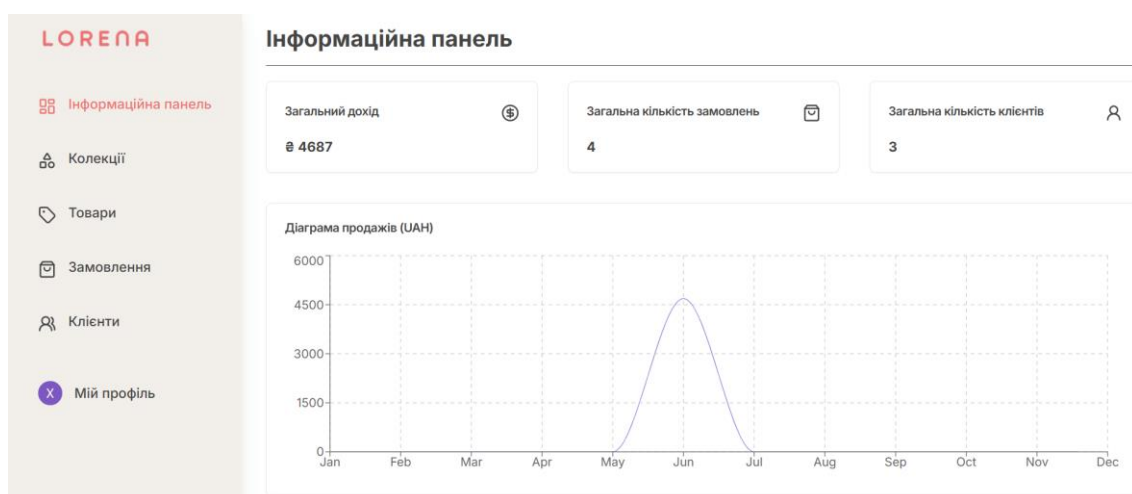


Рисунок 2.19 – Вигляд адміністративної частини інтернет-магазину Lorena

Вона відіграє роль централізованого оглядового модуля, що дозволяє швидко оцінити поточний стан магазину за ключовими показниками.

На інформаційній панелі відображаються такі статистичні дані:

- загальний дохід – сума всіх оплат, які пройшли через систему;
- загальна кількість замовлень – число підтверджених покупок;

- загальна кількість клієнтів кількість зареєстрованих клієнтів у системі.

Крім того, нижче розміщено діаграму продажів, яка візуалізує динаміку щомісячного доходу у гривнях (UAH). Такий графік допомагає аналізувати ефективність роботи магазину впродовж року.

Зліва розташовано вертикальне меню навігації, яке містить наступні розділи:

- колекції управління категоріями продукції;
- товари створення, редагування та видалення продуктів;
- замовлення перегляд історії замовлень і їхній статус;
- клієнти інформація про покупців;
- мій профіль налаштування облікового запису адміністратора.

Такий інтерфейс дозволяє зручно управляти всіма аспектами онлайн-магазину з однієї централізованої точки доступу.

2.4.3 Управління колекціями та додавання нових товарів

Однією з ключових можливостей адміністративної частини інтернет-магазину Lorena є гнучке управління товарним асортиментом, зокрема створення колекцій та додавання нових товарів.

У вкладці «Колекції» (див. рисунок 2.20) адміністратор має змогу створювати нові категорії продукції, які групують товари за певною тематикою або призначенням (наприклад: Догляд за шкірою, Макіяж, Подарункові набори тощо).

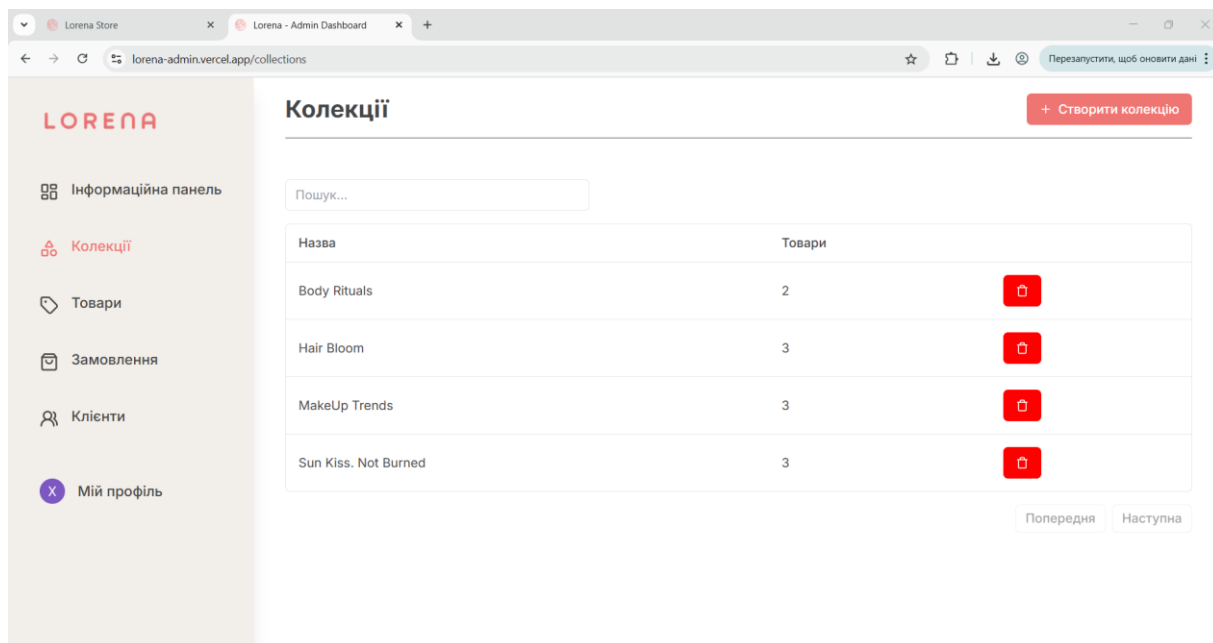


Рисунок 2.20 – Вигляд сторінки з інформацією про колекції

Процес створення нової колекції включає:

- натискання кнопки «Додати колекцію», що перенаправляє на сторінку із формою;
- введення назви колекції, додавання короткого опису та зображення, що ілюструє цю категорію (див. рисунок 2.21);
- підтвердження створення шляхом натискання кнопки «Зберегти».

Рисунок 2.21 – Вигляд форми створення нової колекції

Усі створені колекції автоматично зберігаються у базі даних та стають доступними при створенні товарів.

У вкладці «Товари» адміністратор може переглянути інформацію про вже існуючі товари або створити картку нового продукту, натиснувши кнопку «Додати товар» (див. рисунок 2.22).

Назва	Категорія	Колекції	Продажна ціна (€)	Закупівельна ціна (€)	
Mixa Urea Cica	Body Cream	Body Rituals	290	235	
NIVEA Creme Soft	Shower Gel	Body Rituals	103	87	
Syoss Oleo Intense	Hair Conditioner	Hair Bloom	149	112	
Daeng Gi Meo Ri	Hair Conditioner	Hair Bloom	119	100	
OGX Morocco Shampoo	Shampoo	Hair Bloom	300	255	
Focallure Go Travel	Eyeshadow	MakeUp Trends	740	650	
Pinkflash Chic	Blush	MakeUp Trends	55	42	

Рисунок 2.22 – Сторінка з інформацією про товари

Функціонал форми додавання нового товару передбачає введення всіх необхідних характеристик товару, а саме:

- назва товару офіційна назва продукту;
- опис докладна інформація про властивості, склад, спосіб застосування або інші важливі деталі;
- продажна ціна та закупівельна ціна це дозволяє автоматично обчислювати чистий дохід з продажу;
- зображення товару для візуального представлення товару;
- колекція та категорія вибір розділу, до якого належить товар;
- теги ключові слова, за якими користувач зможе легко знайти товар у пошуку;
- варіації товару такі як запахи, кольори, відтінки;
- розміри товару наприклад, об'єм (мл), вага (г), тощо.

Після заповнення всіх обов’язкових полів, адміністратор натискає кнопку «Зберегти», після чого новий товар з’являється у відповідній категорії в інтерфейсі користувача (див. рисунок 2.23).

The screenshot shows a web browser window with two tabs: 'Lorena Store' and 'Lorena - Admin Dashboard'. The active tab is 'Lorena - Admin Dashboard' at the URL 'lorena-admin.vercel.app/products/new'. The page title is 'Створити товар' (Create Product). The form contains the following fields:

- Назва** (Name): A text input field.
- Опис** (Description): A large text area.
- Зображення** (Image): A section with a '+ Завантажити зображення' (Upload Image) button.
- Продажна ціна (€)** (Selling Price): A text input field with '0,1' entered.
- Закупівельна ціна (€)** (Purchase Price): A text input field with '0,1' entered.
- Категорія** (Category): A dropdown menu with 'Категорії' selected.
- Теги** (Tags): A text input field with 'Теги' entered.
- Колекції** (Collection): A dropdown menu with 'Q. Колекції' selected.
- Варіанти** (Variants): A text input field with 'Варіанти' entered.
- Сizes** (Sizes): A text input field with 'Розміри/об'єм' entered.

At the bottom of the form are two buttons: 'Зберегти' (Save) and 'Скасувати' (Cancel).

Рисунок 2.23 – Форма додавання нового товару

Такий функціонал забезпечує повний контроль над асортиментом магазину, дозволяє гнучко адаптувати пропозицію до потреб клієнтів і легко підтримувати актуальність товарів у продажу

2.4.4 Управління замовленнями та клієнтами

Адміністративна частина платформи Lorena містить окремі вкладки «Замовлення» та «Клієнти», які забезпечують зручний контроль над процесами продажу та взаємодії з покупцями.

Вкладка «Замовлення» дозволяє адміністраторам бачити список усіх замовлень, що надходять через інтернет-магазин. Інтерфейс включає такі стовпці:

- замовлення унікальний ідентифікатор або номер замовлення;
- клієнт ім'я або контактні дані покупця;

- товари список придбаних товарів у межах цього замовлення;
- загальна сума (₴) підсумкова вартість усього замовлення;
- час створення дата і час оформлення замовлення.

У верхній частині присутній рядок пошуку, що дозволяє швидко знаходити потрібне замовлення за ключовими словами (див. рисунок 2.24). Також передбачено пагінацію (кнопки «Попередня» та «Наступна») для зручної навігації між сторінками списку замовлень.

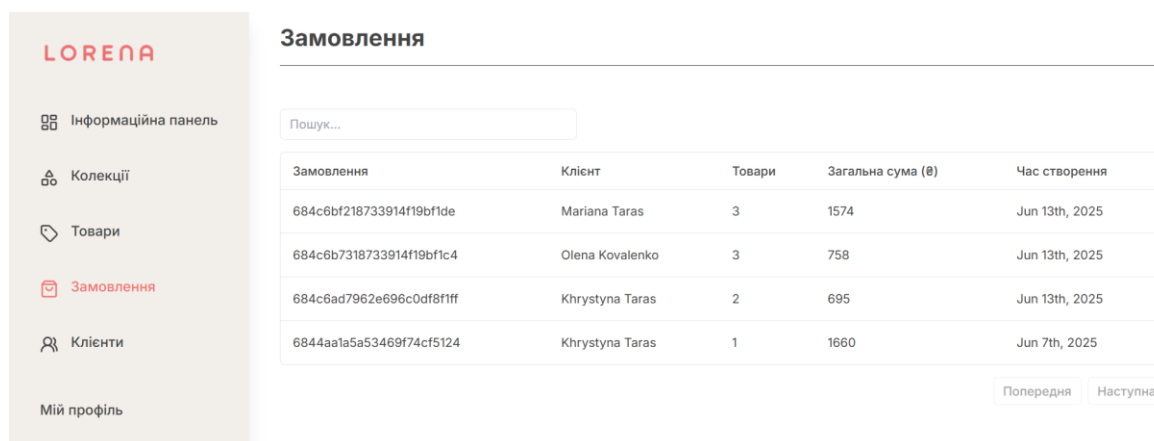


Рисунок 2.24 – Інтерфейс вкладки «Замовлення» у кабінеті адміністратора

У вкладці «Клієнти» адміністратор отримує доступ до даних зареєстрованих користувачів (див. рисунок 2.25). Таблиця містить наступні стовпці:

- ID Клієнта унікальний ідентифікатор клієнта, який формується через сервіс автентифікації Clerk;
- ім'я ім'я, вказане користувачем при реєстрації;
- ел. пошта електронна адреса, пов'язана з обліковим записом.

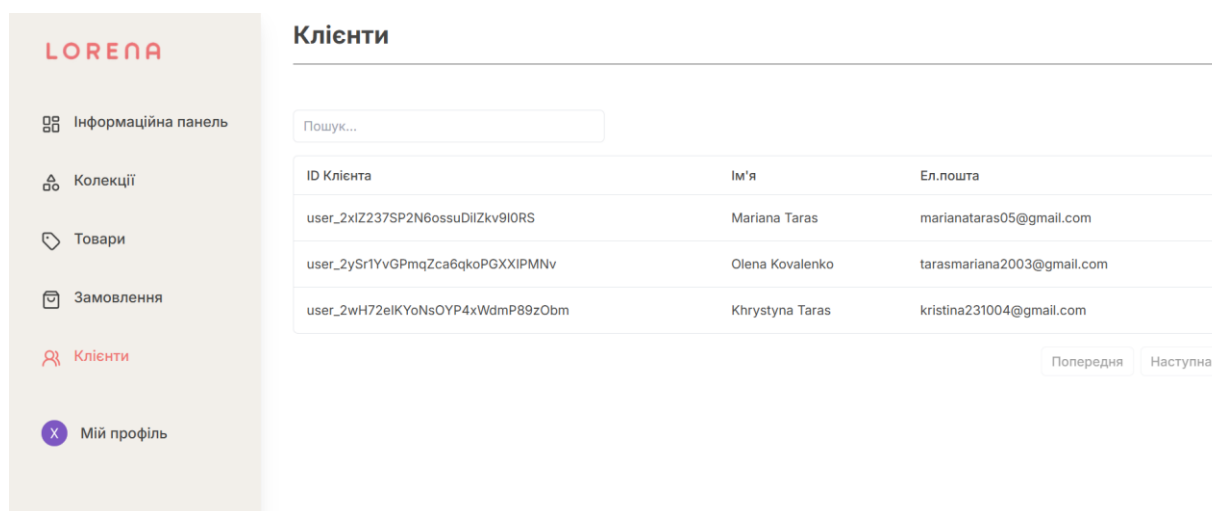


Рисунок 2.25 – Вкладка «Клієнти» у кабінеті адміністратора

Для пошуку клієнтів реалізоване поле фільтрації. У нижній частині також присутні кнопки перемикавання між сторінками (у разі великої кількості записів).

2.5 Огляд дизайну інтернет-магазину Lorena

2.5.1 Кольорова схема інтернет-магазину та логотипи

Візуальна ідентичність інтернет-магазину Lorena сформована на основі м'якої та естетично привабливої кольорової гами, що викликає відчуття чистоти, жіночності й турботи. Основні кольори, використані у дизайні наведено на рисунку 2.26.



Рисунок 2.26 – Основні кольори інтерфейсу та стилю бренду Lorena

Нижче наведено опис кожного з кольорів та їхнє призначення в інтерфейсі інтернет-магазину:

- `#EF7573` основний акцентний колір бренду. Використовується у логотипі, кнопках та ключових візуальних елементах для привернення уваги;
- `#F8F8F8` світлий фон, що створює враження простору та легкості в інтерфейсі;
- `#454545` насичений темно-сірий колір, який забезпечує чіткість основного тексту та зручність читання;
- `#E5E7EB` використовується для побудови меж, ліній поділу та другорядних елементів інтерфейсу;
- `#F2EFEA` та `#F5F7F9` ніжні пастельні відтінки, що виступають допоміжними фонами та підкреслюють візуальну гармонійність;
- `#FF0000` яскравий червоний колір, зарезервований для повідомлень про помилки або важливі попередження.

Усі ці кольори були централізовано оголошені у конфігураційному файлі `tailwind.config.ts` (див. лістинг 2.2).

Лістинг 2.2 – Фрагмент конфігурації кольорової палітри в Tailwind CSS (файл `tailwind.config.ts`)

```
extend: {  
  colors: {  
    "white-1": "#F8F8F8",  
    "grey-1": "#454545",  
    "grey-2": "#E5E7EB",  
    "pink-1": "#EF7573",  
    "beige-2": "#F2EFEA",  
    "beige-3": "#F5F7F9",  
    "red-1": "#FF0000",  
  },  
},
```

Це дозволяє легко масштабувати або модифікувати дизайн у разі потреби. Наприклад, якщо виникає потреба змінити акцентний колір на зовсім інший (скажімо, з рожевого на зелений), достатньо змінити одне значення в цьому файлі. Tailwind автоматично оновить усі компоненти, які використовують цей

колір. Такий підхід забезпечує гнучкість, підтримку стилістичної єдності та пришвидшує роботу над редизайном.

Логотип Lorena був спеціально розроблений для бренду й відображає його головні цінності: ніжність, красу та елегантність. Логотип представлений у кількох варіантах:

- основний логотип стилізоване зображення жіночого обличчя в круглому контурі поруч з написом “LORENA”;
- скорочений варіант лише графічна іконка, яка використовується як favicon або в мобільних інтерфейсах;
- текстовий логотип назва бренду без графічного елемента, що застосовується у вузьких компонентах, таких як хедер або мобільна навігація.



Рисунок 2.27 – Основна та альтернативні версії логотипу Lorena

Логотип реалізовано у фірмовому кольорі #EF7573, що гармонійно поєднується з усією палітрою проєкту. Стиль шрифтів м’який, округлий, що підсилює загальну атмосферу дружності та турботи.

2.5.2 Розробка адаптивного дизайну за допомогою Tailwind CSS

Адаптивний дизайн є важливою ознакою сучасної веброзробки, оскільки дозволяє забезпечити коректне відображення інтерфейсу на різних пристроях – від мобільних телефонів до настільних комп’ютерів. У рамках розробки інтернет-магазину Lorena було обрано утилітарний CSS-фреймворк Tailwind

CSS, який значно спрощує реалізацію адаптивного та стилістично уніфікованого інтерфейсу.

Tailwind CSS надає набір готових класів, що дозволяють задавати стилі безпосередньо в HTML-розмітці, зокрема розміри шрифтів, відступи, кольори, тіні, рамки та інше. Для реалізації адаптивності використовувалися breakpoints ключові точки перегляду (наприклад, sm, md, lg, xl), які задають правила відображення компонентів в залежності від ширини екрану.

Прикладом є використання класів типу `max-md:px-3` чи `md:flex-row`, які дозволяють по-різному компоувати елементи на малих та середніх екранах. Таке рішення гарантує, що структура сторінок буде залишатися зручною та естетично привабливою незалежно від пристрою, з якого користувач здійснює перегляд (див. рисунок 2.28).

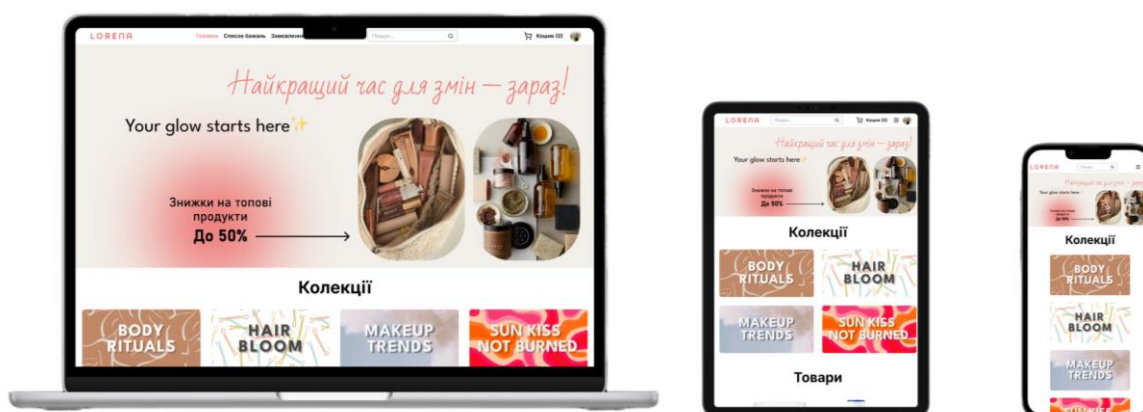


Рисунок 2.28 – Відображення інтерфейсу інтернет-магазину Lorena на різних типах пристроїв (десктоп, планшет, смартфон)

Крім того, Tailwind CSS дозволив уникнути дублювання стилів і досягти більшої уніфікації в оформленні. Розробка адаптивного дизайну також передбачала тестування на різних екранах, включно з мобільними, планшетами та десктопами, що дало змогу виявити та усунути потенційні візуальні дефекти до моменту розгортання проєкту.

2.6 Висновок до другого розділу

У другому розділі було детально розглянуто процес проектування та реалізації інтернет-магазину Lorena. Описано вибір технологічного стеку, архітектуру вебзастосунку, структуру файлової системи та використані бібліотеки як для клієнтської, так і для адміністративної частини. Особливу увагу приділено моделюванню бази даних, реалізації ключових функціональних можливостей для користувачів, включаючи авторизацію, перегляд товарів, роботу зі списком бажань, кошиком і оформленням замовлення.

Окремим блоком представлено адміністративну панель, яка надає власникам магазину інструменти для керування колекціями, товарами, клієнтами та замовленнями. Також розглянуто підходи до візуального оформлення проєкту, зокрема реалізацію адаптивного дизайну засобами Tailwind CSS та розробку уніфікованої кольорової схеми.

Результати, викладені в цьому розділі, демонструють комплексний підхід до створення повноцінного e-commerce рішення, що відповідає сучасним вимогам зручності, функціональності та візуальної привабливості.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ LORENA

3.1 Розгортання інтернет-магазину в Інтернеті

Для розгортання клієнтської та адміністративної частини інтернет-магазину Lorena було обрано платформу Vercel, яка є спеціалізованим рішенням для хостингу застосунків, розроблених із використанням фреймворку Next.js. Цей вибір обумовлений насамперед високим рівнем сумісності Vercel з обраною технологією, адже саме Next.js лежить в основі розробленого вебзастосунку. Платформа надає зручний інтерфейс, підтримує автоматичне розгортання після кожного оновлення в репозиторії GitHub та забезпечує оптимізацію продуктивності через вбудовану підтримку серверного рендерингу [38].

Розгортання інтернет-магазину здійснювалося через підключення репозиторію проєкту до облікового запису на Vercel. Після авторизації система автоматично розпізнала використовуваний фреймворк та запропонувала типове середовище для запуску. Далі було здійснено налаштування змінних середовища, необхідних для роботи інтеграцій із сервісами Clerk, Stripe та MongoDB (див. рисунок 3.1).

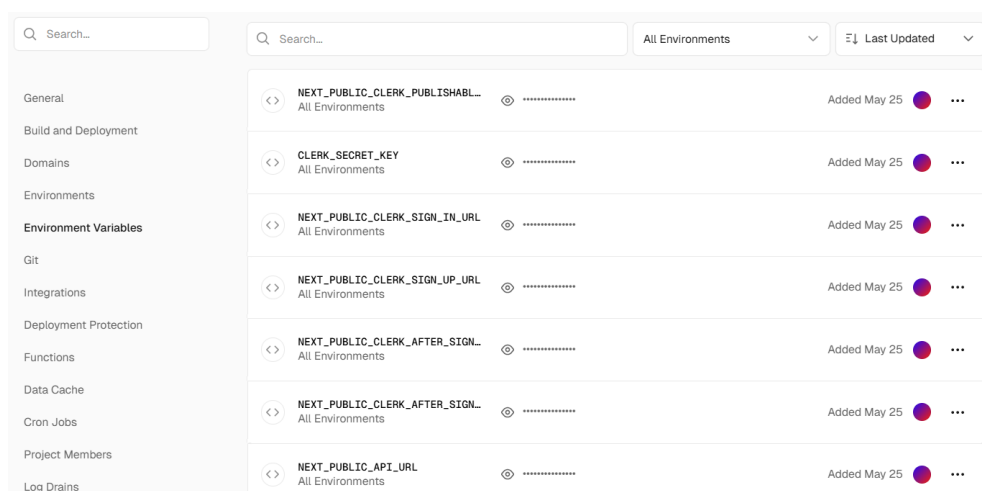


Рисунок 3.1 – Налаштування змінних середовища на платформі Vercel

Завершальним етапом стала автоматична збірка та публікація сайту на платформі, після чого магазин став доступним онлайн за наданою Vercel адресою.

Вибір Vercel як платформи для хостингу дозволив забезпечити швидке розгортання, стабільну роботу сайту та зручне управління середовищем розробки та продакшн-версією інтернет-магазину.

3.2 Валідація й тестування функціональності інтернет-магазину

3.2.1 Валідація розмітки інтерфейсу та форм

Для перевірки якості HTML-розмітки інтернет-магазину Lorena було використано сервіс DataYze Site Validator, який дозволяє здійснити автоматичну перевірку структури сторінки, її відповідності базовим вимогам до доступності, SEO та загальної якості коду [39]. У результаті валідації було виявлено дрібні зауваження, які стосувалися відсутності деяких атрибутів або дублювання елементів (див. рисунок 3.2).

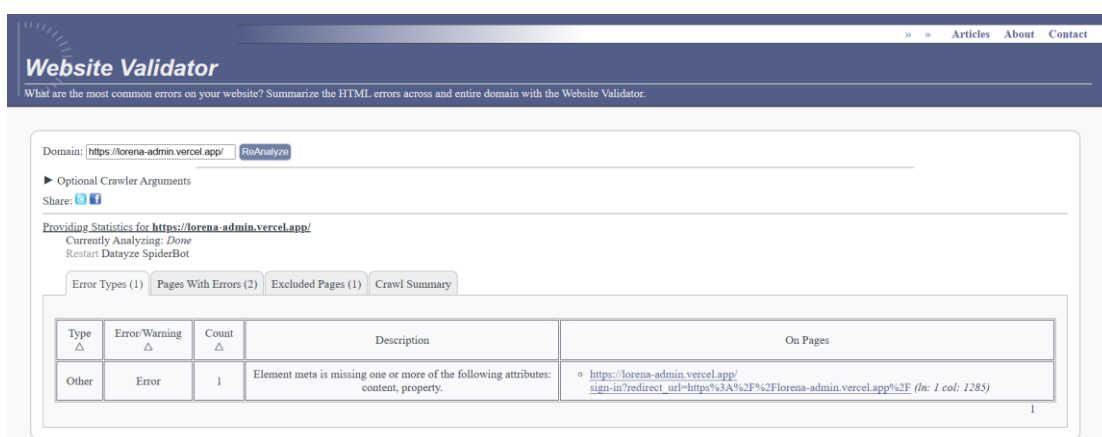


Рисунок 3.2 – Результат перевірки сайту на наявність помилок

Після виправлення зауважень сторінки сайту успішно пройшли перевірку на коректність, що підтверджує їхню технічну якість та придатність до публічного використання.

Окрему увагу приділено формам як на клієнтській, так і на адміністративній частинах проєкту. Усі форми проходять валідацію введених даних за допомогою бібліотеки zod у поєднанні з react-hook-form. Такий підхід дозволяє ефективно перевіряти правильність форматів електронної пошти, паролів, номерів телефонів та інших обов'язкових полів, а також забезпечує миттєвий зворотний зв'язок із користувачем при введенні даних. Це підвищує зручність взаємодії з сайтом та зменшує кількість помилкових запитів.

3.2.2 Перевірка інтернет-магазину на кросбраузерність та тестування його швидкості

У процесі тестування інтернет-магазину Lorena важливу роль відіграла перевірка кросбраузерної сумісності. Сайт було протестовано в найпопулярніших веббраузерах, зокрема Google Chrome, Mozilla Firefox, Microsoft Edge, Safari та Opera (див. рисунок 3.3).

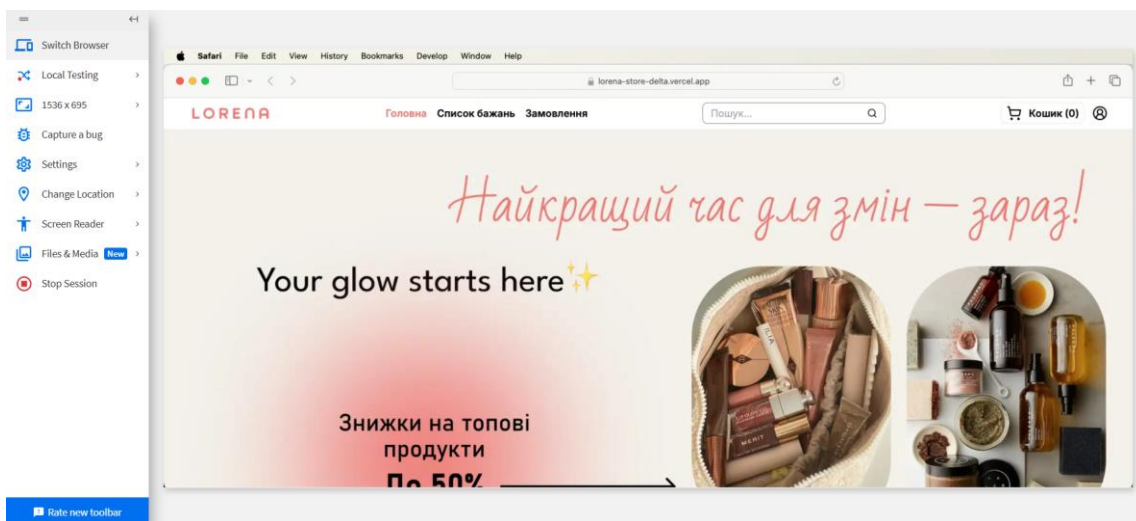


Рисунок 3.3 – Процес тестування продуктивності клієнтської частини сайту Lorena за допомогою PageSpeed Insights

У результаті перевірки підтверджено коректне відображення всіх елементів інтерфейсу, функціонування інтерактивних компонентів і загальну стабільність роботи. Завдяки використанню сучасних технологій розробки

таких як Tailwind CSS, Next.js і React — вдалося досягти високої сумісності з актуальними версіями браузерів.

Окрім цього, було проведено тестування швидкодії сайту за допомогою сервісу PageSpeed Insights від Google, оскільки він забезпечує комплексну оцінку продуктивності на основі реальних даних користувачів (field data) та лабораторних вимірів (lab data), а також дає практичні поради з покращення показників, таких як LCP, CLS, TTI тощо [40]. Таблиця 3.1 відображає результати тестування швидкодії клієнтської та адміністративної частин інтернет-магазину Lorena для різних типів пристроїв.

Таблиця 3.1 – Результати тестування швидкодії клієнтської та адміністративної частин інтернет-магазину Lorena за допомогою сервісу PageSpeed Insights

Платформа	Пристрій	Ефективність	Доступність	Оптимальні методи	Оптим. пошук. систем
Клієнтська частина	Комп'ютер	90	84	96	100
Клієнтська частина	Мобільний	68	84	96	100
Адміністративна частина	Комп'ютер	98	100	96	100
Адміністративна частина	Мобільний	81	100	96	100

Отримані результати свідчать про високу загальну продуктивність як клієнтської, так і адміністративної частин інтернет-магазину Lorena. Особливо відзначається адміністративна панель, яка демонструє майже максимальні показники на всіх пристроях. Клієнтська частина також має добрі результати, хоча ефективність на мобільних пристроях дещо нижча, що може бути пов'язано зі складністю візуальних елементів або вагою медіаконтенту. Оптимізація ресурсів, ліниве завантаження зображень, а також використання серверного рендерингу (SSR) у Next.js позитивно вплинули на загальну швидкість завантаження сторінок.

Таким чином, інтернет-магазин Lorena відповідає сучасним технічним стандартам і забезпечує стабільну роботу в різних браузерях і на різних пристроях.

3.3 Підтримка та подальший розвиток інтернет-магазину Lorena

Після завершення розробки інтернет-магазину Lorena необхідним наступним етапом є його підготовка до повноцінного запуску, що передбачає наповнення платформи актуальними товарами з відповідними зображеннями, описами, цінами, тегами та варіаціями. Лише після цього користувачі зможуть ознайомитися з асортиментом та здійснювати покупки.

Для забезпечення довготривалого функціонування та конкурентоспроможності ресурсу доцільно розробити план технічної підтримки, який включає оновлення залежностей, контроль за стабільністю роботи серверної частини, відстеження помилок та зворотного зв'язку користувачів. Окрему увагу варто приділити SEO-оптимізації: впровадженню метаданих, покращенню структури контенту, швидкості завантаження та мобільної адаптивності. Це сприятиме підвищенню позицій сайту в результатах пошукових систем і залученню нових клієнтів.

Також перспективним напрямом розвитку є розробка маркетингової стратегії: запуск рекламних кампаній у соціальних мережах, налаштування email-розсилок, використання Google Ads і аналітики користувацької поведінки. У подальшому можливе масштабування проєкту за рахунок інтеграції нових функцій, розширення на інші країни та введення мультивалютності або мультимовності.

3.4 Висновок до третього розділу

У третьому розділі було розглянуто практичну реалізацію інтернет-магазину Lorena, починаючи з його розгортання у веб-середовищі. Це

дозволило протестувати всі компоненти магазину в умовах, наближених до реального використання, та переконатися в їх коректному функціонуванні.

Особливу увагу приділено валідації інтерфейсу, форм введення та користувацьких сценаріїв. Проведено перевірку кросбраузерної сумісності магазину, а також здійснено тестування швидкості завантаження сторінок, що є критично важливим для забезпечення позитивного користувацького досвіду.

Насамкінець, розглянуто питання підтримки та подальшого розвитку інтернет-магазину. Запропоновані напрями масштабування та оптимізації функціоналу дозволяють забезпечити його довготривалу ефективність і адаптивність до змін ринку.

Результати, отримані в цьому розділі, підтверджують успішну реалізацію поставлених технічних завдань і готовність проєкту до практичного використання.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Психологічні чинники небезпеки

Психологічні чинники становлять важливу складову в системі причин виникнення небезпек, як у професійній діяльності, так і в повсякденному житті. За оцінками експертів з безпеки життєдіяльності, до 90% нещасних випадків є наслідком людського чинника – тобто дій чи бездіяльності самої особи в умовах потенційної загрози [41]. Поведінка людини, її психічний та емоційний стан, здатність до концентрації, швидкість прийняття рішень і рівень усвідомлення ризику мають безпосередній вплив на ймовірність виникнення небезпечної ситуації [42].

Усі психологічні чинники, що можуть сприяти формуванню небезпеки, умовно поділяються на стійкі (довготривалі, хронічні) та тимчасові (ситуаційні, епізодичні) [43].

До стійких психологічних чинників небезпеки належать:

- 1) хронічні захворювання, що негативно впливають на психофізичний стан особи, знижують увагу та працездатність;
- 2) порушення сенсорних функцій, які ускладнюють адекватне сприйняття сигналів небезпеки;
- 3) координаційні порушення – проблеми з точністю та узгодженістю рухів, що можуть спричинити помилки у виконанні завдань;
- 4) емоційна нестабільність – підвищена тривожність, дратівливість або апатія, які порушують адекватне реагування в критичних ситуаціях;
- 5) шкідливі звички та залежності, що погіршують когнітивні функції, притуплюють пильність та відповідальність;
- 6) психоемоційне вигорання або незадоволеність трудовою діяльністю, що веде до втрати мотивації, ігнорування інструкцій з охорони праці та безпечної поведінки.

До тимчасових психологічних чинників небезпеки відносять:

- 1) недосвідченість або погана обізнаність із процедурою виконання завдання, що призводить до невірної оцінки ситуації та неправильних дій;
- 2) необережність і легковажність, які проявляються в ігноруванні інструкцій, невикористанні засобів індивідуального захисту, нехтуванні потенційними ризиками;
- 3) фізична та психічна втома, що виникає внаслідок перевантаження, недосипання або монотонної роботи, значно знижує концентрацію уваги, швидкість реакції та точність дій;
- 4) стресові стани, які викликані емоційним перенапруженням, конфліктами або тиском з боку керівництва, призводять до втрати контролю над поведінкою та прийняттям необґрунтованих рішень.

Психологічна природа ризику полягає у складній взаємодії зовнішніх факторів з індивідуальними психофізіологічними особливостями особистості [44]. Саме тому особлива увага має приділятися формуванню психологічної культури безпеки, що включає:

- розвиток самодисципліни та здатності до самоконтролю;
- усвідомлення наслідків ризикованої поведінки;
- навички конструктивного реагування у стресових ситуаціях;
- стабільну мотивацію до дотримання правил безпеки;
- відповідальність за власне життя та безпеку оточення.

Упровадження тренінгів з психологічної підготовки, профілактика стресу, контроль психоемоційного стану працівників, а також ефективна система адаптації нових працівників – усе це є необхідними елементами зменшення впливу психологічних чинників небезпеки [45].

Розуміння психологічних механізмів формування небезпек дозволяє не лише ефективніше запобігати нещасним випадкам, але й формувати свідоме ставлення до особистої безпеки як важливої складової професійної та життєвої компетентності.

4.2 Загальні вимоги безпеки з охорони праці для користувачів ПК

У сучасному інформаційному суспільстві робота з персональним комп'ютером (ПК) стала невід'ємною частиною трудового процесу у багатьох сферах діяльності. Водночас така діяльність супроводжується низкою потенційних небезпек, пов'язаних з тривалим впливом на організм людини несприятливих факторів – електромагнітного випромінювання, статичної пози, зорового навантаження, психоемоційного напруження тощо [46]. Саме тому розроблено та законодавчо затверджено низку вимог охорони праці, яких необхідно дотримуватись користувачам ПК.

Основні фактори ризику при роботі за ПК:

- 1) тривале статичне сидіння, що може призвести до порушень опорно-рухового апарату;
- 2) навантаження на зорову систему, спричинене яскравим або бликуючим екраном, невірним розташуванням монітора;
- 3) психоемоційне перевантаження, яке виникає через монотонність або надмірну інтенсивність роботи.

Організація робочого місця користувача ПК здійснюється відповідно до чинних стандартів і нормативних документів, які регламентують вимоги до ергономіки, безпеки та умов праці [47, 48, 49, 50].

З метою забезпечення безпечних умов праці під час роботи з персональним комп'ютером важливу роль відіграє правильна організація робочого місця. Зокрема, розміщення обладнання має відповідати ергономічним вимогам згідно з ДСТУ 8604:2015 [47].

У відповідності до стандарту, стіл і крісло повинні бути підібрані з урахуванням зросту працівника, що дозволяє уникнути зайвого навантаження на хребет і м'язово-скелетну систему. Монітор доцільно розміщувати на рівні очей або трохи нижче, на відстані не менш ніж 50–70 см від користувача, що дозволяє знизити напруження зорового апарату. Особливу увагу слід приділити

освітленню: воно має бути м'яким, комбінованим (природне та штучне) і не створювати прямих відблисків на поверхні екрана.

Для попередження перевтоми очей та нервової системи рекомендується чергувати періоди роботи з короткими перервами. Після кожної години безперервної роботи необхідно зробити паузу на 10–15 хвилин, під час якої бажано виконати прості фізичні вправи або гімнастику для очей [51]. Такі перерви сприяють зменшенню м'язового напруження, покращують кровообіг та знижують ризик розвитку професійних захворювань.

Комфорт користувача значною мірою залежить від якості використовуваних меблів і обладнання. Ергономічне крісло з регульованою висотою і наявністю підлокітників дозволяє зберігати правильну поставу протягом усього робочого часу. Використання антивідблискових фільтрів або спеціального покриття на екрані допомагає зменшити вплив яскравого світла на очі.

Важливо також підтримувати належний мікроклімат у приміщенні – температура має бути комфортною, а прибирання проводитись регулярно для запобігання скупченню пилу, який негативно впливає на дихальну систему.

Особливу увагу слід приділяти й дотриманню вимог електробезпеки. Усі електроприлади мають бути технічно справними, заземленими та підключеними до мережі через сертифіковані пристрої [52]. Категорично забороняється використання пошкоджених подовжувачів чи розеток, адже це може призвести до короткого замикання чи ураження електричним струмом. У кожному робочому приміщенні повинні бути передбачені засоби первинного пожежогасіння – зокрема вогнегасники, а працівники зобов'язані знати порядок дій на випадок надзвичайної ситуації.

У межах профілактичної роботи особи, які проводять значну частину робочого часу за комп'ютером, мають регулярно проходити медичні огляди. Це дає змогу вчасно виявити порушення зору, опорно-рухової системи або нервово-психічного стану. Для зменшення стресового навантаження і підвищення ефективності праці в організаціях запроваджуються заходи

психологічного розвантаження: тренінги, консультації, спеціально облаштовані зони відпочинку тощо.

Дотримання вимог охорони праці під час роботи з ПК не лише забезпечує безпеку працівника, а й сприяє збереженню його фізичного та психоемоційного здоров'я у довгостроковій перспективі.

4.3 Висновок до четвертого розділу

У четвертому розділі розглянуто основні проблеми безпеки життєдіяльності та охорони праці при роботі з ПК, що є актуальними в умовах зростаючої цифровізації трудових процесів. Наведено аналіз психологічних чинників небезпеки, дав змогу усвідомити, що інформаційне перевантаження, високі когнітивні навантаження, а також тривала ізоляція в робочому середовищі можуть мати серйозні наслідки для психоемоційного стану користувачів. Зниження працездатності, хронічна втома, синдром професійного вигорання – це лише деякі з ризиків, які потребують належної профілактики та управління.

В розділі представлено загальні вимоги безпеки при експлуатації комп'ютерної техніки. Встановлено, що правильна організація робочого місця, дотримання ергономічних норм, регламентовані перерви в роботі, підтримання комфортного мікроклімату в приміщенні та дотримання електробезпеки є необхідними умовами для здоров'я користувача. Необхідно зазначити, що важливу роль відіграє формування культури безпеки, що включає як індивідуальну відповідальність за дотримання правил, так і системні дії з боку роботодавця.

Комплексний підхід до організації безпечного цифрового робочого середовища дозволяє мінімізувати ризики професійних захворювань, зберегти здоров'я працівників та забезпечити стабільну продуктивність праці в умовах інформаційного суспільства.

ВИСНОВКИ

У першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр» подано характеристику предметної області інтернет-комерції у сфері косметичних товарів, розглянуто актуальні тенденції розвитку цього сегмента ринку та висвітлено типові вимоги до сучасних e-commerce платформ, включаючи зручність користування, безпеку, мобільну адаптивність, персоналізацію контенту, інтеграцію з платіжними системами та ефективне управління товарними позиціями. Проаналізовано важливість адміністративної панелі як ключового інструменту для управління онлайн-магазином. Також здійснено огляд існуючих рішень на ринку, що дозволило сформулювати власні вимоги до функціоналу та дизайну інтернет-магазину Lorena. Обґрунтовано вибір технологій розробки, таких як Next.js, Tailwind CSS, MongoDB, Stripe і Clerk, відповідно до поставлених цілей та потреб проєкту.

У другому розділі досліджено архітектуру вебзастосунку та структуру клієнтської і адміністративної частин, обґрунтовано вибір технологічного стеку та реалізовано моделювання бази даних. Сформовано ключові функціональні можливості платформи, включаючи реєстрацію та авторизацію, перегляд товарів, роботу з кошиком, обробку замовлень, а також інструменти для адміністрування колекцій, клієнтів і товарних позицій. Особливу увагу приділено розробці адаптивного та естетично привабливого інтерфейсу із застосуванням Tailwind CSS, що забезпечує позитивний досвід користувача.

У третьому розділі розглянуто практичну реалізацію інтернет-магазину Lorena та описано процес його розгортання у веб-середовищі. Проведено тестування усіх функціональних компонентів в умовах, наближених до реального використання. Значну увагу зосереджено на валідації форм, перевірці користувацьких сценаріїв, кросбраузерній сумісності та оптимізації швидкості завантаження сторінок. Для цього було використано Google PageSpeed Insights, що дозволило оцінити ефективність застосованих рішень і підтвердити

відповідність ресурсу сучасним вимогам продуктивності. Окрім того, окреслено напрями подальшого розвитку та масштабування застосування, що забезпечує його адаптивність до змін ринку й стабільну експлуатацію в майбутньому.

У розділі «Безпека життєдіяльності, основи охорони праці» висвітлено ключові вимоги до організації безпечного робочого місця розробника, зокрема ергономіку, електробезпеку, мікроклімат та режим праці. Також розглянуто вплив інформаційного навантаження та стресу на психоемоційний стан, що підкреслює важливість дотримання заходів охорони праці в ІТ-сфері.

ПЕРЕЛІК ДЖЕРЕЛ

1 Online Cosmetics Market Size & Share Analysis - Growth Trends & Forecasts (2025 - 2030). Mordor Intelligence. [Електронний ресурс]. Режим доступу до ресурсу: <https://www.mordorintelligence.com/industry-reports/online-cosmetics-market?> (дата звернення: 19.05.2025).

2 The Beauty Category's Dynamic Growth: A 2024 Mid-Year Update. NIQ. [Електронний ресурс]. 09.08.2024. Режим доступу до ресурсу: <https://nielseniq.com/global/en/insights/commentary/2024/beauty-2024-mid-year-update/?> (дата звернення: 19.05.2025).

3 Струтинська, І. В., Дмитроца, Л. П., Сороківська, О. А., & Козбур, Г. В. (2024). Особливості цифрового розвитку малого і середнього бізнесу України, країн Європи та G7. In Трансформація бізнесу для сталого майбутнього: дослідження, цифровізація та інновації: монографія (pp. 411-427). : ФОП Паляниця В. А.

4 Beauty E-commerce Trends 2024: A Comprehensive Guide for Founders and Marketing Leaders. Peel. [Електронний ресурс]. 24.02.2025. Режим доступу до ресурсу: <https://www.peelinsights.com/post/beauty-e-commerce-trends-2024-a-comprehensive-guide-for-founders-and-marketing-leaders?> (дата звернення: 19.05.2025).

5 Струтинська, І. В., Дмитроца, Л. П., Козбур, Г. В., & Дмитрук, У. І. (2020). Ключові фактори, що сприяють розвитку цифрової економіки. Тези доповідей міжнародної науково-практичної конференції „Цифрова економіка як фактор інноваційного розвитку суспільства“, 43-45.

6 Струтинська, І. В., & Мельник, В. О. (2022). Роль автоматизованих систем керування замовленнями. Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 133-134.

7 Moskvichova A. 30 Major Requirements for Ecommerce Websites: Checklist to Start a Store. Digitalsuits. [Електронний ресурс]. 20.08.2024. Режим

доступу до ресурсу: <https://digitalsuits.co/blog/10-major-requirements-for-e-commerce-website/>? (дата звернення: 19.05.2025).

8 Shah P. E-Commerce System Requirements: A Comprehensive Guide. DhiWise. [Електронний ресурс]. 25.03.2025. Режим доступу до ресурсу: <https://www.dhiwise.com/blog/requirement-builder/ecommerce-system-requirements> (дата звернення: 19.05.2025).

9 Marzal R. Key Metrics to Track on Your Ecommerce Dashboard. NextScenario. [Електронний ресурс]. Режим доступу до ресурсу: <https://nextscenario.com/ecommerce-dashboard/> (дата звернення: 19.05.2025).

10 Shvetsova Y. Functional and Non-Functional Requirements of Online Shopping System: List & Examples. Elogic. [Електронний ресурс]. 08.06.2023. Режим доступу до ресурсу: <https://elogic.co/blog/functional-and-non-functional-requirements-for-ecommerce-websites/>? (дата звернення: 19.05.2025).

11 Pattamatta R. Ecommerce Admin Dashboard: Key Features, Benefits, and Best Practices. Databrain. [Електронний ресурс]. 08.05.2025. Режим доступу до ресурсу: <https://www.usedatabrain.com/blog/ecommerce-admin-dashboard?> (дата звернення: 19.05.2025).

12 Alferez A. Unveiling the Power of Admin Panels: A Comprehensive Guide. Ecommerce Next. [Електронний ресурс]. 26.08.2024. Режим доступу до ресурсу: <https://www.ecommercenext.org/unveiling-the-power-of-admin-panels-a-comprehensive-guide/> (дата звернення: 19.05.2025).

13 Kirvan P. What is a Waterfall model? Definition and guide. TechTarget. [Електронний ресурс]. 24.11.2024. Режим доступу до ресурсу: <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model> (дата звернення: 24.05.2025).

14 Jaja D. Routing in Next.js – How to Use App Router in your Next Apps. FreeCodeCamp. [Електронний ресурс]. 24.08.2023. Режим доступу до ресурсу: <https://www.freecodecamp.org/news/routing-in-nextjs/> (дата звернення: 24.05.2025).

15 Garg R. Next.js: The Ultimate Guide to SSR and SSG in React. Medium. [Електронний ресурс]. 07.07.2023. Режим доступу до ресурсу: <https://medium.com/womenintechnology/next-js-the-ultimate-guide-to-ssr-and-ssg-in-react-245598d765c3> (дата звернення: 24.05.2025).

16 Прус О. Next.js чи Angular: який фреймворк обрати замовнику для розробки?. Wezom. [Електронний ресурс]. 11.11.2024. Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/author/aleksandr-prus> (дата звернення: 24.05.2025).

17 Wili Rahmat M. How to deploy Next.js to Vercel with CI/CD Integration. Medium. [Електронний ресурс]. 12.05.2023. URL: <https://medium.com/@wilirahmatm/how-to-deploy-next-js-to-vercel-with-ci-cd-integration-c2645cb77ff0> (дата звернення: 24.05.2025).

18 Готович, В. А., & Мачужак, А. В. (2022). Застосування методології CI/CD для автоматизації процесів тестування та розгортання програмного забезпечення. Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 131-132.

19 Sytnyk D. Використання CORS в Next.js для обробки cross-origin запитів. Medium. [Електронний ресурс]. 09.07.2023. Режим доступу до ресурсу: <https://medium.com/@dmitry.sytnyk/%D0%B2%D0%B8%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D0%BD%D1%8F-cors-%D0%B2-next-js-%D0%B4%D0%BB%D1%8F-%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B8-cross-origin-%D0%B7%D0%B0%D0%BF%D0%B8%D1%82%D1%96%D0%B2-d8fe2f8eae8a> (дата звернення: 24.05.2025).

20 Hernandez I. Tailwind Vs. Bootstrap: Який CSS фреймворк вам потрібен?. DreamHost. [Електронний ресурс]. 29.05.2024. Режим доступу до ресурсу: <https://www.dreamhost.com/blog/uk/tailwind-proti-bootstrap/> (дата звернення: 24.05.2025).

21 Introduction to Tailwind CSS. GeeksForGeeks. [Електронний ресурс]. 07.10.2024. Режим доступу до ресурсу:

<https://www.geeksforgeeks.org/introduction-to-tailwind-css/> (дата звернення: 24.05.2025).

22 Тарас Х. Дизайнерські компетенції програміста у контексті використання Tailwind CSS. VIII Міжнародна студентська науково-технічна конференція «Природничі та гуманітарні науки. Актуальні питання», Тернопіль. Тернопільський національний технічний університет ім. І.Пулюя (м.Тернопіль, 24–25 квіт. 2025 р.), 2025. С. 204–206.

23 Bhaskar V. Tailwind CSS Best Practices for Performance Optimization. TailGrids. [Електронний ресурс]. 09.04.2025. Режим доступу до ресурсу: <https://tailgrids.com/blog/tailwind-css-best-practices-and-performance-optimization> (дата звернення: 24.05.2025).

24 Singh N. Understanding the Variances Between JSON and BSON in MongoDB. Medium. [Електронний ресурс]. 29.12.2023. Режим доступу до ресурсу: <https://medium.com/@navneetskahlon/understanding-the-variances-between-json-and-bson-in-mongodb-258b39c169f7> (дата звернення: 24.05.2025).

25 Kristbergsson E. S. The MongoDB and Next.js Stack for Beginners. Medium. [Електронний ресурс]. 09.01.2024. Режим доступу до ресурсу: <https://medium.com/stackademic/the-mongodb-and-next-js-stack-for-beginners-b4394872f5bb> (дата звернення: 24.05.2025).

26 Prisma ORM MongoDB database connector. Prisma /docs. [Електронний ресурс]. Режим доступу до ресурсу: <https://www.prisma.io/docs/orm/overview/databases/mongodb> (дата звернення: 24.05.2025).

27 Ramanayake B. What you need to know about MongoDB Atlas ?. Medium. [Електронний ресурс]. 08.07.2023. Режим доступу до ресурсу: <https://medium.com/@bdhanushka65/what-you-need-to-know-about-mongodb-atlas-b4743727e7f1> (дата звернення: 24.05.2025).

28 Stripe: найдорожча фінтех-компанія у Кремнієвій долині. Blog.imena.ua. [Електронний ресурс]. 05.06.2024. Режим доступу до ресурсу: <https://www.imena.ua/blog/stripe/> (дата звернення: 08.05.2025).

29 Security at Stripe. Stripe Docs. [Електронний ресурс]. Режим доступу до ресурсу: <https://docs.stripe.com/security> (дата звернення: 24.05.2025).

30 Nyabuto, M. G. M., Mony, V., & Mbugua, S. (2024). Architectural review of client-server models. *International journal of scientific research and engineering trends*, 10(1), 139-143.

31 Ansari K. Next.js App Router: Routing. Medium. [Електронний ресурс]. 19.04.2024. Режим доступу до ресурсу: <https://medium.com/@kulsumansari4/next-js-app-router-routing-8d795dbe324c> (дата звернення: 15.05.2025).

32 Salnik R. Опановуємо CRUD в Next.js. Поради для інтеграції та оптимізації. DOU.UA. [Електронний ресурс]. 17.12.2024. Режим доступу до ресурсу: <https://dou.ua/forums/topic/51587/> (дата звернення: 24.05.2025).

33 Колесніков Д. Що таке фреймворк: пояснюємо простими словами. Brainlab. [Електронний ресурс]. 02.03.2023. Режим доступу до ресурсу: <https://brainlab.com.ua/uk/blog-uk/shho-take-frejmwork-poyasnyuyemo-prostymu-slovamy> (дата звернення: 24.05.2025).

34 Колесніков Д. Що таке фреймворк: пояснюємо простими словами. Brainlab. [Електронний ресурс]. 02.03.2023. Режим доступу до ресурсу: <https://brainlab.com.ua/uk/blog-uk/shho-take-frejmwork-poyasnyuyemo-prostymu-slovamy> (дата звернення: 24.05.2025).

35 Rashkovits, R., & Lavy, I. (2021). Mapping common errors in entity relationship diagram design of novice designers. *International Journal of Database Management Systems*, 13(1), 1-19.

36 Shwetha, H., Prajwal, D., & Sridharan, S. (2024, April). From MongoDB to React: Unleashing the Power of MERN in E-commerce. In *2024 International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS)* (pp. 1-6). IEEE.

37 <UIButton /> component. Clerk Docs. [Електронний ресурс]. Режим доступу до ресурсу: <https://clerk.com/docs/components/user/user-button> (дата звернення: 16.05.2025).

38 Nousiainen, A. (2024). The potential of WebAssembly in Edge Computing.

39 Gulshan N. Streamlining Your Code: Top Online HTML Validators for Web Developers. Medium. [Електронний ресурс]. 18.09.2023. Режим доступу до ресурсу: https://medium.com/@gulshan_n/streamlining-your-code-top-online-html-validators-for-web-developers-49182f1e2d5c (дата звернення: 21.05.2025).

40 Дмитроца, Л. П., & Старицький, О. Т. (2024). Оптимізація продуктивності та SEO при масштабуванні проєктів на основі React і Next.js. Збірник тез доповідей XIII Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 465-466.

41 Безпека життєдіяльності: підручник / за ред. В. Г. Беспалька. – К.: Центр учбової літератури, 2012. – 384 с.

42 Золотарьова І. М. Психологія безпеки: навч. посіб. – Харків: НУЦЗУ, 2018. – 164 с.

43 Козяр М. М. Психологічні основи формування безпечної поведінки людини / М. М. Козяр // Вісник ЛНУ ім. Івана Франка. Серія: Психологія. – 2019. – № 6. – С. 112–118.

44 Яковлєв Є. О. Людський фактор в системі безпеки праці / Є. О. Яковлєв // Науковий вісник УкрДУЗТ. – 2020. – № 192. – С. 85–91.

45 Стельмах Л. В. Психологічні аспекти небезпеки в професійній діяльності / Л. В. Стельмах // Безпека життєдіяльності. – 2017. – № 2. – С. 21–25.

46 Гелетій М. М. Основи охорони праці: підручник / М. М. Гелетій, М. П. Лапчук, І. М. Гнатюк. – Львів: Новий Світ – 2000, 2020. – 352 с.

47 ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. – [Чинний від 01.07.2017]. – К.: ДП «УкрНДНЦ», 2017. – 10 с.

48 ДСТУ EN ISO 9241-5:2018. Ергономіка взаємодії людини і системи. Частина 5. Вимоги до робочих місць із дисплеями інформації. – К. : ДП «УкрНДНЦ», 2018. – 28 с.

49 ДСТУ 4196:2003. Освітлення робочих місць. Норми та методи вимірювань. – К. : ДП «УкрНДНЦ», 2003. – 22 с.

50 НПАОП 0.00-4.10-18. Організація робочих місць: загальні положення. – К., 2018. – 40 с.

51 Литвиненко О. М. Ергономіка та охорона праці при роботі з ПК / О. М. Литвиненко. – Харків: ХНАМГ, 2020. – 180 с.

52 Правила улаштування електроустановок (ПУЕ). – Харків: Видавництво „Форт,, 2017. 760 с.

ДОДАТКИ

**Тези VIII Міжнародної студентської науково - технічної конференції
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ
ПИТАННЯ"**

Міністерство освіти і науки України
Тернопільський національний технічний університет
імені Івана Пулюя
Маріборський університет (Словенія)
Технічний університет в Кошице (Словаччина)
Каунаський технологічний університет (Литва)
Львівський національний університет
імені Івана Франка,
Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)
Луцький національний технічний університет,
Чернівецький національний університет
імені Юрія Федьковича,
Вроцлавський економічний університет (Польща)
Університет технологій та економіки
імені Хелени Ходковської (Польща)
Донбаська державна машинобудівна академія



*Студентське наукове
товариство*



**VIII МІЖНАРОДНА
студентська науково - технічна конференція**

**"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ
НАУКИ.**

АКТУАЛЬНІ ПИТАННЯ"

24-25 квітня 2025 р.

(збірник тез конференції)

Тернопіль 2025

ББК 72+34 (Укр)
М34

Матеріали VIII Міжнародної студентської науково - технічної конференції
/ Тернопіль: Тернопільський національний технічний університет ім.
І.Пулюя (м. Тернопіль, 24-25 квітня 2025 р.), 2025.- 391 с.

УДК 004.8

Тарас Х. - ст. гр. СН-43

Тернопільський національний технічний університет імені Івана Пулюя

ДИЗАЙНЕРСЬКІ КОМПЕТЕНЦІЇ ПРОГРАМІСТА У КОНТЕКСТІ ВИКОРИСТАННЯ TAILWIND CSS

Науковий керівник: к.т.н, доцент Дмитроца Л.П.

Taras K.

Ternopil Ivan Puluj National Technical University

DESIGN COMPETENCES OF A PROGRAMMER IN THE CONTEXT OF USING TAILWIND CSS

Supervisor: PhD, Associate Professor Dmytrotsa L.P.

Ключові слова: вебінтерфейс, фронтенд-розробка, UI/UX дизайн

Keywords: web interface, front-end development, UI/UX design

У сучасному світі веброзробки все частіше стираються межі між ролями, які ще донедавна вважалися окремими професійними спеціалізаціями. Програміст більше не є виключно технічним виконавцем, а дизайнер – не лише творцем естетичного вигляду інтерфейсів. У нових умовах ринку інформаційних технологій дедалі більшу цінність набувають фахівці, які здатні поєднувати аналітичне мислення з візуальною грамотністю, створюючи не лише функціональні, а й візуально привабливі цифрові продукти.

Фронтенд-розробка переживає активну трансформацію, що зумовлена зростанням вимог до користувацького досвіду та естетики вебінтерфейсів. Якщо раніше розробник здебільшого відповідав за функціональність і коректну роботу елементів інтерфейсу, то сьогодні його зона відповідальності значно ширша. Користувачі очікують не просто працюючий сайт, а привабливе, інтуїтивно зрозуміле середовище, у якому взаємодія з елементами інтерфейсу відбувається природно і легко. Це, у свою чергу, вимагає від фронтенд-розробника не лише знання мов програмування, бібліотек і фреймворків, а й базових навичок дизайну та розуміння принципів UI/UX.

Зміна ринку праці підтверджує цю тенденцію [1]. У більшості вакансій на позицію frontend developer серед обов'язкових або бажаних вимог зазначаються знання системи дизайну, досвід роботи з Figma, Adobe XD, а також розуміння принципів побудови інтерфейсів. Такий підхід обумовлений необхідністю оптимізувати командну роботу, зменшити кількість ітерацій при погодженні макетів і спростити адаптацію дизайну під різні пристрої та екрани.

Крім того, в умовах зростання кількості стартапів та невеликих проєктів, де часто немає окремих дизайнерів, саме на програміста покладається обов'язок створення візуальної частини продукту. У таких випадках його здатність до естетичного мислення, чуття пропорцій, контрасту, ритму та гармонії стає не менш важливою, ніж технічна грамотність. Таким чином, сучасний фронтенд – це синтез технологій і дизайну, де програміст дедалі частіше виконує роль повноцінного творця інтерфейсу.

З-поміж сучасних CSS-інструментів особливе місце займає Tailwind CSS – фреймворк, що реалізує концепцію utility-first [2], за якої розробник оперує

низькорівневими класами без необхідності створювати кастомні стилі. Унікальність Tailwind CSS полягає в тому, що він дозволяє будувати інтерфейс безпосередньо в HTML-структурі, використовуючи зрозумілі, логічно названі класи. Це робить його надзвичайно привабливим для програмістів, які не мають професійного дизайнерського досвіду.

Фреймворк орієнтований на швидкість і гнучкість. Замість того, щоб писати власні стилі або адаптовувати готові компоненти з дизайн-систем, розробник може «збирати» інтерфейс із будівельних блоків, орієнтуючись на технічну логіку. Tailwind забезпечує точний контроль над кожним елементом, а також передбачає систему кольорових палітр, відступів, розмірів і шрифтів, які вже оптимізовані для зручного візуального сприйняття.

Завдяки Tailwind CSS програмісти отримали інструмент, який не лише пришвидшує процес розробки, а й дозволяє створювати візуально якісні інтерфейси навіть без глибокого занурення у дизайнерські концепції. Tailwind значно знижує бар'єр входу у візуальну частину розробки, пропонуючи певну «дизайн-підказку» через систему класів, що базуються на перевірених UI-практиках.

Однак така уніфікована система не гарантує створення продуманого й ефективного дизайну. Tailwind CSS – це лише інструмент, і те, наскільки вдало він буде використаний, залежить саме від компетенцій розробника, зокрема його вміння бачити інтерфейс не лише як набір блоків, а як цілісний продукт із власною логікою взаємодії, естетикою та користувацькою цінністю.

Попри функціональність і зручність Tailwind CSS, його використання не є гарантією створення якісного та ефективного дизайну. Фреймворк надає набір готових інструментів, але не замінює дизайнерське бачення. Без розуміння основ композиції, принципів візуальної ієрархії, типографіки, балансу кольору та ритму інтерфейс, зібраний за допомогою Tailwind, ризикує залишитися лише технічно коректним, але візуально слабким або навіть незручним для користувача.

Програміст, який не володіє дизайнерськими навичками, може легко створити перевантажений або дезорганізований інтерфейс, де важливі елементи губляться, а користувачеві важко зорієнтуватися.

Наприклад, недотримання контрастності може ускладнити читання тексту, а надлишок декоративних елементів – відволікти від основної функції сторінки. У реальних проектах такі помилки можуть призвести до зниження зручності користування, погіршення досвіду користувача та втрати клієнтів.

Ще одним важливим аспектом є розуміння контексту взаємодії користувача з інтерфейсом. Tailwind не дає відповіді на питання, які дії очікуються від користувача, які елементи повинні бути помітними в першу чергу або як адаптувати інтерфейс до різних сценаріїв використання. Ці питання належать до сфери UX-дизайну [3], який виходить за межі лише візуальної стилізації. Без урахування таких аспектів навіть естетично вивірений інтерфейс не зможе забезпечити зручну та ефективну взаємодію з користувачем.

Таким чином, Tailwind CSS може стати потужним інструментом у руках програміста, але лише за наявності хоча б базових дизайнерських компетенцій. Без них використання фреймворку перетворюється на механічне компонування елементів, позбавлене логіки, глибини і стратегічного бачення інтерфейсу як цілісної системи.

Здатність програміста мислити візуально й володіти основами дизайну відкриває перед ним нові можливості та переваги в професійному середовищі.

По-перше, розробник із дизайнерським мисленням краще розуміє потреби користувача. Він здатен проєктувати інтерфейс не лише з погляду функціональності, а й з урахуванням логіки взаємодії, зручності сприйняття інформації та естетичних

очікувань аудиторії. Це особливо важливо у фронтенд-розробці, де саме візуальний контакт є першим пунктом взаємодії з продуктом.

По-друге, дизайнерські навички програміста сприяють підвищенню його автономності. У невеликих командах або фріланс-проектах, де немає окремого UI/UX-дизайнера, вміння самостійно створювати макети, компоувати інтерфейс і приймати стилістичні рішення стає великою перевагою. Tailwind CSS у цьому випадку працює як ефективний інструмент реалізації візуальних ідей без потреби писати CSS з нуля.

По-третє, програмісти з дизайнерським досвідом мають кращу комунікацію з командами дизайнерів. Вони швидше розуміють вимоги до верстки, логіку адаптивного дизайну, специфіку кольорових схем та типографіки, а також можуть запропонувати адекватні технічні рішення для реалізації креативних ідей.

У результаті, програміст із дизайнерськими навичками перетворюється на універсального фахівця, здатного створювати повноцінні продукти – від логіки до візуальної оболонки. Tailwind CSS лише підсилює цей ефект, виступаючи зручним середовищем для реалізації дизайну «зсередини коду».

Висновок. У сучасній фронтенд-розробці, де інтерфейс є ключовим елементом взаємодії з користувачем, дизайнерські компетенції програміста набувають особливої ваги. Tailwind CSS значно спрощує процес створення інтерфейсів, але сам по собі не замінює дизайнерського мислення. Здатність програміста орієнтуватися в принципах візуальної композиції, кольорів, типографіки та логіки UX дозволяє не лише ефективніше використовувати Tailwind, а й створювати більш цілісні, зручні та естетично привабливі продукти. Таким чином, синергія технічних і дизайнерських навичок стає запорукою професійної гнучкості та конкурентоспроможності розробника у цифровій екосистемі.

Література

1. Скороход Д. Що має знати Senior Front-end Developer. Результати аналізу вакансій в Україні та Каліфорнії. DOU.UA. URL: <https://dou.ua/lenta/articles/knowledge-for-senior-front-end-developer/> (дата звернення: 11.04.2025).
2. Szymanski M. Utility-first approach. TW Elements. URL: <https://tw-elements.com/learn/te-foundations/tailwind-css/utility-first/> (дата звернення: 12.04.2025).
3. Stevens E. What Is User Experience (UX) Design? Everything You Need to Know. CareerFoundry. 22.01.2025. URL: <https://careerfoundry.com/en/blog/ux-design/what-is-user-experience-ux-design-everything-you-need-to-know-to-get-started/> (дата звернення: 12.04.2025).

Схеми Mongoose для баз даних lorena_store та lorena_admin**Лістинг Б.1 – Вміст файлу User.ts**

```
import mongoose from "mongoose";

const userSchema = new mongoose.Schema({
  clerkId: String,
  wishlist: {
    type: Array,
    default: []
  },
  createdAt: {
    type: Date,
    default: Date.now
  },
  updatedAt: {
    type: Date,
    default: Date.now
  }
});

const User = mongoose.models.User || mongoose.model("User",
userSchema);

export default User;
```

Лістинг Б.2 – Вміст файлу Collection.ts

```
import mongoose from "mongoose";

const collectionSchema = new mongoose.Schema({
  title: {
    type: String,
    required: true,
    unique: true,
  },
  description: String,
  image: {
    type: String,
    required: true,
  },
  products: [
    {
      type: mongoose.Schema.Types.ObjectId,
      ref: "Product",
    }
  ],
  createdAt: {
    type: Date,
    default: Date.now,
  },
  updatedAt: {
    type: Date,
```

```

        default: Date.now,
    }
})

const Collection =mongoose.models.Collection ||
mongoose.model("Collection", collectionSchema);

export default Collection;

```

Лістинг Б.3 – Вміст файлу Customer.ts

```

import mongoose from "mongoose";

const customerSchema = new mongoose.Schema({
  clerkId: String,
  name: String,
  email: String,
  orders: {
    type: [{ type: mongoose.Schema.Types.ObjectId, ref: "Order"}]
  },
  createdAt: {
    type: Date,
    default: Date.now,
  },
  updatedAt: {
    type: Date,
    default: Date.now,
  }
});

const Customer = mongoose.models.Customer ||
mongoose.model("Customer", customerSchema);

export default Customer;

```

Лістинг Б.4 – Вміст файлу Order.ts

```

import mongoose from "mongoose";

const orderSchema = new mongoose.Schema({
  customerClerkId: String,
  products: [
    {
      product: {
        type: mongoose.Schema.Types.ObjectId,
        ref: "Product",
      },
      color: String,
      size: String,
      quantity: Number,
    },
  ],
  shippingAddress: {
    street: String,
    city: String,
    state: String,
  },
});

```

```

        postalCode: String,
        country: String,
    },
    shippingRate: String,
    totalAmount: Number,
    createdAt: {
        type: Date,
        default: Date.now,
    },
});

const Order = mongoose.models.Order || mongoose.model("Order",
orderSchema);

export default Order;

```

Лістинг Б.5 – Вміст файлу Product.ts

```

import mongoose from "mongoose";

const ProductSchema = new mongoose.Schema({
    title: String,
    description: String,
    media: [String],
    category: String,
    collections: [{ type: mongoose.Schema.Types.ObjectId, ref:
"Collection" }],
    tags: [String],
    sizes: [String],
    colors: [String],
    price: { type: mongoose.Schema.Types.Decimal128, get: (v:
mongoose.Schema.Types.Decimal128) => { return
parseFloat(v.toString()) }},
    expense: { type: mongoose.Schema.Types.Decimal128, get: (v:
mongoose.Schema.Types.Decimal128) => { return
parseFloat(v.toString()) }},
    createdAt: { type: Date, default: Date.now },
    updatedAt: { type: Date, default: Date.now },
}, { toJSON: { getters: true } });

const Product = mongoose.models.Product ||
mongoose.model("Product", ProductSchema);
export default Product

```