

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Інтернет-магазин електроніки»

Виконав студент групи ПЗ-21-3
ПАВЛЮК Андрій Сергійович

Керівник роботи:
ЧИЖМОТРА Олексій Володимирович

Рецензент:
МОРОЗОВ Андрій Васильович

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії програмного
забезпечення

Тетяна ВАКАЛЮК

«14.02.2025»

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ПАВЛЮК Андрій Сергійович**

Керівник роботи: **ЧИЖМОТРА Олексій Володимирович**

Тема роботи: **«Інтернет-магазин електроніки»,**

затверджена Наказом закладу вищої освіти від **«14.02.2025» №61/с**

Вихідні дані для роботи: **процес створення інтернет-магазину як інформаційної системи електронної комерції, який використовує сучасні вебтехнології, адаптивний інтерфейс, базу даних та інтеграцію з платіжними системами для забезпечення зручного пошуку, замовлення та оплати товарів у режимі онлайн.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Чижмотря Олексій	14.02.2025	14.02.2025
2	Чижмотря Олексій	14.03.2025	14.03.2025
3	Чижмотря Олексій	14.04.2025	14.04.2025

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебдодатку інтернет-магазину та пояснювальної записки. Пояснювальна записка до випускної роботи містить 84 сторінки, 31 ілюстрацію та 9 таблиць.

Метою роботи є створення повнофункціонального інтернет-магазину з використанням сучасних вебтехнологій, який дозволяє користувачам здійснювати перегляд товарів, додавання до кошика, оформлення замовлення, оплату онлайн та взаємодію з системою доставки.

Архітектура інтернет-магазину реалізована з використанням фреймворку Next.js, мови TypeScript, модульної SCSS-стилізації та бази даних MongoDB. Було реалізовано компоненти для каталогів товарів, детальних сторінок, кошика, особистого кабінету користувача, підписки та сторінки оформлення замовлення. Платіжна інтеграція здійснена за допомогою платформи Stripe, а доставка – через API служб доставки.

Додаток має адаптивний інтерфейс, що забезпечує зручне користування з мобільних, планшетів та десктопів. Було також проведено тестування продуктивності за допомогою сервісів Google PageSpeed та GTMetrix, що підтвердило високий рівень ефективності роботи застосунку.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ-МАГАЗИН, КОШИК, ЗАМОВЛЕННЯ, КОРИСТУВАЧ.

					ІПЗ.КР.Б-121-25-ІПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.		А.С. Павлюк			Інтернет-магазин електроніки		Літ.	Арк.
Керівник		О.В. Чижмотря						3
								84
Рецензент		А.В. Морозов					Житомирська політехніка, група ІПЗ-21-3	
Зав. каф.		Т.А. Вакалюк						

ABSTRACT

The bachelor's qualification thesis consists of a web application for an online store and an explanatory report. The explanatory report comprises 84 pages, 31 illustrations, and 9 tables.

The aim of the thesis is to develop a fully functional online store using modern web technologies, which enables users to browse products, add them to the cart, place orders, make online payments, and interact with the delivery system.

The architecture of the online store is implemented using the Next.js framework, the TypeScript language, modular SCSS styling, and the MongoDB database. The application includes components for product catalogues, detailed product pages, shopping cart, user profile, newsletter subscription, and checkout page. Payment integration is implemented via the Stripe platform, and delivery services are connected through API integrations.

The application features a responsive user interface that ensures usability across mobile devices, tablets, and desktops. Performance testing was conducted using Google PageSpeed and GTMetrix services, confirming a high level of efficiency and responsiveness of the application.

KEYWORDS: ONLINE STORE, SHOPPING CART, CHECKOUT, USER.

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ІНТЕРНЕТ-МАГАЗИНІВ У СФЕРІ ЕЛЕКТРОННОЇ КОМЕРЦІЇ	9
1.1. ОГЛЯД СУЧАСНОГО СТАНУ РИНКУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ.....	9
1.2. Аналіз існуючих аналогів інтернет-магазинів: функціонал, переваги та недоліки.....	15
1.3. Обґрунтування вибору інструментальних засобів	21
1.4. Постановка технічного завдання	24
Висновки до першого розділу.	26
РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ	27
2.1. Визначення функціональних і нефункціональних вимог до платформи .	27
2.2. Розробка структури бази даних	29
2.3. Проектування та реалізація алгоритмів роботи системи	33
2.4. Реалізація функціоналу магазину	35
Висновки до другого розділу.....	38
РОЗДІЛ 3. ІНТЕРФЕЙС ТА КОРИСТУВАННЯ ІНТЕРНЕТ-МАГАЗИНОМ...	40
3.1. Користувацький інтерфейс і адаптивність верстки	40
3.2. Реалізація особистого кабінету, реєстрації та авторизації.....	46
3.3. Оформлення замовлення: кошик, оплата та інтеграція з API служб доставки	50
3.4. Тестування інтернет магазину	57
Висновки до третього розділу	63
ВИСНОВКИ	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТКИ.....	69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API (Application Programming Interface) – інтерфейс прикладного програмування, який дозволяє обмінюватися даними між різними системами.

UI (User Interface) – інтерфейс користувача.

UX (User Experience) – користувацький досвід при взаємодії з інтерфейсом.

B2C (Business to Customer) – модель електронної комерції, орієнтована на кінцевого споживача.

CRUD (Create, Read, Update, Delete) – основні операції з базами даних.

DB (Database) – база даних.

JSON (JavaScript Object Notation) – текстовий формат обміну даними між клієнтом і сервером.

SEO (Search Engine Optimization) – оптимізація вебсторінок для покращення їх видимості в пошукових системах.

HTTPS (HyperText Transfer Protocol Secure) – захищений протокол передавання гіпертексту.

CMS (Content Management System) – система керування вмістом сайту.

CDN (Content Delivery Network) – мережа доставки контенту, яка пришвидшує завантаження вебсторінок.

JWT (JSON Web Token) – формат токенів для безпечної передачі інформації та автентифікації.

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми полягає у зростаючій ролі електронної комерції в сучасному цифровому суспільстві. В умовах глобалізації та інтенсивного розвитку інформаційних технологій онлайн-торгівля стає невід’ємною частиною економічної діяльності підприємств. Інтернет-магазини є ефективним інструментом продажу товарів та послуг, що забезпечує зручність для користувачів, оперативність обробки замовлень і доступ до широкого ринку незалежно від географічного розташування.

Розвиток інтернет-торгівлі обумовлений зростанням попиту на дистанційні покупки, мобільність користувачів, постійне вдосконалення платіжних систем, інтеграцію служб доставки та можливість цілодобової роботи платформи. Інтернет-магазини дозволяють оптимізувати витрати підприємств, спрощують процес взаємодії з клієнтами та відкривають нові можливості для аналітики й персоналізації покупок.

Метою випускної кваліфікаційної роботи бакалавра є розробка функціонального інтернет-магазину як інформаційної системи електронної комерції з використанням сучасних вебтехнологій.

Для реалізації поставленої мети потрібно виконати наступні завдання:

1. Провести аналіз ринку інтернет-магазинів, дослідити функціональні вимоги та технічні рішення, які використовуються у сучасних системах електронної комерції.
2. Здійснити вибір технологій для реалізації інтернет-магазину, з урахуванням вимог до масштабованості, безпеки, зручності для користувача та інтеграції з платіжними й логістичними сервісами.
3. Реалізувати інтернет-магазин із можливістю перегляду товарів, формування кошика, оформлення замовлення, реєстрації користувачів.

Об’єктом дослідження є інформаційна система електронної комерції, що забезпечує функціонування інтернет-магазину.

Предметом дослідження є архітектура, функціональність і програмна

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

реалізація інтернет-магазину, включаючи базу даних товарів, модулі управління кошиком, оформленням замовлень, обліку клієнтів і механізми інтеграції з зовнішніми сервісами.

Дослідження також охоплює аналіз технологій веброзробки, зокрема використання Next.js, React, TypeScript, MongoDB, Stripe, з метою забезпечення зручності користування, швидкодії, адаптивності інтерфейсу та безпеки обробки персональних і платіжних даних.

Змн.	Арк.	№ докум.	Підпис	Дата

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ІНТЕРНЕТ- МАГАЗИНІВ У СФЕРІ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

1.1. Огляд сучасного стану ринку електронної комерції

Український ринок електронної комерції демонструє стабільне зростання, незважаючи на виклики, спричинені війною, нестабільністю економіки та порушенням логістичних ланцюгів. За підрахунками експертів, загальний обсяг eCommerce в Україні сягнув приблизно 5 мільярдів доларів США, що свідчить про поступове відновлення галузі після кризи 2022 року.

Варто зазначити, що до повномасштабного вторгнення ринок електронної торгівлі в Україні перебував у фазі активного росту, і навіть у воєнний період він зберіг потенціал до адаптації та інновацій. Онлайн-магазини змушені були переорієнтувати логістику, оптимізувати асортимент товарів, а також вдосконалити цифрові інструменти, щоб забезпечити безперебійне обслуговування клієнтів. Це стосується і нішевих магазинів, зокрема тих, що спеціалізуються на продажі електроніки – ноутбуків, смартфонів, комплектуючих тощо.

Однією з ключових тенденцій розвитку електронної комерції в Україні на 2024 рік є активне поширення мобільної торгівлі (m-commerce). Вже у 2023 році понад 60% інтернет-замовлень здійснювалося за допомогою смартфонів, і ця частка продовжує зростати. Основними факторами, що сприяють цьому процесу, є доступ до швидкого мобільного інтернету, зручність мобільних застосунків та широке використання бюджетних моделей смартфонів серед населення.

Впродовж 2024 року проникнення мобільного інтернету в Україні збільшиться з 71% до приблизно 75% (за даними Statista). У відповідь на це ритейлери спрямовують зусилля на розвиток мобільних платформ: створюють зручні додатки та адаптують свої сайти до потреб мобільних користувачів [42]. Таким чином, мобільна комерція стає не лише зручною альтернативою традиційним форматам онлайн-покупок, а й одним із найдинамічніших напрямів для масштабування бізнесу у сфері e-commerce (рис. 1.1) [42].

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

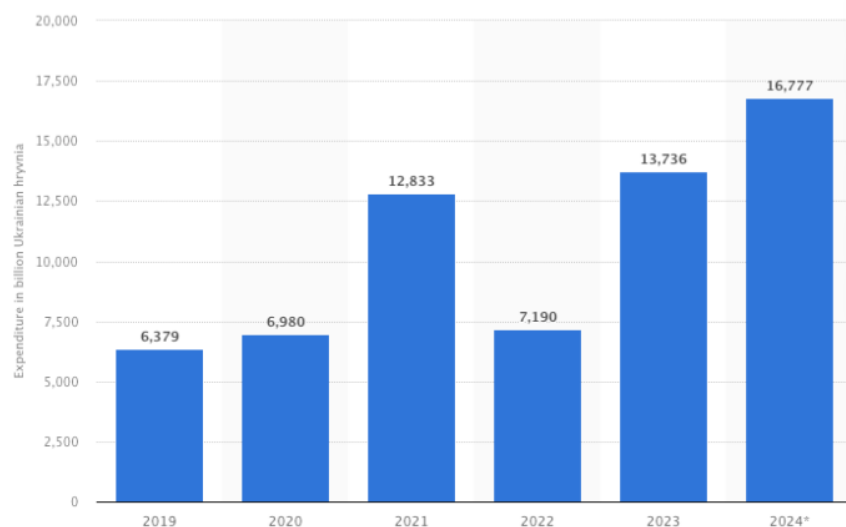


Рис.1.1 – Динаміка витрат на цифрову рекламу в Україні у 2019–2024 роках (млрд грн)

У 2024 році продажі через соціальні мережі займають важливе місце у стратегіях багатьох українських компаній. Такі майданчики, як Instagram, Facebook і TikTok, дозволяють бізнесам ефективно охоплювати молодіжну аудиторію, формуючи прямий контакт із клієнтом. Соціальна комерція набирає обертів завдяки зручним інструментам просування товарів, підтримці бренду та збільшенню довіри споживачів.

Особливу роль відіграє інфлюенс-маркетинг – співпраця з популярними блогерами та лідерами думок, які через рекомендації формують попит і перенаправляють аудиторію безпосередньо в онлайн-магазини. Додатково, підприємці експериментують з лайвстрим-продажами, тобто прямими ефірами з демонстрацією товарів. Такий формат, поширений в країнах Азії, поступово приживається і в Україні – особливо у сферах моди, косметики та аксесуарів, де інтерактивність та залучення мають ключове значення.

Швидкий перехід до цифрових способів оплати став потужним стимулом для зростання онлайн-торгівлі в Україні. За даними за 2023 рік, понад 80% усіх транзакцій у сфері e-commerce здійснювалися за допомогою електронних засобів оплати – включаючи банківські картки, мобільні додатки та сервіси типу Google Pay

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

та Apple Pay.

Українські користувачі активно користуються інтернет-банкінгом. Такі системи, як Monobank та Privat24, забезпечують миттєві платежі, грошові перекази, оплату рахунків і онлайн-покупки. Вони поєднують у собі функціональність для особистих та бізнес-цілей, підвищуючи зручність і швидкість фінансових операцій.

Крім того, швидко зростає популярність моделей “Купуй зараз – плати пізніше” (BNPL). Ці сервіси дозволяють розподіляти оплату за товар частинами, що особливо актуально під час придбання техніки, гаджетів і одягу. Такі гнучкі фінансові рішення сприяють зростанню лояльності покупців і збільшують обсяги продажів.

Проблеми із доставкою тривалий час стримували розвиток української електронної комерції, особливо у зв’язку з нестабільною інфраструктурою та воєнними діями. Проте у 2024 році акцент зосереджено на вдосконаленні логістичних процесів, щоб гарантувати надійне та оперативне отримання замовлень. Нова Пошта та Укрпошта значно розширили свої мережі навіть у складних регіонах, забезпечивши покриття більшої частини країни. Серед актуальних сервісів – доставка в день замовлення, поштові відділення самообслуговування та пункти видачі, які відповідають зростаючим очікуванням клієнтів щодо швидкості та зручності.

Крім традиційних каналів, з’являються інноваційні підходи до доставки. Серед них – використання дронів та автономного транспорту, що потенційно здатні розв’язати проблему останньої милі, особливо в густонаселених містах. Таким чином, ефективна логістика стає конкурентною перевагою для e-commerce бізнесу в Україні.

Таким чином, на основі аналізу поточного стану галузі можна виділити ключові тренди розвитку електронної комерції в Україні у 2024 році. Вони охоплюють як технологічні, так і поведінкові зміни у сфері онлайн-продажів [42].

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.1

Основні тренди українського e-commerce у 2024 році

№	Тренд	Короткий опис	Очікуваний вплив
1	Мобільна комерція (m-commerce)	Зростання покупок зі смартфонів, оптимізація сайтів та мобільних застосунків.	Понад 60% продажів – через мобільні пристрої.
2	Торгівля через соцмережі	Instagram, Facebook, TikTok, інфлюенс-маркетинг, прямі ефіри (лайвстрими).	Прямі продажі, залучення молоді, формування лояльності.
3	Розвиток фінтеху та оплат	BNPL (розстрочка), Apple Pay, Google Pay, Monobank, Privat24.	Зростання обсягів покупок, покращення зручності оплати.
4	Інновації в логістиці	Швидка доставка, пункти видачі, дрони, автономний транспорт.	Підвищення швидкості доставки та доступності сервісу.

У контексті зростаючої ролі електронної комерції в умовах воєнного часу, змінюється і структура споживчого попиту. Все більше українців віддають перевагу онлайн-покупкам, уникаючи відвідування торгових центрів, які можуть бути тимчасово недоступними через повітряні тривоги. Ці обставини вплинули на формування популярних товарних категорій у 2024 році. У табл. 1.2 наведено узагальнений перелік найпопулярніших товарів, які користуються високим попитом в українському e-commerce-сегменті, із зазначенням частки продажів, порівняльної динаміки та ключових причин їх зростання [42].

Таблиця 1.2

Основні товари, популярні серед онлайн-покупців в Україні у 2024 році

№	Категорія товарів	Частка продажів	Зростання до 2023	Ключові причини попиту	Основні тенденції 2024 року
1	2	3	4	5	6
1	Смартфони та мобільні пристрої	22%	+15%	Популярність m-commerce, активне використання 5G	Попит на бюджетні 5G-пристрої (Xiaomi, Samsung)
2	Побутова техніка	18%	+10%	Потреба в енергоощадних рішеннях	Смарт-пристрої з еко-режимами

Продовження таблиці 1.2

1	2	3	4	5	6
3	Одяг і взуття	17%	+12%	Повернення до офлайн-життя, гібридна робота	Популярність українських брендів і функціонального одягу
4	Ноутбуки та планшети	14%	+8%	Дистанційна робота й навчання	Гібридні моделі, тривалий час роботи без зарядки
5	Косметика та догляд за собою	13%	+20%	Успіх соцмереж і доставки	Натуральна косметика, засоби для домашнього догляду
6	Товари для дітей	10%	+9%	Зручність онлайн-замовлення, стабільна народжуваність	Освітні іграшки, екотовари, гаджети для навчання
7	Ігрові приставки та відеоігри	8%	+14%	Зростання геймінгу, дозволя вдома	Онлайн-ігри, DLC, популярність PlayStation та Xbox

Аналіз споживчих переваг демонструє, що у 2024 році лідерами продажів залишаються мобільні пристрої, побутова техніка, а також товари для дистанційної роботи та навчання. Водночас суттєво зростає попит на косметику, дитячі товари та відеоігри – сегменти, що відповідають за комфорт, дозволя та щоденні потреби. Важливою тенденцією є орієнтація споживачів на енергоефективність, екологічність, зручність доставки та активне використання цифрових каналів комунікації. Ці фактори формують нову парадигму онлайн-поведінки, яка буде визначати вектор розвитку українського e-commerce у найближчі роки.

Попри позитивну динаміку зростання, український ринок e-commerce у 2024 році стикається з низкою проблем, що гальмують його стабільний розвиток (рис.1.2). Чимало українських покупців досі з обережністю ставляться до онлайн-шопінгу. Це пов'язано зі страхом перед шахрайськими схемами, крадіжкою особистих даних і фінансових ризиків.

Хоча впроваджуються заходи з підвищення кіберзахисту, випадки фішингу, зламу акаунтів і несанкціонованого доступу до платіжної інформації залишаються актуальними.

Щоб зберегти довіру користувачів, інтернет-магазини повинні посилити системи захисту, впроваджувати сучасні протоколи шифрування та вчасно

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

реагувати на загрози безпеки.

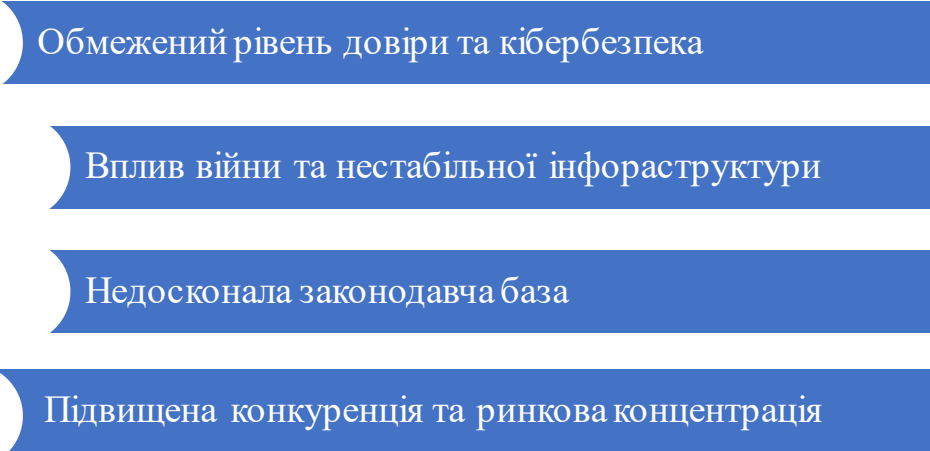


Рис.1.2 – Основні виклики розвитку електронної комерції в Україні у 2024 році

Збройний конфлікт з Росією продовжує негативно впливати на електронну комерцію, особливо в прифронтових областях. Основні труднощі пов’язані з перебоями електропостачання, руйнуванням логістичних маршрутів і нестабільністю поставок. Проте багато компаній демонструють високу здатність до адаптації: оперативно змінюють логістику, працюють з новими складами, і переорієнтовують доставку на відносно безпечні регіони.

Правове регулювання цифрової торгівлі в Україні поки що поступається європейським стандартам. Попри спроби оновлення нормативів – у сфері захисту прав покупців, збереження персональних даних і правил онлайн-торгівлі – законодавство залишається фрагментарним. Для формування стійкої цифрової економіки необхідні чіткі, прозорі та адаптовані до сучасних викликів правові норми.

Сегмент e-commerce стає дедалі більш насиченим: у боротьбу за покупця активно включаються не лише локальні компанії, а й міжнародні гравці.

Водночас великі бренди створюють власні мультибрендові платформи (маркетплейси), які витісняють дрібніших гравців. Станом на 2024 рік маркетплейси

вже займають понад 60% ринку, і ця цифра продовжує зростати, що ускладнює виживання нових або нішевих онлайн-магазинів без чіткої стратегії.

Отже, український ринок електронної комерції у 2024 році демонструє стійке зростання, незважаючи на складні соціально-економічні умови, спричинені війною та загальною нестабільністю. Онлайн-продажі стали не лише альтернативою традиційному шопінгу, а й ключовим інструментом підтримки бізнесу та задоволення потреб споживачів в умовах обмеженого доступу до офлайн-магазинів.

1.2. Аналіз існуючих аналогів інтернет-магазинів: функціонал, переваги та недоліки

Сучасний ринок електронної комерції представлений великою кількістю онлайн-магазинів, які мають різний рівень функціональності, зручності та технічної реалізації. Для успішного створення конкурентоспроможного проєкту важливо проаналізувати найпопулярніші платформи, що вже зарекомендували себе серед українських споживачів. Такий аналіз дозволяє не лише виявити сильні сторони конкурентів, а й зрозуміти їхні недоліки, що стане основою для розробки ефективного, зручного та сучасного інтернет-магазину.

Для повноцінного аналізу аналогів інтернет-магазинів необхідно визначити чіткі критерії, які дозволяють об'єктивно порівняти різні платформи за ключовими параметрами. Такий підхід дає змогу не лише оцінити переваги та недоліки кожного ресурсу, а й виокремити оптимальні рішення для впровадження у власному проєкті.

Розглянемо основні критерії, за якими будемо здійснювати оцінку (рис.1.3).



Рис.1.3. Критерії аналізу функціональності та якості існуючих інтернет-магазинів

Для порівняльного аналізу інтернет-магазинів було визначено низку ключових критеріїв, які дозволяють комплексно оцінити рівень їх функціональності та якості. Серед основних параметрів розглядаються зручність інтерфейсу, що включає логічну побудову структури сайту та простоту навігації, а також функціональність для користувача – можливість реєстрації, пошуку товарів, додавання до кошика та оформлення замовлення.

Також важливими є адаптивність і мобільна оптимізація, адже більшість користувачів здійснюють покупки з мобільних пристроїв. Значну роль відіграє система фільтрації й пошуку, яка дозволяє швидко знаходити потрібні товари, а також наявність відгуків і рейтингів, що впливають на довіру до магазину.

Швидкість завантаження сайту безпосередньо впливає на задоволеність клієнтів

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

і конверсію. Сукупність цих критеріїв дозволяє виявити сильні й слабкі сторони існуючих рішень на ринку.

Rozetka.ua [37] є найбільшою та найвідомішою платформою електронної комерції в Україні. Заснована у 2005 році як онлайн-магазин техніки, з часом вона трансформувалася у повноцінний маркетплейс, де окрім власних товарів реалізуються також пропозиції сторонніх продавців. Станом на 2024 рік Rozetka охоплює десятки товарних категорій – від електроніки та побутової техніки до одягу, книг, косметики, продуктів і навіть будівельних матеріалів.

Однією з головних переваг платформи є масштаб – широкий асортимент, наявність тисяч продавців, гнучка система фільтрів і високий рівень довіри з боку покупців. Rozetka пропонує користувачам повноцінний функціонал сучасного онлайн-шопінгу: від реєстрації та персонального кабінету до зручного кошика, оформлення замовлення, перегляду історії покупок та відстеження доставки. Особливу увагу приділено контенту – більшість товарів мають відеогляди, фото в кількох ракурсах, докладні характеристики та рейтинги покупців.

Сайт адаптований під мобільні пристрої, а також має функціональний мобільний застосунок, що забезпечує повноцінний доступ до каталогу, акцій, push-сповіщень та особистого кабінету.

Разом із тим Rozetka має і певні недоліки. Зокрема, інтерфейс платформи часто перевантажений рекламними блоками, банерами акцій та промо-контентом, що ускладнює навігацію, особливо для нових користувачів. Ще одним викликом є конкуренція між продавцями всередині маркетплейсу, що іноді створює ситуації з дублюванням товарів і розкиданими цінами на однакові позиції.

У цілому Rozetka виступає взірцем масштабного багатфункціонального ресурсу, що поєднує високу довіру клієнтів, технічну стабільність і широкий вибір товарів. Її досвід є цінним джерелом для вивчення найкращих практик в українському e-commerce-середовищі.

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Comfy.ua [23] є одним із найвідоміших українських онлайн-магазинів, який спеціалізується на продажу побутової техніки, гаджетів і електроніки. На відміну від універсальних маркетплейсів, таких як Rozetka, Comfy зосереджує увагу на вузькому асортименті, пропонуючи в першу чергу техніку для дому, смартфони, ноутбуки, телевізори та аксесуари. Цей підхід дозволяє компанії зосередитись на глибшому опрацюванні клієнтського досвіду в межах обраного сегменту.

Платформа має сучасний та мінімалістичний інтерфейс, який вирізняється зручністю навігації. Каталог товарів структурований логічно, а процес пошуку й фільтрації є швидким і зрозумілим навіть для недосвідчених користувачів. Для кожного товару доступна докладна інформація, характеристики, актуальні ціни, а також система рейтингу та відгуків.

Серед ключових функціональних переваг – онлайн-консультант, можливість купівлі в розстрочку або з оплатою частинами, а також програма лояльності, яка заохочує повторні покупки через бонуси. Важливою перевагою є також висока швидкість оформлення замовлення: оформити покупку можна в кілька кліків без зайвих переходів і перезавповнення форм.

Comfy активно використовує акційні кампанії, розпродажі, сезонні пропозиції та «чорні п'ятниці», що робить платформу привабливою для ціново чутливих споживачів.

Разом із тим варто зазначити і обмеження. Асортимент Comfy є меншим у порівнянні з маркетплейсами, оскільки компанія працює лише зі своєю товарною лінійкою й не надає торгових майданчиків стороннім продавцям. Це знижує конкурентність цін і обмежує варіативність пропозицій. Крім того, у деяких категоріях може бути недостатньо додаткових функцій, таких як відеоогляди чи розгорнуті порівняння.

Таким чином, Comfy позиціонується як якісний сервіс із вузькою спеціалізацією, який робить ставку на простоту, швидкість та лояльність до клієнта. Його приклад є корисним для проєктування інтерфейсу та UX-архітектури в електронній комерції.

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

MOYO.ua [36] – український онлайн-магазин, який спеціалізується на продажу електроніки, побутової техніки, комп'ютерних комплектуючих та аксесуарів.

На відміну від масштабних маркетплейсів, MOYO робить акцент не на обсязі товарних позицій, а на якісному сервісі, зручності взаємодії з користувачем і професійному контенті.

Сайт має стилістично вивірений інтерфейс, простий у навігації, з чіткою структурою каталогу. Кожна товарна позиція супроводжується розгорнутими характеристиками, якісними фото, а в багатьох випадках – відеооглядами, що дозволяють покупцеві ознайомитися з продуктом ще до оформлення замовлення. Також доступна форма зворотного зв'язку, де клієнти можуть поставити запитання або залишити відгук.

Окрема увага приділяється післяпродажному обслуговуванню та прозорій логістиці: MOYO гарантує інформування про статус замовлення, чітко вказує терміни доставки й активно підтримує клієнта на кожному етапі покупки. Це є однією з конкурентних переваг у сегменті електроніки, де довіра до продавця має вирішальне значення.

Платформа підтримує онлайн-оплату, оплату частинами, бонусні програми, а також здійснює доставку у співпраці з провідними логістичними операторами. Мобільна версія сайту адаптована до смартфонів і працює без збоїв, забезпечуючи комфортне користування.

Водночас MOYO має і певні обмеження. Порівняно з великими маркетплейсами, асортимент тут є суттєво вужчим, що пов'язано з власною спеціалізацією. Відсутність мультивендорної моделі означає, що покупець не може обирати з пропозицій різних продавців – уся відповідальність за якість і доставку лежить на одному операторі. Крім того, бренд MOYO менш впізнаваний серед широкого кола споживачів, особливо поза межами великих міст.

Незважаючи на це, MOYO є прикладом інтернет-магазину з професійним підходом до клієнтського сервісу, продуманим інтерфейсом та якісною

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

інформаційною подачею. Ці риси можуть бути особливо корисними для проєктування нішевих e-commerce платформ, орієнтованих на довіру та стабільність.

Allo.ua [2] – одна з найбільших українських e-commerce платформ, що поєднує в собі функціонал інтернет-магазину та маркетплейсу. Спочатку бренд був відомий як мережа роздрібних магазинів мобільної техніки, однак з розвитком цифрових технологій успішно трансформувався у повноцінну онлайн-платформу з широким асортиментом товарів і активною присутністю в діджитал-просторі.

Allo пропонує значну товарну лінійку, до якої входять смартфони, ноутбуки, побутова техніка, аксесуари, товари для дому, краси, спорту, а також окремі сегменти одягу та автотоварів. Платформа функціонує за гібридною моделлю: частина продукції реалізується безпосередньо компанією «Алло», решта – через сторонніх продавців, що робить її конкурентоспроможною у відношенні до таких маркетплейсів, як Rozetka.

Сайт вирізняється динамічним, яскравим інтерфейсом, який активно використовує рекламні банери, відеоогляди, блоки з акційними пропозиціями. Для залучення та утримання клієнтів використовуються сучасні маркетингові інструменти – регулярні знижки, бонусні програми, інтеграції з банками для покупки в розстрочку або «оплати частинами». Також функціонує онлайн-чат та підтримка 24/7, що покращує взаємодію з клієнтом на кожному етапі шопінгу.

Однією з сильних сторін Allo є активна присутність у соціальних мережах – компанія веде YouTube-канал із відеооглядами, сторінки в Instagram, TikTok і Facebook, що дозволяє їй ефективно комунікувати з аудиторією, зокрема молоддю. Крім того, доступний мобільний застосунок, який підтримує усі основні функції платформи.

Проте, як і в багатьох великих платформах, інтерфейс Allo може здаватися перевантаженим – надлишок рекламних елементів, яскравих банерів та акційних блоків іноді ускладнює навігацію. Також спостерігаються складнощі з логікою

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

побудови деяких категорій, що може впливати на швидкість пошуку потрібного товару. Ще одним моментом є неоднорідна якість обслуговування у випадку сторонніх продавців, що характерно для маркетплейсів.

У цілому Allo – це приклад потужної мультикатегорійної платформи, що динамічно розвивається і активно залучає аудиторію за допомогою контенту, цифрового маркетингу та омніканальної стратегії. Її модель цікава як з точки зору побудови гібридного магазину, так і з огляду на інструменти залучення клієнтів.

Проведений огляд дозволив детально дослідити особливості чотирьох провідних платформ електронної комерції в Україні: Rozetka, Comfy, MOYO та Allo. Кожен із розглянутих ресурсів має свою бізнес-модель, унікальні функціональні рішення, рівень сервісу, підхід до клієнта, маркетингу та структуру інтерфейсу. З метою систематизації та наочності порівняння їх основних характеристик, переваг і недоліків результати аналізу були узагальнені у Додатку А.

Таким чином, аналіз популярних українських платформ-аналогів показав, що кожна з них реалізує унікальний підхід до організації e-commerce, що залежить від масштабу, цільової аудиторії, товарного асортименту та технічної реалізації. Rozetka виступає універсальним маркетплейсом з максимальним покриттям аудиторії, Comfy орієнтується на швидкий і простий сервіс у вузькому сегменті, MOYO акцентує увагу на якісному обслуговуванні та довірі, а Allo вдало поєднує маркетплейс і роздрібний формат з активною присутністю в медіа.

1.3. Обґрунтування вибору інструментальних засобів

У процесі розробки сучасного, швидкого та зручного інтернет-магазину було вирішено використовувати Next.js [12] як основний фреймворк для реалізації Full-Stack веб-платформи. Такий вибір є обґрунтованим з огляду на низку технічних і практичних переваг, що забезпечують ефективну розробку, масштабування та подальше обслуговування проєкту.

По-перше, Next.js – це фреймворк на базі React, який підтримує гібридну

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

рендеринг-архітектуру: як серверний рендеринг (SSR), так і статичну генерацію сторінок (SSG). Це дозволяє будувати універсальні (isomorphic) веб-застосунки, що поєднують у собі переваги клієнтської й серверної обробки. Завдяки цьому інтернет-магазин може одночасно забезпечувати високу швидкість завантаження сторінок і SEO-оптимізацію, що є критично важливим для залучення трафіку з пошукових систем.

По-друге, висока продуктивність і оптимізація – одна з головних переваг Next.js. Фреймворк підтримує автоматичне кешування, оптимізацію зображень, інкрементальну генерацію (ISR), а також швидкий перехід між сторінками без повного перезавантаження. Це значно покращує користувацький досвід – важливий фактор у контексті e-commerce, де навіть затримка в кілька секунд може знизити конверсію.

Крім того, Next.js має вбудований роутинг без потреби у сторонніх бібліотеках, підтримує динамічні маршрути та API-роути, що дозволяє реалізувати бекенд-логіку безпосередньо в межах фреймворку. Це особливо корисно для таких функцій, як авторизація, обробка замовлень, підключення до бази даних, керування контентом тощо. Таким чином, Next.js забезпечує повноцінний Full-Stack функціонал у межах єдиної екосистеми.

Також важливо відзначити високу інтеграцію з TypeScript, що дозволяє створювати масштабовані та типобезпечні рішення, а також підтримку модульного CSS, Tailwind CSS, Redux або інших бібліотек за потреби. Це забезпечує гнучкість у виборі інструментів відповідно до специфіки конкретного проєкту.

Окремої уваги заслуговує потужна спільнота розробників та офіційна документація, що дає змогу швидко знаходити рішення типових проблем, використовувати готові компоненти та бібліотеки, а також впроваджувати найкращі практики сучасної веб-розробки.

У якості системи керування базами даних для інтернет-магазину було обрано MongoDB [33] – сучасну документоорієнтовану NoSQL-базу даних, яка

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечує високу продуктивність, гнучкість у структурі даних та ефективну масштабованість. Це рішення повністю відповідає потребам сучасного e-commerce-проєкту, в якому структура даних може бути динамічною та змінюватися в процесі розробки.

Однією з головних причин вибору MongoDB є її схемна гнучкість. На відміну від реляційних баз, MongoDB не вимагає жорсткого визначення структури таблиць – це дозволяє легко адаптувати схему документа під нові вимоги (наприклад, при розширенні інформації про товари або користувачів). У сфері інтернет-торгівлі це особливо корисно, адже різні категорії товарів можуть мати унікальні атрибути, які важко вмістити у фіксовану SQL-схему.

Крім того, MongoDB чудово інтегрується з Next.js через Mongoose або MongoDB Driver, що дозволяє реалізувати повноцінний Full-Stack стек на TypeScript без необхідності використовувати сторонні інтерфейси або ORM для реляційних СКБД. Така інтеграція спрощує розробку API-ендпоінтів, зменшує кількість проміжного коду та забезпечує високу швидкість обробки запитів.

Ще однією перевагою є горизонтальна масштабованість MongoDB. В умовах, коли інтернет-магазин може стрімко рости (збільшення кількості товарів, користувачів, замовлень), MongoDB легко масштабується шляхом додавання нових вузлів до кластеру, забезпечуючи стабільну роботу без втрати продуктивності.

Також варто відзначити широку підтримку хмарної інфраструктури, зокрема через MongoDB Atlas – офіційний керований хмарний сервіс, який дозволяє швидко розгорнути, моніторити та захищати базу даних без додаткових витрат на адміністрування. Завдяки цьому MongoDB є чудовим вибором для команд, які прагнуть зосередитись на функціональності проєкту, а не на технічному обслуговуванні СКБД.

З огляду на вищезазначені фактори, використання MongoDB у складі технологічного стеку є доцільним та обґрунтованим рішенням, що дозволяє побудувати масштабовану, гнучку та ефективну систему зберігання даних для

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

потреб інтернет-магазину.

Таким чином, з урахуванням поставлених цілей та функціональних вимог до інтернет-магазину комп'ютерних комплектуючих, телефонів і ноутбуків, було обґрунтовано вибір сучасного технологічного стеку, до складу якого входять Next.js як основний фреймворк для Full-Stack розробки та MongoDB як гнучка, масштабована база даних.

Next.js забезпечує високу продуктивність, підтримку SEO, SSR/SSG, гнучкий роутинг і зручну інтеграцію із зовнішніми API, що робить його ідеальним вибором для створення швидкого, зручного та сучасного вебзастосунку. MongoDB, у свою чергу, дозволяє ефективно зберігати та обробляти структуровані й напівструктуровані дані з можливістю швидкого масштабування, що є критично важливим для розвитку інтернет-магазину. Обрані інструменти повністю відповідають потребам проєкту та сприяють його успішній реалізації.

1.4. Постановка технічного завдання

Розробка повноцінного інтернет-магазину потребує чіткого визначення технічних вимог, функціональних можливостей та інструментальних засобів, які будуть використані в процесі реалізації. У межах цієї кваліфікаційної роботи бакалавра створюється сучасна Full-Stack веб-платформа для онлайн-продажу комп'ютерних комплектуючих, ноутбуків і мобільних телефонів.

1. Аналіз сучасних технологій та підходів:

- вивчення основних принципів розробки Full-Stack додатків з використанням Next.js, зокрема SSR, SSG, ISR;
- дослідження особливостей клієнтської частини на базі React.js;
- використання Context API або useReducer для локального управління станом додатку;
- розгляд найкращих практик інтеграції з REST API та підключення зовнішніх сервісів.

2. Формування функціональних та інформаційних вимог:

- створення головної сторінки з промо-контентом і навігацією;
- реалізація сторінки каталогу з категоріями товарів, сортуванням і фільтрами;
- виведення детальної інформації про кожен товар: опис, характеристики, фото, ціна;
- можливість додавання товарів до кошика з подальшим оформленням замовлення;
- реєстрація, авторизація та особистий кабінет користувача;
- підключення системи онлайн-оплати (наприклад, Stripe);
- інтеграція з API Нової Пошти та Укрпошти для вибору зручного способу доставки.

3. Проектування інтерфейсу користувача:

- адаптивна верстка з підтримкою мобільних, планшетних і десктопних пристроїв;
- впровадження сучасного інтерфейсу за допомогою Next.js + module.scss;
- побудова зручної навігаційної системи та логічної структури сторінок;

4. Розробка структури бази даних:

- створення колекцій у MongoDB: користувачі, товари, замовлення, оплати;
- забезпечення CRUD-функціоналу через API-ендпоінти Next.js;
- створення зв'язків між замовленням, товаром і користувачем;

5. Реалізація серверної логіки:

- розробка серверної частини з використанням API Routes у Next.js;
- обробка запитів на реєстрацію, вхід, оформлення замовлення, оплату;
- валідація даних користувача, захист від SQL-ін'єкцій;
- інтеграція з платіжним сервісом та логістичними API.

6. Тестування та перевірка працездатності:

- функціональне тестування основних сценаріїв;
- перевірка UI/UX компонентів;

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

– симуляція процесу покупки та перевірка обробки оплати.

Технічне завдання є основою для реалізації ефективного, масштабованого та адаптивного інтернет-магазину, що враховує сучасні вимоги до продуктивності, зручності та інтеграції з сервісами доставки й оплат.

Висновки до першого розділу

У результаті аналізу предметної області було визначено основні потреби й очікування користувачів, які взаємодіють з онлайн-магазинами техніки. Було окреслено ключові цілі платформи, серед яких – надання зручного доступу до каталогу товарів, реалізація процесу оформлення замовлення, оплати та доставки з урахуванням сучасних вимог до безпеки та користувацького досвіду.

У процесі дослідження аналогічних платформ-лідерів українського ринку, таких як Rozetka, Comfy, MOYO та Allo, проведено порівняльний аналіз функціоналу, переваг та недоліків. Це дало змогу сформулювати обґрунтовані вимоги до майбутньої системи, включаючи функціональні можливості, нефункціональні характеристики, а також критерії якості інтерфейсу та технічної реалізації.

На основі цих вимог було здійснено обґрунтований вибір інструментальних засобів, зокрема Next.js як фреймворку для Full-Stack розробки та MongoDB як бази даних. Також було визначено потребу в інтеграції з API платіжних сервісів і служб доставки (Нова Пошта, Укрпошта), що забезпечить комплексну реалізацію функціоналу вебзастосунку.

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ

2.1. Визначення функціональних і нефункціональних вимог до платформи

На етапі проєктування інформаційної системи одним із ключових завдань є чітке визначення вимог до майбутнього застосунку. Саме функціональні та нефункціональні вимоги визначають, що повинна робити система та якими характеристиками вона повинна володіти, щоб відповідати потребам користувачів і забезпечити стабільну роботу.

Функціональні вимоги визначають поведінку системи у відповідь на дії користувача. Вони описують, які саме можливості повинен мати веб-сайт для повноцінної реалізації процесів онлайн-торгівлі – від реєстрації до оформлення й оплати замовлення. У межах цього проєкту вимоги сформовані з урахуванням особливостей ринку електроніки та потреб кінцевого користувача (табл. 2.1).

Таблиця 2.1

Функціональні вимоги до інтернет-магазину

№	Функціональний модуль	Опис функціональності
1	Реєстрація та авторизація користувачів	Створення облікового запису, вхід, вихід
2	Каталог товарів	Перегляд товарів за категоріями, фільтрація, пошук, перегляд детальної сторінки товару
3	Кошик	Додавання, редагування кількості та видалення товарів, автоматичне підрахування суми
4	Оформлення замовлення	Заповнення контактних даних, вибір доставки через API Нової Пошти/Укрпошти
5	Онлайн-оплата	Оплата замовлення через інтеграцію з платіжною системою (наприклад, Stripe, LiqPay)
6	Особистий кабінет користувача	Перегляд історії замовлень, статусів, базове редагування профілю

У проєкті реалізується система користувацької аутентифікації, яка дозволяє створювати облікові записи, входити до системи та зберігати персональні дані. Каталог товарів реалізовано у вигляді категоризованого списку, що містить розширену фільтрацію, сортування та пошук із автозаповненням. Сторінка товару надає детальну інформацію про продукт, включаючи фото, опис,

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

характеристики та ціну.

Користувач може формувати замовлення за допомогою інтерактивного кошика, де передбачено зміну кількості товарів, видалення та підрахунок загальної суми. Після цього він переходить до оформлення, в якому заповнює контактні дані, обирає спосіб доставки через API Нової Пошти або Укрпошти, а також переходить до оплати замовлення онлайн через інтегровану платіжну систему (наприклад, Stripe [38]).

Отже, функціональні вимоги охоплюють повний цикл користувацької взаємодії з інтернет-магазином: від реєстрації до оплати замовлення. Вони забезпечують базову комерційну логіку платформи та створюють умови для зручного, безпечного й ефективного онлайн-шопінгу. Відповідність цим вимогам є критично важливою для забезпечення позитивного досвіду користувача та досягнення цілей проєкту.

Нефункціональні вимоги визначають властивості платформи, які не пов'язані безпосередньо з конкретними функціями, але мають істотне значення для якості, надійності та зручності її використання. Вони забезпечують технічну відповідність системи сучасним стандартам веб-розробки, зокрема продуктивність, безпеку, масштабованість, адаптивність та SEO (табл. 2.2).

Таблиця 2.2

Нефункціональні вимоги до інтернет-магазину

№	Нефункціональний модуль	Опис вимог
1	Продуктивність	Швидке завантаження сторінок завдяки SSR/SSG, оптимізація зображень та запитів до API
2	Адаптивність	Коректне відображення та повна функціональність сайту на мобільних пристроях, планшетах, ПК
3	Безпека	Шифрування даних користувача, захист паролів, захист від XSS, CSRF, HTTPS-з'єднання
4	Надійність	Стабільна робота серверної логіки, обробка помилок, інформування користувача у разі збоїв
5	SEO-оптимізація	Генерація мета-тегів, дружні URL, sitemap.xml
6	Масштабованість	Гнучка структура проєкту (Next.js), можливість розширення функціоналу без суттєвих змін у базі
7	Зручність користування (UX)	Інтуїтивний інтерфейс, логічна навігація, модульна SCSS-стилізація з урахуванням user experience

Сайт повинен демонструвати високу продуктивність – сторінки мають швидко завантажуватися завдяки використанню SSR/SSG, оптимізації зображень та кешуванню даних. Важливо також забезпечити адаптивність інтерфейсу: користувачі повинні мати комфортний доступ до функціоналу незалежно від типу пристрою (мобільний, планшет, десктоп).

Одним із ключових аспектів є безпека, що включає шифрування паролів, захист персональних даних, протидію XSS і CSRF-атакам, а також використання захищених HTTPS-з'єднань. Надійність передбачає стабільну роботу API-ендпоінтів, систему обробки помилок і можливість інформування користувача у випадку збоїв.

Сайт також має бути оптимізований для пошукових систем (SEO) – через використання мета-тегів, мікророзмітки, канонічних URL та генерацію sitemap.xml. Крім того, важливою вимогою є масштабованість проєкту: структура повинна дозволяти швидке додавання нових сторінок, категорій або функцій без потреби в глобальних змінах. З точки зору UX, застосунок має бути інтуїтивно зрозумілим, мати логічну навігацію та структуровану стилізацію.

Нефункціональні вимоги доповнюють функціональні, встановлюючи стандарти якості розробки. Вони дозволяють досягти високої продуктивності, доступності, безпеки та масштабованості веб-застосунку. Врахування цих вимог забезпечує стабільну та конкурентоспроможну роботу інтернет-магазину в умовах сучасного ринку.

Таким чином, визначення функціональних і нефункціональних вимог є ключовим етапом проєктування будь-якої інформаційної системи, зокрема й веб-платформи інтернет-магазину. У результаті цього аналізу було чітко окреслено, який функціонал має бути реалізований для користувача, а також які технічні, якісні та експлуатаційні характеристики повинен забезпечити сайт.

2.2 Розробка структури бази даних

Проектування структури бази даних є критично важливим етапом у створенні інформаційної системи електронної комерції. Від якості цієї структури

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

залежить не лише коректність збереження та обробки даних, але й продуктивність усього застосунку, масштабованість системи, простота супроводу та захист персональної інформації користувачів. Для реалізації функціоналу інтернет-магазину з продажу комп'ютерних комплектуючих, ноутбуків та телефонів у цьому дипломному проєкті розроблено п'ять основних моделей даних: Product, CartItem, Order, User та Subscription.

Модель Product описує основну бізнес-сутність інтернет-магазину – товар. Вона включає повну інформацію про позицію каталогу: назву, опис, зображення, ціну, кількість на складі та технічні характеристики (табл.2.3).

Таблиця 2.3

Опис таблиці Product

Поле	Тип	Призначення
_id	ObjectId	Унікальний ідентифікатор товару
name.UA	varchar	Назва товару українською мовою
name.EN	varchar	Назва товару англійською мовою
description.UA	varchar	Опис товару українською
description.EN	varchar	Опис товару англійською
price	number	Ціна товару
images	array of varchar	Масив посилань на зображення товару
slug	varchar	Унікальний slug товару для URL
category	enum (phone, laptop, component)	Категорія товару
specs	Map<string, string>	Технічні характеристики (CPU, RAM, Battery і т.д.)
quantity	number	Кількість одиниць товару на складі

Ця модель забезпечує основний набір функцій для побудови каталогу, пошуку та локалізованого виведення даних.

Модель CartItem використовується для тимчасового збереження вибраних товарів перед оформленням замовлення (табл. 2.4).

Таблиця 2.4

Опис таблиці CartItem

Поле	Тип	Призначення
id	varchar	Унікальний ідентифікатор товару в кошику
slug	varchar	Slug товару для маршрутизації
quantity	number	Кількість одиниць товару

Продовження таблиці 2.4

price	number	Ціна за одиницю товару
category	enum ('phones', 'laptops', 'components')	Категорія товару
name	varchar	Назва товару
image	varchar	Посилання на зображення товару

Ця модель служить базою для реалізації кошика як у локальному сховищі (frontend), так і у разі авторизованих користувачів (backend-підтримка).

Модель Order – центральна частина бізнес-логіки магазину, що забезпечує збереження всіх замовлень клієнтів (табл. 2.5).

Таблиця 2.5

Опис таблиці Order

Поле	Тип	Призначення
firstName	varchar	Ім'я замовника
lastName	varchar	Прізвище замовника
phone	varchar	Номер телефону
email	varchar	Email
address	varchar	Адреса доставки
comment	varchar	Коментар до замовлення
payment	varchar	Спосіб оплати
delivery	varchar	Спосіб доставки
items	array	Список товарів у замовленні
total	number	Загальна вартість замовлення
payed	boolean	Позначка про оплату
date	date	Дата оформлення замовлення
lang	enum ('UA', 'EN')	Мова замовлення
user	ObjectId	Посилання на користувача

Дана модель забезпечує повний облік усіх замовлень з можливістю інтеграції з платіжними системами (Stripe) та логістичними API (Нова Пошта, Укрпошта).

Модель User – ядро авторизації та персоналізації користувацького досвіду (табл. 2.6).

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

Опис таблиці User

Поле	Тип	Призначення
name	varchar	Ім'я користувача
email	varchar	Email користувача
password	varchar	Пароль (зашифрований)
phone	varchar	Номер телефону
city	varchar	Місто проживання
accessToken	varchar	Токен доступу
refreshToken	varchar	Токен оновлення
avatarURL	varchar	Посилання на аватар користувача
birthday	varchar (дата)	Дата народження користувача
favorite	array of ObjectId	Список улюблених товарів

Ця модель дозволяє реалізувати вхід/реєстрацію, авторизацію користувача, особистий кабінет та відображення історії замовлень.

Модель Subscription реалізує просту структуру для підписки на email-розсилку (табл. 2.7).

Таблиця 2.7

Опис таблиці Subscription

Поле	Тип	Призначення
email	varchar	Email підписника
subscribedAt	date	Дата підписки

Ця модель дозволяє інтегрувати email-маркетинг (Mailchimp, Sendgrid).

У структурі бази даних інтернет-магазину основні моделі пов'язані між собою через логічні або безпосередні зовнішні зв'язки. Зокрема, кожне замовлення (Order) пов'язане з користувачем (User) через поле user, що містить посилання на відповідний об'єкт у колекції користувачів. Таким чином забезпечується персоналізація історії покупок і можливість фільтрації замовлень за конкретними користувачами. Модель Order також містить масив товарних позицій, що є збереженими копіями інформації з колекції Product – таким чином, зміна властивостей товару після оформлення замовлення не впливає на історію замовлень.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

Кошик (CartItem) логічно пов'язаний із моделлю Product через ідентифікатор товару – хоча формальний зв'язок у базі не реалізований через зовнішній ключ, логіка додатку передбачає отримання повної інформації про товар за його id. Модель Subscription зберігає електронні адреси підписників і функціонує окремо, не пов'язуючись безпосередньо з User, що дозволяє підписуватись на новини навіть неавторизованим користувачам. Така структура зв'язків забезпечує узгодженість даних і підтримку повного циклу роботи інтернет-магазину – від управління користувачами та товарами до обробки замовлень і розсилок.

2.3 Проектування та реалізація алгоритмів роботи системи

Процес проектування алгоритмів функціонування інтернет-магазину охоплює побудову чіткої логіки взаємодії між користувачем і системою на всіх етапах: від першого візиту до сайту до завершення покупки та підтвердження замовлення. Основна мета – забезпечити зручну, інтуїтивно зрозумілу і безперебійну роботу веб-застосунку, що відповідає сучасним вимогам електронної комерції.

Платформа електронної комерції, яка реалізується у межах дипломного проєкту, базується на архітектурі Full-Stack з використанням фреймворку Next.js та бази даних MongoDB. Для реалізації серверної логіки застосовуються API Routes, а клієнтська частина побудована на React з використанням Context API для управління станом.

Розглянемо основні етапи роботи системи через призму ключових користувацьких сценаріїв:

– реєстрація та авторизація. Алгоритм перевіряє унікальність email користувача, хешує пароль за допомогою bcrypt та зберігає дані у базі. Після успішної авторизації користувачу видається JWT-токен, який зберігається у localStorage та використовується для захисту маршрутизованих сторінок і API-запитів.

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

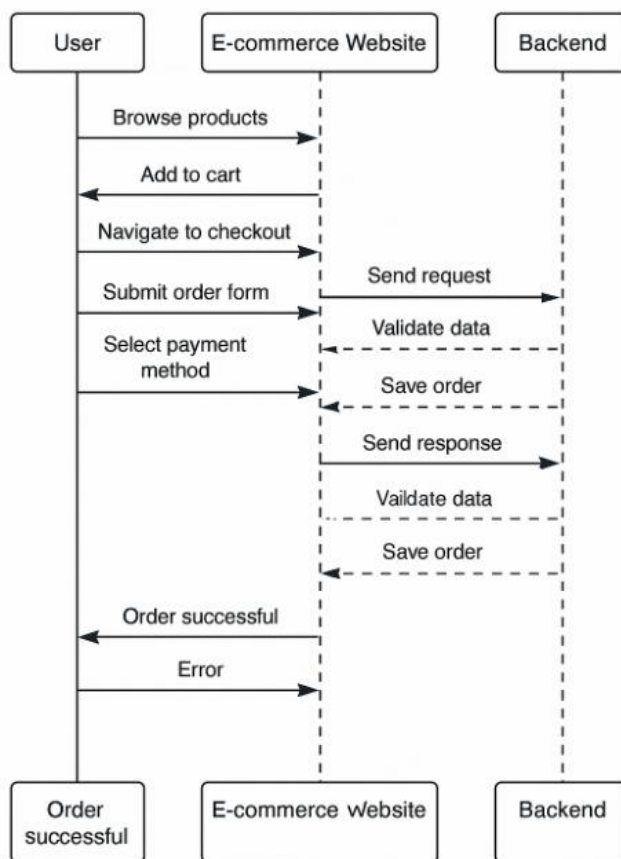


Рис.2.1. Схема алгоритмів

– перегляд та фільтрація товарів. Користувач взаємодіє з компонентом ProductList, який формує запити до серверу на основі активної категорії, мови інтерфейсу, сортування чи параметрів фільтрації. Сервер повертає відфільтровану відповідь, яку відображає UI. Продукти категоризовані як phones, laptops та components, кожен із яких містить специфікації, локалізовані назви та описи.

– додавання товару до кошика. Кошик реалізовано через Context API. При натисканні кнопки «Додати до кошика» запускається функція, яка перевіряє, чи користувач авторизований. Якщо ні – показується попередження. Інакше товар додається до локального localStorage, а також до глобального стею. При повторному додаванні змінюється лише кількість.

– оформлення замовлення. На сторінці /checkout користувач заповнює форму з особистими даними, обирає метод доставки (через інтеграцію з API Нової Пошти та Укрпошти [13]) і метод оплати (накладений платіж або Stripe).

Усі введені дані проходять валідацію через Yup та React Hook Form. Після відправлення форми дані замовлення зберігаються у колекції orders, а якщо було обрано оплату онлайн – користувача перенаправляють на сторінку платежу через Stripe.

Таким чином, алгоритми системи покривають повний цикл користування платформою: від реєстрації та пошуку товарів до оформлення і підтвердження замовлення. Завдяки розмежуванню ролей користувачів та безпечному API, забезпечується масштабованість, захищеність і зручність застосунку для обох сторін – покупців і адміністраторів.

2.4 Реалізація функціоналу магазину

Функціонал інтернет-магазину реалізовано у вигляді повноцінного веб-додатку, який забезпечує користувачам повний цикл взаємодії з платформою: від перегляду товарів до оформлення замовлення й оплати. На етапі реалізації особливу увагу було приділено зручності користування, безпеці, масштабованості та адаптивності для різних пристроїв.

Інтерфейс користувача створено за допомогою Next.js із використанням TypeScript та SCSS-модулів для стилізації. Це дозволило забезпечити SSR (Server-Side Rendering) і SSG (Static Site Generation), що позитивно впливає на SEO та швидкість завантаження. Вся логіка поділена на окремі компоненти – сторінки, елементи інтерфейсу (ProductCard, ProductList, Header, Cart тощо), що забезпечує легку підтримку та масштабування.

Реалізовано повний цикл автентифікації за допомогою Redux Toolkit, JWT-токенів та збереження токенів у localStorage. Після входу користувач отримує доступ до закритих маршрутів, включно зі сторінкою профілю /profile, де реалізовано:

- редагування персональних даних (ім'я, email, телефон, місто);
- завантаження аватара (FormData + Cloudinary) (Додаток Е – authSlice.ts).

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Уся інформація про товари отримується з бази даних MongoDB через відповідні API-маршрути, реалізовані в `pages/api/products/[id].ts`. Сторінки каталогів реалізовані за категоріями:

- телефони (`/phones`);
- ноутбуки (`/laptops`);
- комплектуючі (`/components`).

Детальна сторінка товару реалізована як динамічний маршрут, де виводиться: зображення товару; опис залежно від обраної мови користувачем (EN/UA); технічні характеристики; кнопка додавання до кошика.

На сторінці кожного товару реалізовано детальне представлення інформації, включно з описом, технічними характеристиками, зображенням і ціною. Кнопка “Купити” дозволяє миттєво додати товар до кошика.

Кошик реалізовано з використанням Context API, збереженням стану в `localStorage`, підтримкою трьох типів товарів (`phones`, `laptops`, `components`). Кожен товар має унікальні параметри: `slug`, `category`, `id`.

Функціонал:

- підрахунок загальної суми;
- зміна кількості;
- видалення товарів;
- передача замовлення у `CheckoutForm` (Додаток Г – `CartContext.ts`).

Форма оформлення замовлення реалізована з використанням `react-hook-form` та бібліотеки `yup` для валідації. Дані форми передаються у функцію `sendOrder`, яка здійснює POST-запит на маршрут `/api/checkout/orders`.

Користувач вводить: контактні дані (ПІБ, email, телефон); тип оплати (Stripe або накладений платіж); спосіб доставки (Нова Пошта або Укрпошта).

Інтеграція з API служб доставки реалізована у вигляді каскадного вибору області, міста та відділення:

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

```
const fetchCities = async (areaId) => {
  const response = await fetch(`/api/cities?areaId=${areaId}`);
  return await response.json();
};
```

Онлайн-оплата реалізована через Stripe Checkout Session. Після натискання на кнопку “Оплатити” викликається серверний маршрут `/api/stripe/session`, де створюється сесія та здійснюється редирект на сторінку Stripe (Додаток Д). Після успішної оплати користувача перенаправляє на `/success-order`:

```
const session = await stripe.checkout.sessions.create({
  payment_method_types: ['card'],
  line_items: [...],
  success_url: `${domain}/success`,
});
```

Після завершення замовлення його дані записуються до бази у вигляді моделі `Order.ts`, яка містить інформацію про товари, користувача, адресу та спосіб оплати. У схемі `Order` також передбачено зв’язок із користувачем:

```
user: {
  type: Schema.Types.ObjectId,
  ref: "User",
  required: true,
}
```

Користувач може підписатися на розсилку, ввівши email у відповідне поле. Дані зберігаються в моделі `Subscription` у базі MongoDB. Валідація email відбувається перед збереженням:

```
const subscriptionSchema = new Schema({
  email: { type: String, required: true, unique: true },
  subscribedAt: { type: Date, default: Date.now },
});
```

Реалізовано механізми захисту даних, зокрема: шифрування паролів за допомогою `bcrypt`; перевірку валідності токенів; захищені маршрути для

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

авторизованих користувачів; валідацію вхідних даних через уур.

Таким чином, реалізована система охоплює всі базові функціональні компоненти сучасного інтернет-магазину: каталогізацію, кошик, оформлення замовлення, обробку платежів, профіль користувача, а також підписку. Платформа повністю адаптивна, масштабована та може бути інтегрована з зовнішніми сервісами. Вся логіка взаємодії з базою даних побудована на основі Mongoose, що забезпечує стабільність, валідацію та зручну роботу з колекціями. Такий підхід дозволяє легко масштабувати застосунок, доповнюючи його новим функціоналом без ризику втрати сумісності.

Висновки до другого розділу

У другому розділі було здійснено поетапне проєктування функціональної структури веб-застосунку інтернет-магазину з урахуванням вимог до сучасних інформаційних систем електронної комерції. Було визначено основні функціональні складові системи, сформульовано вимоги до функціоналу платформи та обґрунтовано архітектурні рішення.

Окрему увагу приділено проєктуванню структури бази даних, зокрема визначено сутності користувачів, товарів, замовлень, підписок та елементів кошика. Було встановлено зв'язки між моделями з урахуванням логіки платформи – наприклад, зв'язок між користувачем і замовленням, а також між товарами та категоріями. Створено табличні описи моделей, що дозволяє чітко структурувати дані для зберігання та обробки.

Алгоритми взаємодії користувача з платформою проаналізовано за допомогою діаграм активності та послідовності. Описано, як реалізується типовий сценарій користування: перегляд каталогу, додавання товару в кошик, оформлення замовлення, вибір доставки та оплати, а також взаємодія з особистим кабінетом.

Також було реалізовано функціонал оформлення та обробки замовлення із інтеграцією зовнішніх API служб доставки (Нова Пошта, Укрпошта) та

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

платіжної системи Stripe. Для кожного з компонентів системи наведено схеми, що підтверджують його архітектурну доцільність і взаємозв'язок із загальною логікою застосунку.

Таким чином, на основі спроектованої логіки та реалізованих алгоритмів взаємодії, веб-застосунок інтернет-магазину набув завершеного функціонального вигляду, здатного забезпечити зручний та безпечний процес онлайн-купівлі, відповідно до сучасних технічних і UX-вимог.

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ІНТЕРФЕЙС ТА КОРИСТУВАННЯ ІНТЕРНЕТ-МАГАЗИНОМ

3.1 Користувацький інтерфейс і адаптивність верстки

Одним із ключових аспектів розробки інформаційної системи інтернет-магазину є створення інтуїтивно зрозумілого, логічно структурованого й візуально комфортного користувацького інтерфейсу (UI). В умовах стрімкого розвитку мобільних технологій усе більше користувачів взаємодіє з веб-платформами через смартфони та планшети, що зумовлює необхідність впровадження адаптивного дизайну.

Користувацький інтерфейс структурується за допомогою компонентного підходу, що дозволяє розділити логіку й візуальне оформлення на ізольовані та повторно використовувані частини. У складі UI передбачено окремі компоненти для заголовка (Header), підвалу (Footer), картки товару (ProductCard), списку товарів (ProductList), кошика (Cart), сторінки оформлення замовлення (CheckoutForm) тощо.

На рис. 3.1 представлено приклад зовнішнього вигляду головної сторінки інтернет-магазину у десктопній версії.

Застосування SCSS-модулів забезпечує як ізольованість стилів між компонентами, так і контроль над модифікаторами залежно від стану або ширини екрана. У проєкті активно використано медіа-запити для підтримки адаптивного відображення інтерфейсу. Наприклад, зміну орієнтації картки товару або сортування елементів на сторінці залежно від типу пристрою реалізовано завдяки кастомним breakpoint-мікسينам. Візуальні елементи автоматично підлаштовуються під розміри екрана, змінюючи сітку, відступи, шрифти та розміщення елементів.

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

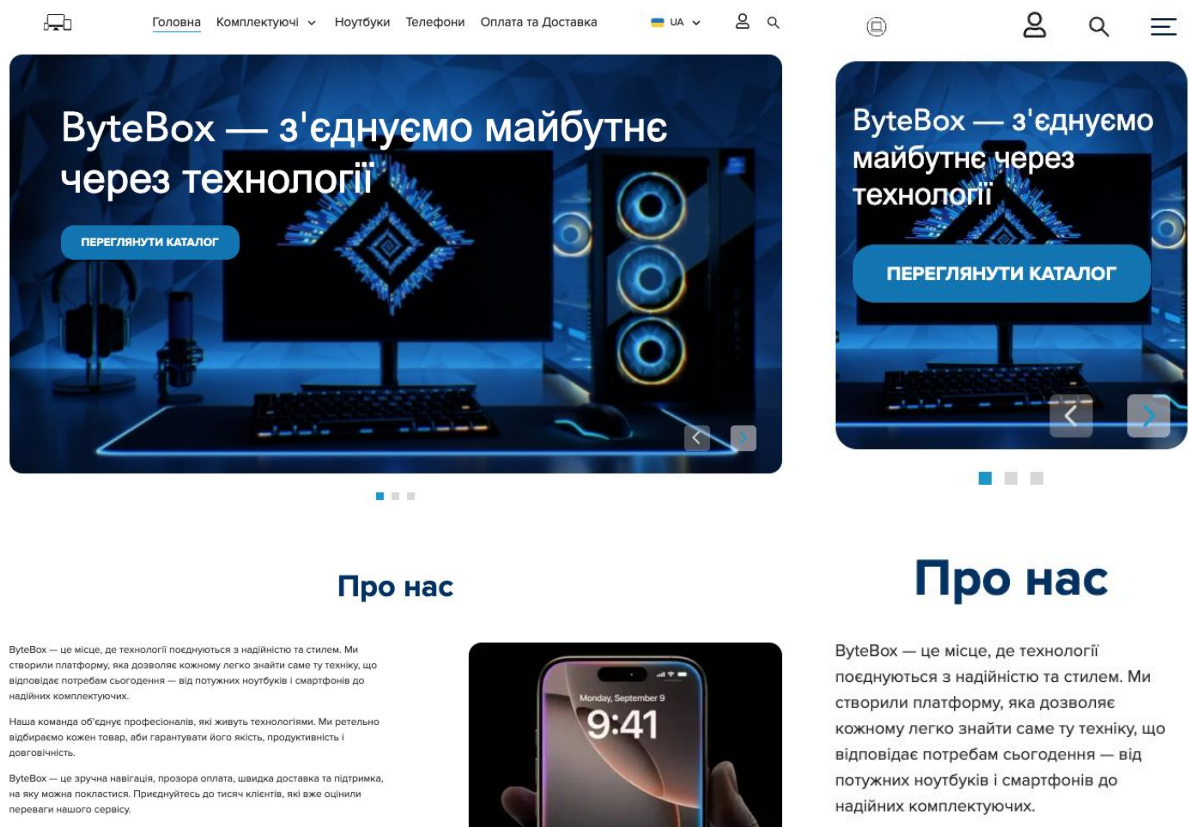


Рис.3.1. Головна сторінка інтернет-магазину в десктопній версії та адаптивне відображення на мобільному пристрої

Як видно з рисунку, при зменшенні ширини вікна браузера елементи сторінки товару переходять у вертикальне розміщення з адаптованими кнопками та полями. Основні елементи інтерфейсу (зображення товару, назва, ціна, опис, функціональні кнопки «Купити» та «Переглянути») змінюють своє положення відповідно до розміру екрана користувача. Такий підхід дозволяє забезпечити зручність перегляду товарів на смартфонах та планшетах без втрати доступу до ключових функцій.

Рис. 3.2-3.4 демонструють сторінки конкретного виду товару (телефони, ноутбуки, комплектуючі), де відображено зображення товару, його назву, опис, ціну та функціональні кнопки взаємодії. Особливу увагу було приділено читабельності шрифтів, розміщенню кнопок і логічному групуванню інформації.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

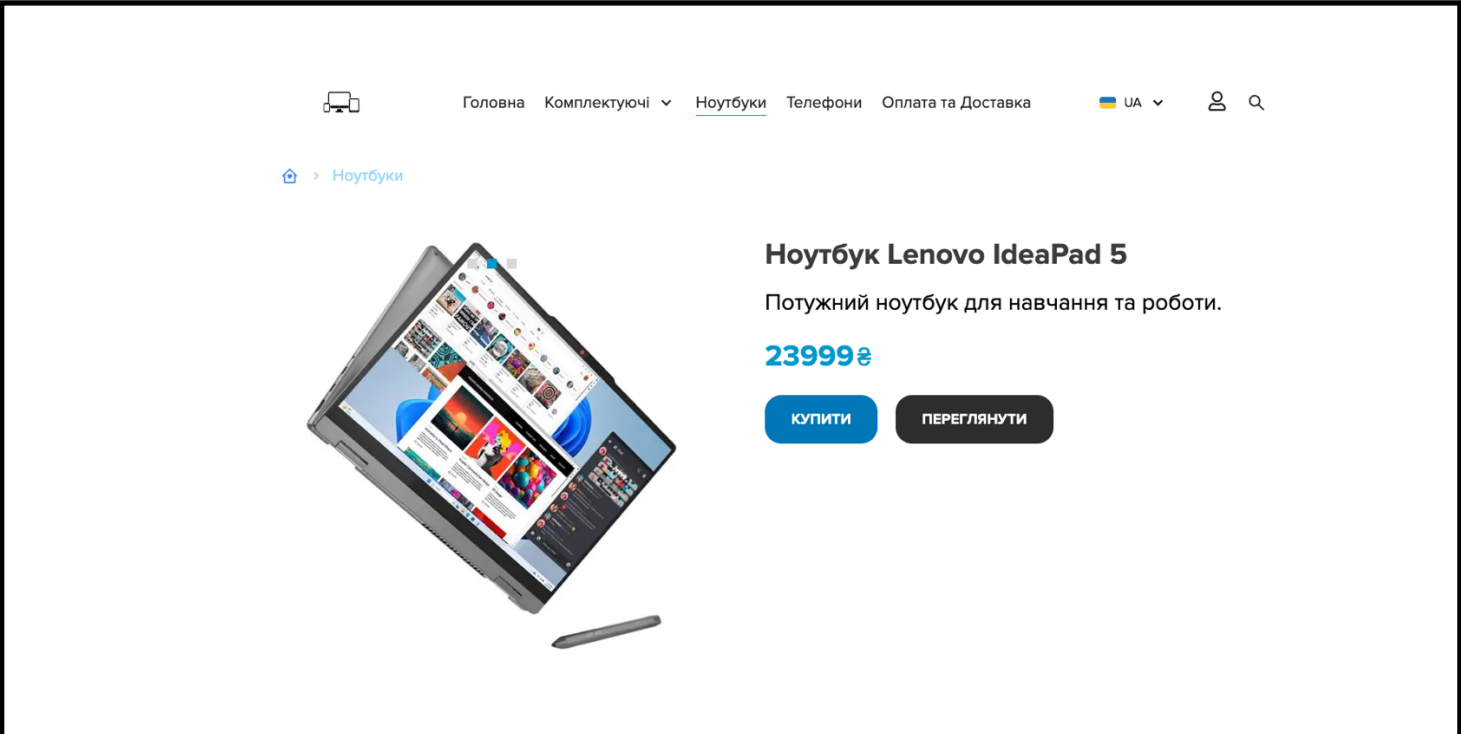
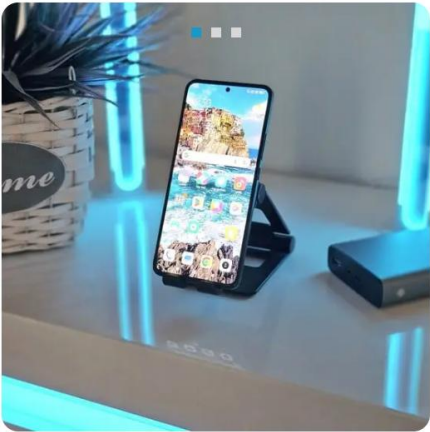


Рис. 3.2. Сторінка ноутбуків з кнопками купівлі та перегляду



Xiaomi 13T Pro Smartphone
High-performance smartphone with Leica camera, vivid AMOLED display, and powerful processor.
27999€

BUYREVIEW

Smartphones for Work and Life
Our collection features only reliable, modern smartphones that meet today’s demands. From budget-friendly models to flagship devices — we’ll help you find the perfect option for studying, working, creativity, or entertainment. All phones come with an official warranty, full set of accessories, and fast delivery. Choose what suits you best.

Рис. 3.3. Сторінка телефонів з кнопками купівлі та перегляду

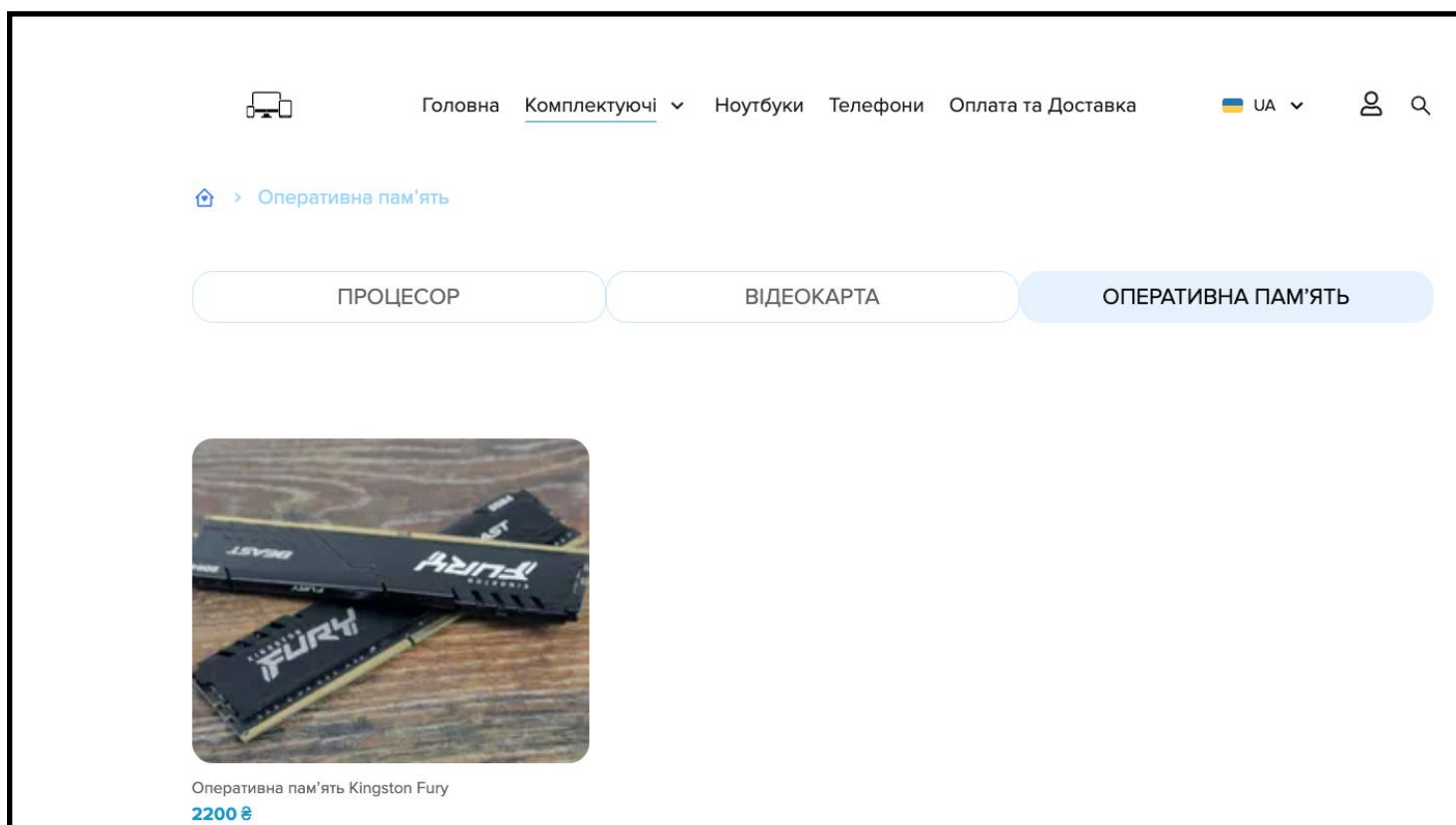


Рис. 3.4. Сторінка комплектуючих

Важливою складовою інтерфейсу є не лише зовнішнє представлення, але й взаємодія користувача із системою. Кожна дія – додавання товару до кошика, перехід на сторінку оформлення, реєстрація чи оновлення профілю – супроводжується візуальним або текстовим підтвердженням через toast-повідомлення або зміну інтерфейсного стану компонента. Наприклад, при успішному додаванні товару в кошик користувач бачить спливаюче повідомлення у верхній частині екрана (рис. 3.5), що створює відчуття контрольованого процесу та покращує загальний досвід взаємодії.

Окрім того, на сторінці комплектуючих реалізовано додаткові механізми зручності: користувач може здійснити швидке перемикання між підкатегоріями (процесори, відеокарти, оперативна пам'ять), не залишаючи поточної сторінки. Це досягається завдяки динамічному завантаженню товарів на основі активного фільтра, що суттєво зменшує час очікування та знижує навантаження на сервер.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

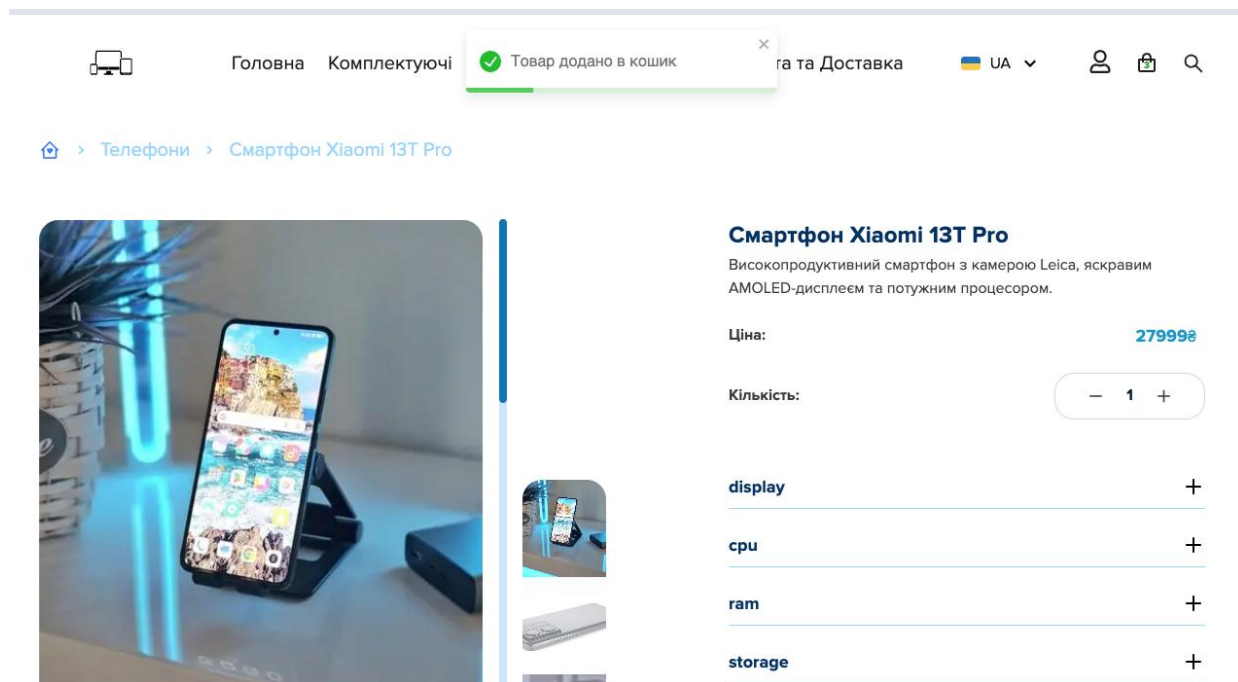


Рис.3.5. Повідомлення про успішне додавання товару до кошика

Для забезпечення максимальної доступності інтерфейс протестовано на різних типах пристроїв і браузерів за допомогою інструментів Chrome DevTools, BrowserStack, а також ручного UX-тестування. Приділено увагу доступності (accessibility), включаючи контрастність тексту, достатній розмір клікабельних елементів, логічну ієрархію заголовків та aria-атрибути, що дозволяє користувачам із обмеженими можливостями повноцінно взаємодіяти із сайтом.

Інтерфейс checkout-форми має важливе значення для користувача і був реалізований з особливою увагою до зручності. Форма включає валідацію даних у реальному часі, динамічне оновлення залежно від обраного методу доставки, а також інтеграцію з API поштових сервісів. Наприклад, поле вибору міста та відділення оновлюється асинхронно на основі обраної області, що реалізується через CustomSelect та виклики API при виборі опції (рис. 3.4). Форма адаптивна, працює на всіх екранах, включаючи мобільні пристрої, що про демонстровано на рис. 3.6 та 3.7.

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Доставка

☒ Нова Пошта (Відділення)
 ☐ Нова Пошта (Поштомат)

☐ Укр Пошта

Оберіть область доставки *

Житомирська
 

Оберіть ваше місто *

Місто
 

Авратин

 Андріївчії

 Андріївка (Пулинський р-н, Житомирська обл.)

 Андріївка (Черняхівський р-н, Житомирська обл.)

 Андрушівка

 Андрушки

 Артинськ

Рисунок 3.6 – Динамічне оновлення полів доставки у формі замовлення

Доставка

☒ Нова Пошта (Відділення)

☐ Нова Пошта (Поштомат)

☐ Укр Пошта

Оберіть область доставки *

Житомирська
 

Оберіть ваше місто *

Житомир
 

Оберіть номер відділення або поштомату *

Номер відділення або поштомату
 

Відділення №2 (до 30 кг на одне місце):
вул. Небесної Сотні, 52

 Відділення №3 (до 30 кг на одне місце):
майдан Перемоги, 10

 Відділення №4 (до 30 кг): вул.
Чуднівська, 113А

 Відділення №5 (до 30 кг): вул. Б. Тена, 9

 Відділення №6 (до 30 кг): вул. Київська,

Рисунок 3.7 – Мобільна версія checkout-форми на етапі оформлення замовлення

Візуальна структура інтерфейсу підтримує логіку маршрутизації, реалізовану через систему динамічних сторінок у Next.js. Наприклад, сторінка конкретного товару генерується динамічно на основі slug, а URL має вигляд /phones/123, що також сприяє SEO-оптимізації й дозволяє ефективно індексувати

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

сторінки в пошукових системах. Таким чином, навігаційна модель підтримує як статичні, так і динамічні маршрути, об'єднуючи їх у зручну систему доступу до різних розділів сайту.

Загалом, реалізація користувацького інтерфейсу інтернет-магазину базується на сучасних принципах frontend-розробки, що поєднують адаптивність, модульність, продуктивність і зручність. Адаптивна верстка дозволяє забезпечити повноцінний досвід користувача незалежно від пристрою, а чітка компонентна архітектура – підтримку та масштабування інтерфейсу в майбутньому.

3.2 Реалізація особистого кабінету, реєстрації та авторизації

Однією з ключових складових інтернет-магазину є система реєстрації, авторизації та особистого кабінету користувача. Її впровадження є обов'язковою умовою для забезпечення індивідуалізації користувацького досвіду, безпеки персональних даних, а також реалізації базових комерційних функцій, таких як формування замовлень, редагування профілю тощо.

Реєстрація користувача є першим етапом взаємодії користувача з платформою. Вона реалізована у вигляді окремої сторінки з формою, яка включає такі поля: ім'я, email, номер телефону, місто проживання, пароль та підтвердження пароля. Для перевірки правильності введених даних використовується бібліотека уцр, яка дозволяє задати правила валідації: довжина пароля, формат email, відповідність поля підтвердження тощо.

Після успішного заповнення форми та натискання кнопки «Зареєструватися» дані надсилаються POST-запитом на сервер, де на рівні API обробляються й додаються до колекції users у базі MongoDB. Паролі попередньо шифруються за допомогою алгоритму bcrypt.

На рис. 3.8 представлено інтерфейс форми реєстрації, яка включає обов'язкові поля та візуальну індикацію помилок.

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		



Електронна пошта *

test_ac@gmail.com

Пароль *

Підтвердження пароля *

ДАЛІ

Ім'я *

Ім'я

Обов'язково

Місто *

Місто

Обов'язково

Телефон *

+380

_ _ _ _ _

Обов'язково

ЗАРЕЄСТРУВАТИСЯ

НАЗАД

Рисунок 3.8 – Сторінка реєстрації користувача

```
_id: ObjectId('6826f4849e66a221fcc7b50b')
name: "Tester"
email: "Tester@ukr.net"
password: "$2b$10$Zr0fopZbHd0x5Yw0SDFr5edSPx9GtYaqKeW2gC3aWyclggIhzpLSC"
phone: "0639872535"
city: "Lviv"
accessToken: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjZmNDg0OWU2NmEyMjFmYy..."
refreshToken: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MjZmNDg0OWU2NmEyMjFmYy..."
avatarURL: "https://cdn.pixabay.com/photo/2015/10/05/22/37/blank-profile-picture-9..."
birthday: ""
favorite: Array (empty)
createdAt: 2025-05-16T08:17:08.404+00:00
updatedAt: 2025-05-24T15:00:48.087+00:00
```

Рис.3.9. – Успішна реєстрація і додавання користувача до бази даних

Функціонал авторизації надає можливість зареєстрованим користувачам входити до системи за допомогою email та пароля. Після успішної перевірки облікових даних сервер формує JWT-токени (accessToken і refreshToken), які

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

зберігаються у локальному сховищі (localStorage) або у глобальному стані додатку через Redux Toolkit. У разі неправильного пароля або незареєстрованого користувача сервер повертає відповідну помилку, яка відображається у вигляді сповіщення.

На стороні клієнта реалізовано перевірку токена для обмеження доступу до особистого кабінету. Усі запити до захищених маршрутів супроводжуються accessToken у заголовках, а при його втраті користувача автоматично перенаправляє на сторінку входу.

На рис. 3.10 зображено форму авторизації з полями email та пароль.

Рисунок 3.10 – Авторизація користувача

Після авторизації користувач отримує доступ до особистого кабінету, що реалізований як окрема сторінка профілю. У кабінеті відображаються персональні дані, які користувач вводив під час реєстрації: ім'я, email, телефон, місто, а також аватар (рис. 3.11). Передбачена можливість редагування основних полів, що реалізується через форму з попередньо заповненими значеннями. Дані змінюються шляхом надсилання PATCH-запиту на відповідний API-ендпоінт, де обробляються й оновлюються в базі даних.

На момент реалізації функціонал перегляду історії замовлень ще не реалізовано, однак структура бази даних передбачає зв'язок моделі Order із користувачем через user: Schema.Types.ObjectId, що дозволить у подальшому реалізувати повну історію покупок.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

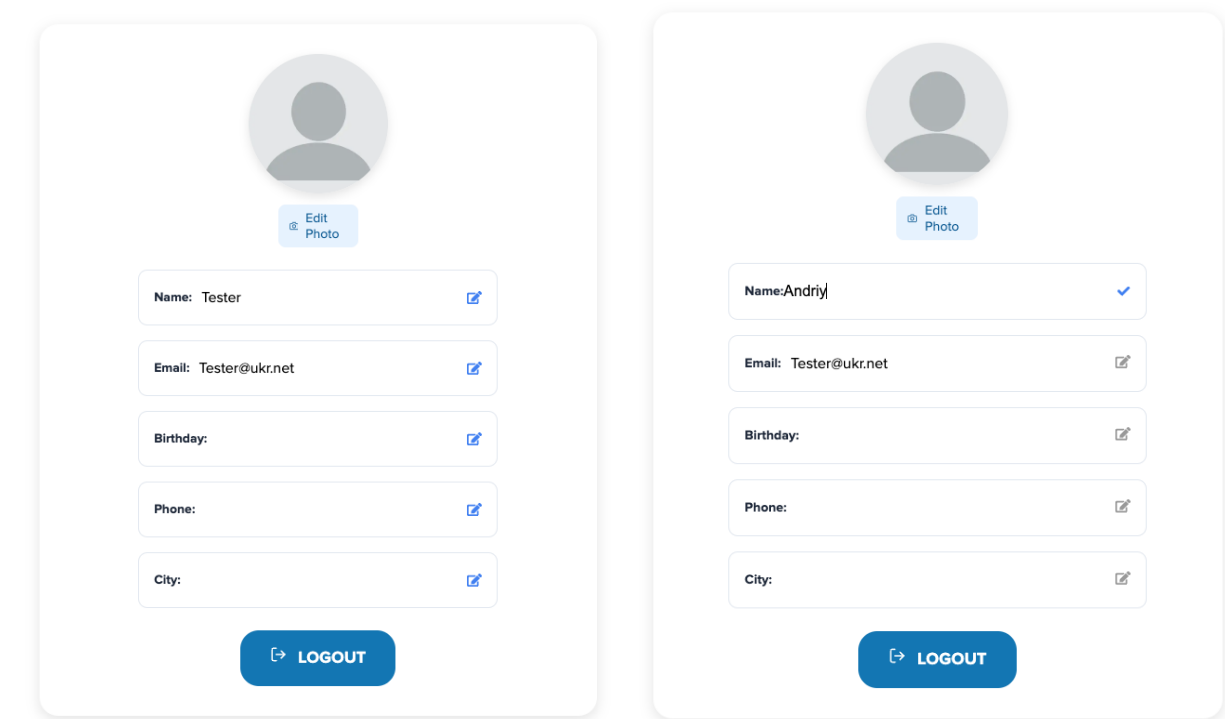


Рисунок 3.11 – Сторінка профілю користувача з можливістю редагування даних

Зміни збережених даних підтверджуються валідацією, а також візуальним відгуком у вигляді тост-повідомлення про успішне оновлення.

Механізм виходу з системи реалізовано через очищення токенів з localStorage або з глобального стану (Redux store), після чого користувача автоматично перенаправляє на головну сторінку. Це гарантує відсутність доступу до закритих маршрутів без повторної авторизації.

Безпечна обробка даних користувачів забезпечується на рівні серверу шляхом: хешування паролів (bcrypt); перевірки токенів (jsonwebtoken); авторизаційного middleware для захисту маршрутів; валідації вхідних даних (yup на клієнті та сервері).

Також обов'язкове використання HTTPS-з'єднання забезпечує захищену передачу конфіденційної інформації між клієнтом і сервером.

Таким чином, у межах реалізації особистого кабінету було створено надійну систему реєстрації, авторизації та редагування даних користувача. Це забезпечило персоналізовану взаємодію з платформою та підвищило рівень безпеки, що є критично важливим для будь-якого сучасного інтернет-магазину.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3 Оформлення замовлення: кошик, оплата та інтеграція з API служб доставки

Завершальний етап взаємодії користувача з платформою інтернет-магазину охоплює механізм формування та обробки замовлення. У цьому процесі тісно інтегруються кілька функціональних компонентів: кошик, форма оформлення, оплата, а також модулі доставки. Їх злагоджена робота гарантує коректне збереження даних користувача, точну передачу замовлення на сервер і можливість гнучкої інтеграції з зовнішніми API служб логістики та онлайн-платежів. У даному підрозділі описано реалізацію кожного з цих компонентів.

У розробленому вебзастосунку інтернет-магазину реалізовано функціонал пошуку товарів за назвою з підтримкою автозаповнення, що забезпечує швидкий доступ до потрібних позицій каталогу без необхідності навігації по сторінках.

З технічного боку, пошук виконано у вигляді динамічного інпуту, який реагує на введення користувачем тексту. У процесі введення даних запит надсилається на бекенд, де виконується пошук товарів за частковим співпадінням у полі `name.uk` (або `name.en` залежно від мови інтерфейсу). Результати пошуку повертаються у вигляді масиву об'єктів товарів, який миттєво відображається у випадяючому списку під інпутом.

Приклад реалізації показано на рис. 3.12, де користувач вводить запит «Xiaomi», а в результаті автозаповнення відображається товар «Смартфон Xiaomi 13T Pro» разом із ціною (27999 грн). Таким чином, користувач може миттєво перейти до перегляду конкретного товару без додаткових кліків.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

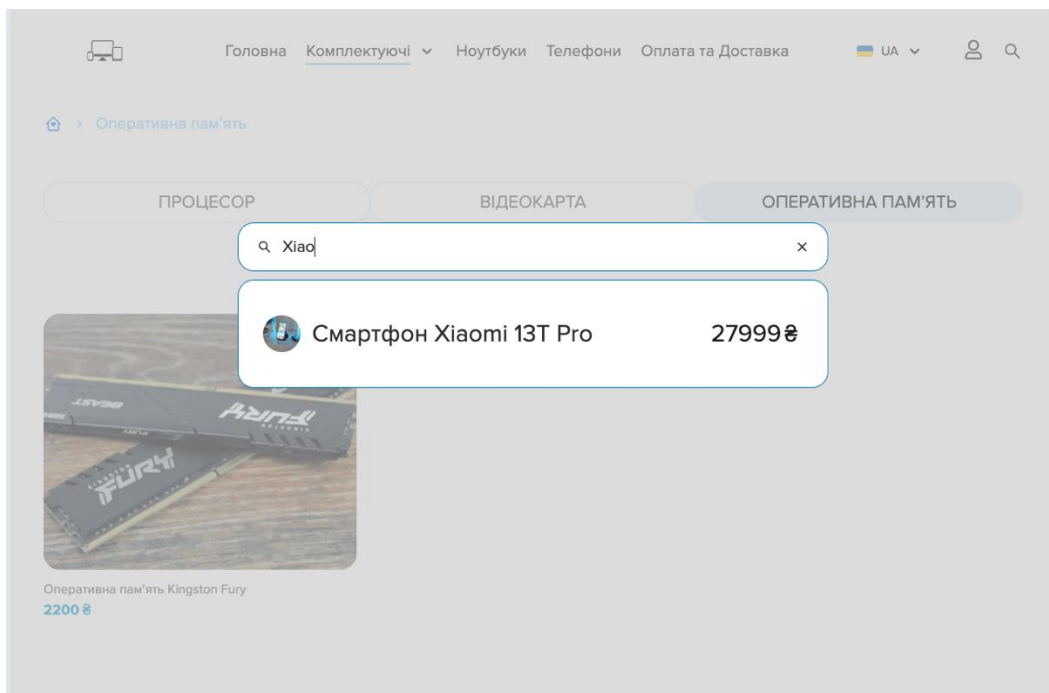


Рис.3.12 – Результати динамічного пошуку за ключовим словом "Xiao"

Цей функціонал значно підвищує зручність використання сайту, особливо при великій кількості товарів, і є важливим елементом користувацького досвіду. Він дозволяє зекономити час користувача та пришвидшує навігацію по каталогу.

Кошик реалізовано з використанням Context API React, що дозволило уникнути зайвого стану в компонентах та централізовано зберігати перелік обраних товарів. Дані кошика синхронізуються з localStorage, що забезпечує збереження товарів між сесіями користувача. Додавання товару викликає контекстну функцію addPhonesToCart, яка приймає об'єкт товару з категорією, кількістю, ціною та локалізацією. Також реалізовано функції редагування кількості (incrementQuantity, decrementQuantity) та видалення товару з кошика (removeFromCart), що підтримують гнучкість управління позиціями.

На рис. 3.13 представлено інтерфейс кошика з обраними товарами, підсумковою сумою та можливістю переходу до оформлення замовлення.

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Головна

Комплектуючі ▾

Ноутбуки

Телефони

Оплата та Доставка

UA ▾

>

Кошик

Смартфон

Високопродуктивний смартфон з камерою Xiaomi 13T Leica, яскравим AMOLED-дисплеєм та Pro потужним процесором.

27999 ₪

– 1 +

Видалити

Ноутбук Lenovo IdeaPad 5

Потужний ноутбук для навчання та роботи.

23999 ₪

– 1 +

Видалити

Разом

51998 ₪

Контактні дані

Ім'я *

Ім'я

Прізвище *

Прізвище

Email *

Email

Введіть номер телефону

+380

93-000-00-00

Доставка

☒ Нова Пошта (Відділення)

☐ Нова Пошта (Поштомат)

☐ Укр Пошта

Оберіть область доставки *

Область

▾

Рис. 3.13 – Інтерфейс кошика з обраними товарами та підсумком замовлення

Форма оформлення замовлення реалізована у вигляді окремого компонента CheckoutForm.tsx (рис. 3.14). Для роботи з формами використовується бібліотека react-hook-form, а валідація виконується за допомогою уцр , що показано на рис. 3.15. Користувач заповнює контактні поля (ім'я, прізвище, email, телефон, місто), обирає спосіб оплати (накладений платіж або онлайн через Stripe) та доставку.

Уся логіка перевірки коректності введених даних реалізована як на клієнтській, так і на серверній частині. При валідації також враховується локаль, передана через query.lang, що перевіряється на відповідність UA або EN.

Крім того, після відправлення форми користувачу надається візуальне підтвердження успішного створення замовлення у вигляді toast-повідомлення, а система автоматично очищує кошик та виконує перенаправлення на сторінку підтвердження. Це створює безперервний і зрозумілий сценарій взаємодії, що забезпечує користувача впевненістю в тому, що його дані оброблені належним чином.

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Контактні дані

Ім'я *

Ім'я

Прізвище *

Прізвище

Email *

Email

Введіть номер телефону

 +380 93-000-00-00

Рис. 3.14 – Форма оформлення замовлення на сторінці Checkout

Контактні дані

Ім'я *

Надія

Введіть ім'я

Прізвище *

Сироїд

Введіть прізвище

Email *

shcherbelova90@gmail.com

Введіть email

Введіть номер телефону

 +380 887665

Введіть дійсний номер телефону

Рис. 3.15 – Приклад валідації полів форми

Платформа підтримує вибір доставки через API служб Нової Пошти або Укрпошти (рис. 3.16). Цей функціонал реалізовано через низку кастомних запитів, які звертаються до відповідних REST API залежно від вибраного способу доставки. Запити здійснюються поетапно: спочатку отримується список областей (fetchAreas, fetchAreasUkr), після чого – міста (fetchCities), а далі – доступні відділення (fetchWarehouses) (рис. 3.17).

На фронтенді ці запити обробляються асинхронно, із застосуванням компонента CustomSelect, який дозволяє користувачеві обрати значення зі списку. Усі запити відправляються з відповідною локалізацією.

Доставка

☒ Нова Пошта (Відділення)

☐ Нова Пошта (Поштомат) ☐ Укр Пошта

Оберіть область доставки *

Житомирська

Оберіть ваше місто *

Місто

- Авратин
- Андріївці
- Андріївка (Пулинський р-н, Житомирська обл.)
- Андріївка (Черняхівський р-н, Житомирська обл.)
- Анрушівка
- Ангомиши

☐ Оплата за рахунком

Рис. 3.16 – Випадаючі списки з даними про область, місто та відділення

```
cities
{
  "success": true,
  "data": Array(10516),
  "errors": Array(0),
  "warnings": Array(0),
  "info": (.-),
  "..."
}
▼ [0 ... 9999]
  ▼ [0 ... 9999]
    ▶ 0: {Description: 'Абазіа (Полтавський р-н, Полтавська обл.)', DescriptionRu: 'Абазовка (Полтавский р-н, Полтавская обл.)', Ref: 'fc5f1e3c-928e-11e9-898c-005056b24375', Delivery1: '1', Delivery2: '0', ...}
    ▶ 1: {Description: 'Абранка', DescriptionRu: 'Абранка', Ref: '1b009444-4e4a-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '0', ...}
    ▶ 2: {Description: 'Абрикосівка', DescriptionRu: 'Абрикосовка', Ref: '6dbe932e-1aad-11e9-8c15-0025b502a06e', Delivery1: '1', Delivery2: '0', ...}
    ▶ 3: {Description: 'Абрикосівка (Хмельницька обл.)', DescriptionRu: 'Абрикосовка (Хмельницкая обл.)', Ref: 'f4d95497-de39-11e9-80fb-b8830365bd04', Delivery1: '1', Delivery2: '1', ...}
    ▶ 4: {Description: 'Абрикосове', DescriptionRu: 'Абрикосовое', Ref: '09add9ee-2544-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '0', ...}
    ▶ 5: {Description: 'Авангард', DescriptionRu: 'Авангард', Ref: '8e1718f5-1972-11e5-add9-005056b24375', Delivery1: '1', Delivery2: '1', ...}
    ▶ 6: {Description: 'Августинівка', DescriptionRu: 'Августиновка', Ref: '93edc540-9e41-11e9-898c-005056b24375', Delivery1: '1', Delivery2: '1', ...}
    ▶ 7: {Description: 'Августівка', DescriptionRu: 'Августовка', Ref: '9523ea02-7302-11e9-898c-005056b24375', Delivery1: '1', Delivery2: '1', ...}
    ▶ 8: {Description: 'Адаїа (Донецька обл.)', DescriptionRu: 'Адаевка (Донецкая обл.)', Ref: 'a9522a7e-caf5-11e7-ba66-005056b2fc3d', Delivery1: '1', Delivery2: '1', ...}
    ▶ 9: {Description: 'Адаїа (Сосницький р-н, Чернівецька обл.)', DescriptionRu: 'Адаевка (Сосницкий р-н, Черновицкая обл.)', Ref: 'e3f238f7-5a94-11e9-a703-005056b24375', Delivery1: '1', Delivery2: '1', ...}
    ▶ 10: {Description: 'Адаїа (Сосницький р-н, Чернівецька обл.)', DescriptionRu: 'Адаевка (Сосницкий р-н, Черновицкая обл.)', Ref: 'e3f238f7-5a94-11e9-a703-005056b24375', Delivery1: '1', Delivery2: '1', ...}
    ▶ 11: {Description: 'Авратинське', DescriptionRu: 'Авратинское', Ref: 'd30a9675-7404-11e5-8d04-005056b24375', Delivery1: '1', Delivery2: '1', ...}
    ▶ 12: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 13: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 14: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 15: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 16: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 17: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 18: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 19: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 20: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 21: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 22: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 23: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 24: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 25: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 26: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 27: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 28: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 29: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 30: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 31: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 32: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 33: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 34: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
    ▶ 35: {Description: 'Авратин', DescriptionRu: 'Авратин', Ref: '832eab07-1a52-11e9-a361-48df37b92096', Delivery1: '1', Delivery2: '1', ...}
```

Рис. 3.17 – JSON-відповідь API Нової Пошти зі списком відділень

Після заповнення форми користувач обирає спосіб оплати (рис 3.18). Якщо обрано накладений платіж, система просто зберігає замовлення у базу даних зі статусом `payed: false`. Якщо ж вибрано онлайн-оплату через Stripe (рис. 3.19), відбувається виклик POST-запиту на `/api/checkout/create-session`, який створює Stripe-сесію та перенаправляє користувача на сторінку оплати.

Накладений платіж - Відправлення здійснюється післяоплатою в 150 грн. При способі доставки післяоплатою при отриманні, Нова пошта додатково стягує комісію за відправку грошей назад 20грн +2%від суми замовлення.

☐ Накладений платіж

☒ Оплата за рахунком

Примітки до замовлень (необов'язково)

Нотатки до вашого замовлення, наприклад спеціальні примітки для доставки

ОФОРМИТИ ЗАМОВЛЕННЯ

Рис. 3.18 – Вибір способу оплати в інтерфейсі користувача

← New business sandbox Sandbox

Pay New business sandbox

UAH 51,998.00

Unnamed product UAH 27,999.00

Unnamed product UAH 23,999.00

Pay with link

Or

Email

Pavlyuk@gmail.com

Card information

4242 4242 4242 4242

03 / 27 121

Cardholder name

Pavlyuk Andriy

Country or region

Ukraine

☐ Securely save my information for 1-click checkout

Pay faster on New business sandbox and everywhere Link is accepted.

Pay

Powered by stripe

Terms Privacy

Рис. 3.19 – Інтерфейс Stripe Checkout з деталями замовлення

					ПЗ.КР.Б-121-25-ПЗ	Арк. 55
Змн.	Арк.	№ докум.	Підпис	Дата		

Після підтвердження форми дані передаються у форматі JSON на API /api/checkout/index.ts. Там здійснюється аутентифікація (JWT), валідація мови, перевірка структури orderData та збереження документа за допомогою Mongoose-моделі Order.

Документ включає ім'я, адресу, спосіб оплати, список товарів, статус оплати (payed), мову інтерфейсу та посилання на користувача (рис. 3.22).

New business sandbox

Pay New business sandbox

UAH 51,998.00

Unnamed product	UAH 27,999.00
Unnamed product	UAH 23,999.00

Pay with link

Or

Email

Pavlyuk@gmail.com

Card information

4242 4242 4242 4242

03 / 27 121

Cardholder name

Pavlyuk Andriy

Country or region

Ukraine

☐ Securely save my information for 1-click checkout

Pay faster on New business sandbox and everywhere Link is accepted.

Powered by

[Terms](#) [Privacy](#)

Рис. 3.20 – Приклад успішної відповіді від API оформлення замовлення

[Оформлення замовлення](#)

Дякуємо за ваш вибір!

Ми раді повідомити вас, що ваше замовлення прийняте і знаходиться в обробці. Наш менеджер зв'яжеться з вами найближчим часом для підтвердження та уточнення деталей.

ПРОДОВЖИТИ ПОКУПКИ

Рис. 3.21 – Перехід на сторінку успішного замовлення

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 3.22 – Документ замовлення у базі MongoDB з усіма параметрами замовлення

Таким чином, реалізація процесу оформлення замовлення включає повний цикл – від додавання товарів у кошик до підтвердження покупки через Stripe або накладений платіж, з інтеграцією служб доставки та збереженням усіх параметрів у базі даних. Ця функціональність є основою комерційної логіки платформи та забезпечує користувачу зручний, безпечний і надійний досвід онлайн-покупки.

3.4 Тестування інтернет магазину

На завершальному етапі реалізації платформи інтернет-магазину було проведено всебічне тестування продуктивності, ефективності, доступності та загальної оптимізації системи. Метою тестування є оцінити ступінь відповідності веб-додатку сучасним веб-стандартам та виявити можливі вузькі місця, які можуть негативно впливати на досвід користувача.

Для цього було використано два популярні інструменти – GTMetrix та Google PageSpeed Insights. Обидва сервіси є авторитетними рішеннями для аналізу веб-додатків і дозволяють отримати деталізовані звіти щодо основних

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

параметрів, зокрема таких як час першого завантаження (First Contentful Paint), час до повної взаємодії (Time to Interactive), загальний об'єм ресурсів, що завантажуються, затримки, ефективність структури DOM, доступність елементів тощо.

GTMetrix (<https://gtmetrix.com/>) надає поглиблений аналіз сторінки на основі Lighthouse та Web Vitals, а також генерує Waterfall Chart – діаграму запитів, що дозволяє проаналізувати завантаження всіх ресурсів сторінки. Інструмент особливо корисний для перевірки структури запитів, швидкості відповіді серверу, розміру ресурсів і блокувальних скриптів.

Google PageSpeed Insights (<https://pagespeed.web.dev/>) – це сервіс від Google, який надає комплексну оцінку продуктивності вебсторінок з точки зору як десктопних, так і мобільних користувачів. Основний акцент робиться на SEO-оптимізації, доступності, кращих практиках (Best Practices), а також параметрах, що впливають на ранжування сторінок у пошукових системах.

Результати тестування вказують на високу ефективність реалізованої системи. Як видно на рис. 3.23, за даними GTMetrix вебдодаток отримав оцінку А з продуктивністю 100% та структурною цілісністю 97%. Ключові метрики, такі як Largest Contentful Paint (401 мс), Total Blocking Time (0 мс) і Cumulative Layout Shift (0), свідчать про практично миттєве завантаження сторінки й повну відсутність критичних помилок візуалізації.

На рис. 3.24 зображено детальну вкладку “Structure”, яка показує список аудитів, що виконувались під час тестування. Виявлено лише одну рекомендацію середнього рівня важливості – усунення блокувальних ресурсів при рендерінгу, що є типовою проблемою для сучасних сайтів з великою кількістю зовнішніх шрифтів. Решта показників – на рівні Low або N/A, що вказує на високий рівень оптимізації коду та стилів.

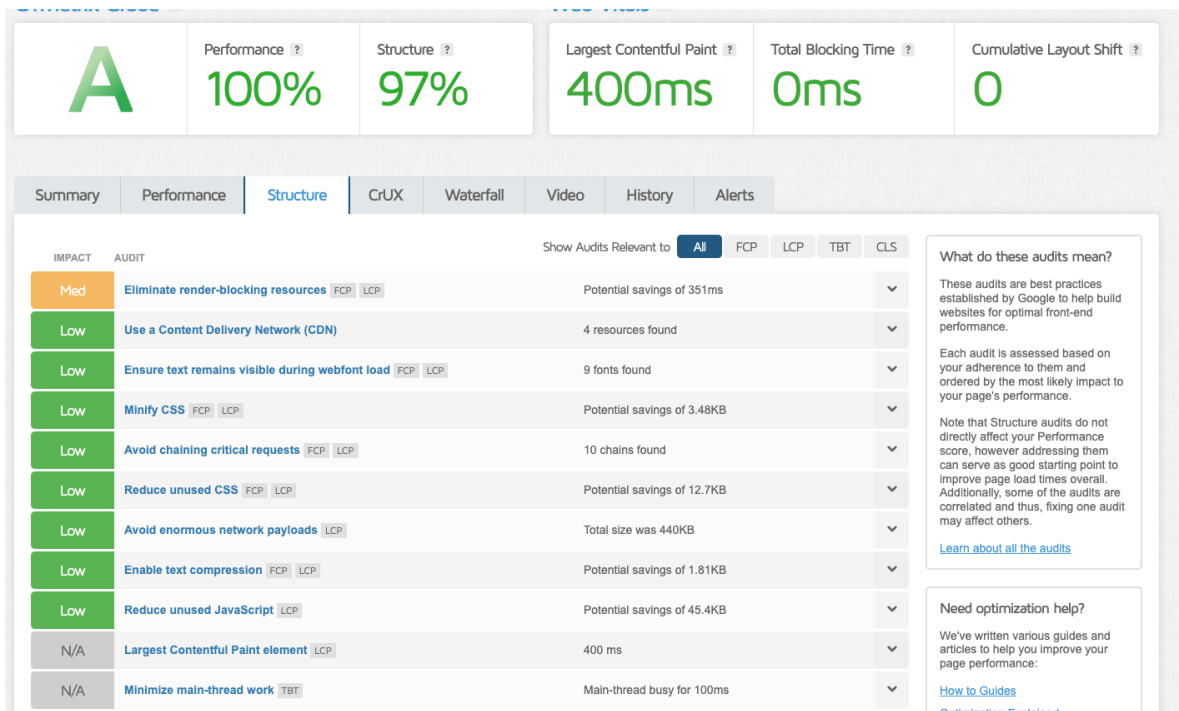


Рис. 3.23 – Аудит структури сторінки за результатами GTMetrix

Важливим аналітичним інструментом є “Waterfall Chart”, зображена на рис. 3.24. Вона демонструє часову шкалу завантаження всіх ресурсів сторінки. Найбільше навантаження припадає на зовнішні шрифти та скрипти, однак усі запити виконуються з мінімальними затримками (до 200 мс), що є показником стабільної серверної архітектури та оптимізації ресурсів.

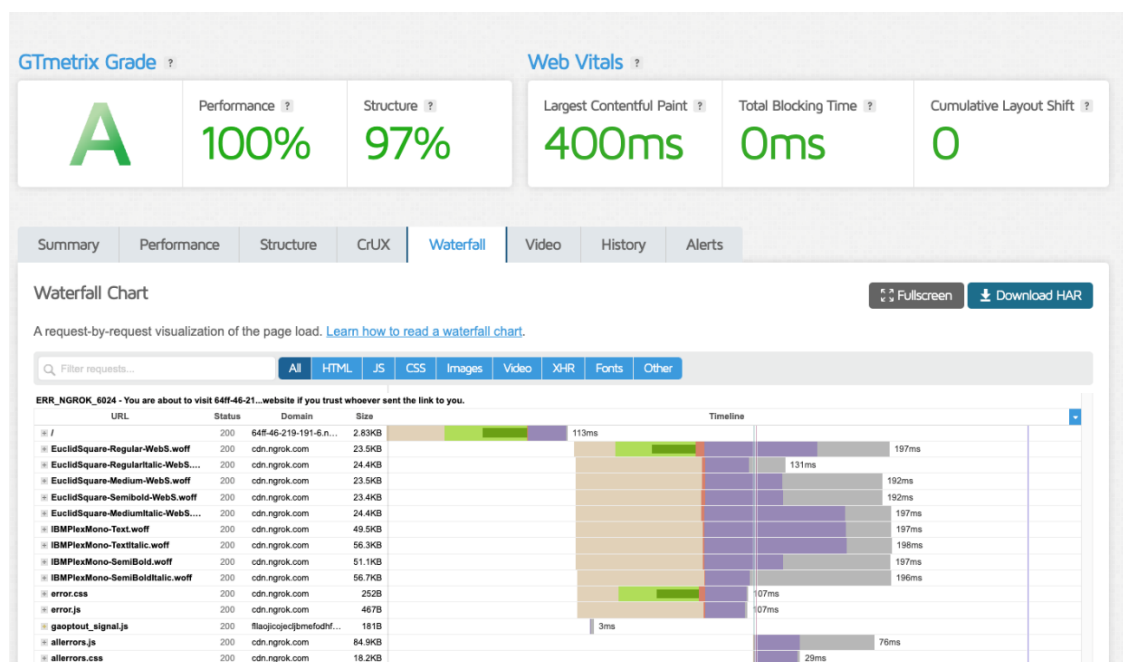


Рис. 3.24– Графік завантаження ресурсів (Waterfall Chart)

Також було проаналізовано вкладку “Performance” (рис. 3.25), яка показує такі показники як Time to Interactive, Speed Index, First Contentful Paint. Всі метрики знаходяться в межах до 400 мс, що вказує на відмінну продуктивність і оптимізацію клієнтської частини програми. Повна відсутність блокувань або затримок свідчить про ефективне використання сучасних практик асинхронного рендерінгу.

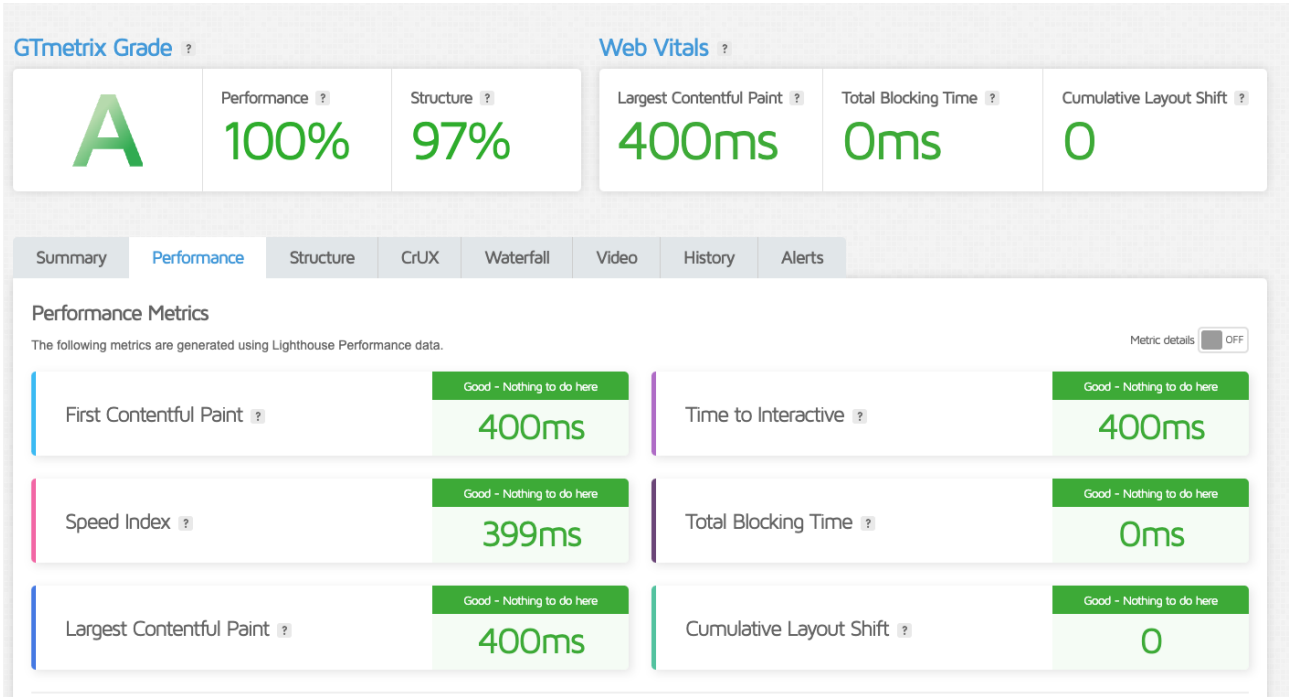


Рис. 3.25 – Оцінка продуктивності на основі Lighthouse в GTMetrix

Крім того, для додаткового порівняння було використано Google PageSpeed Insights. Результати зображено на Рис. 3.26. Платформа отримала 97 балів за ефективність і 94 за доступність. Низький показник SEO (54) пояснюється відсутністю конфігурацій SEO на локальному хості, зокрема генерації sitemap.xml, robots.txt, канонічних URL тощо. Цей недолік не є критичним для MVP-версії платформи й може бути усунений під час її розгортання у production.

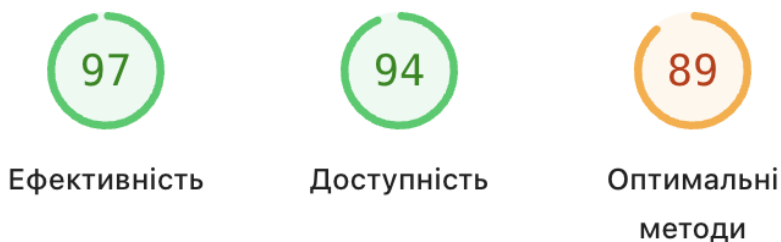


Рис. 3.26 – Оцінка Google PageSpeed Insights (десктопна версія)

У підсумку, результати тестування підтвердили високий рівень оптимізації вебдодатку. Вебсайт має чудову швидкість завантаження, мінімальні затримки у взаємодії з користувачем і коректну побудову структури сторінок. Це свідчить про технічну зрілість продукту та готовність до розгортання у продакшн-середовищі.

На рис. зображено головну сторінку інтернет-магазину ByteBox у момент запуску Lighthouse-аудиту в браузері Chrome. Праворуч представлено результати тестування за ключовими метриками, що оцінюють якість вебзастосунку з позицій ефективності, доступності, кращих практик та оптимізації під пошукові системи (SEO).

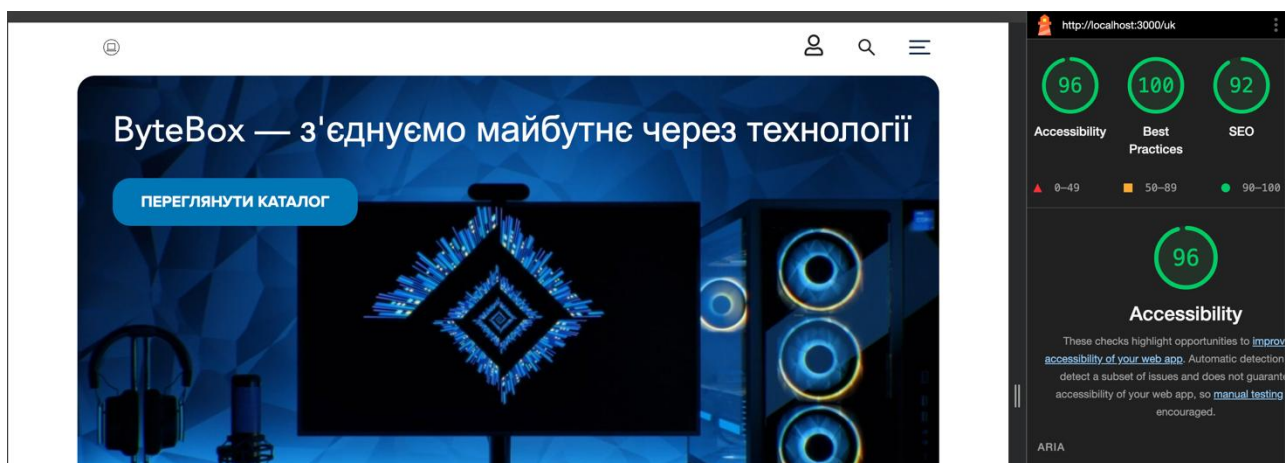


Рис. 3.27 –Аудит інтернет-магазину ByteBox за допомогою Lighthouse

Accessibility – 96 балів із 100. Це свідчить про те, що більшість елементів

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

сайту коректно доступні для користувачів з обмеженими можливостями. Було враховано контрастність кольорів, доступність фокусування елементів, альтернативні описи зображень тощо.

Best Practices – 100 балів. Це вказує на відсутність помилок безпеки, використання сучасних стандартів JavaScript та правильну організацію роботи з HTTP-запитами.

SEO – 92 бали. Незначне зниження показника пов'язане з відсутністю деяких SEO-мета тегів, оскільки сайт наразі запущений лише на локальному сервері і не має реального домену.

На основі результатів тестування, проведеного за допомогою сервісів GTMetrix та Google PageSpeed Insights, можна сформулювати низку рекомендацій для подальшої оптимізації продуктивності вебдодатку. Зокрема, серед найбільш поширених порад, що надали обидва сервіси, були: усунення блокуючих ресурсів рендерингу, використання CDN (Content Delivery Network) для прискорення доставки статичних ресурсів, зменшення обсягу використаних CSS і JavaScript, а також уникнення надмірного ланцюгового завантаження залежностей. Додатково, рекомендується реалізувати стратегії асинхронного завантаження шрифтів та ресурсів, оптимізувати порядок їх підключення та перевірити, які ресурси реально використовуються у перші секунди взаємодії з користувачем. Ці кроки дозволять скоротити Largest Contentful Paint (LCP), покращити Speed Index, зменшити Total Blocking Time (TBT) та інші показники, що мають вирішальне значення для UX і SEO. Сервіс GTMetrix надає візуалізовану «waterfall» діаграму, яка дозволяє розробникам детально відслідковувати процес завантаження кожного ресурсу і приймати технічно обґрунтовані рішення для вдосконалення продуктивності.

Отже, вебзастосунок відповідає сучасним вимогам до якості коду, зручності використання та доступності, що є важливими показниками для підтримки надійної роботи сайту у виробничому середовищі. Дотримання рекомендацій Lighthouse дозволяє гарантувати позитивний користувацький досвід, а також забезпечити

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

технічну готовність до подальшого SEO-просування.

Висновки до третього розділу

У третьому розділі роботи було надано вичерпний опис практичної реалізації інтернет-магазину, що функціонує як повноцінна e-commerce платформа. Кожен етап створення було обґрунтовано архітектурно, функціонально та технологічно.

Розгляд інтерфейсу користувача дозволив встановити принципи адаптивності, зручності та ефективності навігації. Було продемонстровано реалізацію ключових модулів – від виводу продукції до реалізації кошика, оформлення замовлення та підключення Stripe для оплати. Описано процеси взаємодії з бекендом та зовнішніми API служб доставки, що забезпечило гнучкість і масштабованість розробленої системи.

Особливу увагу приділено захисту персональних даних, валідації форм, обробці замовлень та збереженню даних у базі MongoDB. Також у цьому розділі проаналізовано ефективність роботи вебдодатку через тестування з використанням сервісів GTMetrix та PageSpeed. Результати демонструють високі показники продуктивності, що свідчить про правильність обраних рішень на етапах верстки, оптимізації та реалізації логіки додатку.

Таким чином, третій розділ підтвердив, що створений інтернет-магазин є технічно повноцінним, функціонально завершеним та готовим до подальшого впровадження, масштабування або вдосконалення відповідно до бізнес-вимог.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано інтернет-магазин, що забезпечує повний цикл взаємодії користувача з системою – від перегляду товарів до оформлення та оплати замовлення. На основі аналізу функціональних вимог до сучасних e-commerce рішень було визначено ключові компоненти платформи, архітектуру взаємодії клієнтської та серверної частини, а також вимоги до інтерфейсу та зручності використання.

Магазин створений з використанням сучасного технологічного стеку: Next.js, TypeScript, SCSS-модулі, що дозволяє забезпечити високу продуктивність, підтримку серверного рендерингу та адаптивну верстку. Для збереження та обробки даних використано MongoDB, а для взаємодії з базою – бібліотеку Mongoose, що забезпечує чітку типізацію моделей та зв'язки між сутностями. У ході реалізації проєкту також створено реєстрацію, авторизацію, профіль користувача з можливістю редагування особистих даних, інтеграцію із платіжною системою Stripe, а також можливість обрати доставку за допомогою API поштових служб.

Особливу увагу приділено тестуванню продуктивності: за допомогою Google PageSpeed Insights та GTMetrix було проаналізовано швидкість завантаження, індекси інтерфейсної взаємодії, використання ресурсів і потенційні шляхи оптимізації. Результати підтвердили високий рівень ефективності платформи, що демонструє її придатність до розгортання в реальному середовищі.

Таким чином, розроблений інтернет-магазин є повноцінним, масштабованим і адаптивним рішенням, яке відповідає сучасним вимогам до вебзастосунків у сфері електронної комерції. Отриманий функціонал може слугувати основою для подальшого розвитку: додавання адміністративної панелі, персоналізації товарних пропозицій, впровадження багатомовності, аналітики та розширених маркетингових функцій.

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-застосунки на JavaScript / Алекс Маккоу, 2012, 283 с.
2. Алло – Офіційний сайт [Електронний ресурс] – Режим доступу до ресурсу: <https://allo.ua>
3. Комп'ютерні мережі: захист інформації: навчальний посібник / Березін О.Є., Гордієнко Л.Є., Іванченко Є.О. , Київ: видавництво “Кондор”, 2017.
4. Бондаренко С. П., Коваль В. І. Інформаційні системи і технології в економіці: навч. посіб. – К.: Центр учбової літератури, 2021. – 352 с.
5. Захист інформації на комп'ютерах і мережах: підручник. / Борисенко В.О., Бондаренко О.І., Железняк І.О. – Київ: Видавництво “Наукова думка”, 2019.
6. Організація баз даних і знань. Мови баз даних: Конспект лекцій. / Гайна Г.А. – Київ: КНУБА , 2002. – 64 с.
7. Організація баз даних і знань. Організація реляційних баз даних: Конспект лекцій. / Гайна Г.А. , Попович Н.Л. – Київ: КНУБА , 2000. – 76 с.
8. Самовчитель HTML / Гончаров А. 2002. – 240 с.
9. Методи та засоби захисту інформації в комп'ютерних системах: навчальний посібник. / Грінченко С.В., Мартинюк Ю.В., Щербак С.О. – Київ: Видавничий центр КНЕУ , 2018.
10. JavaScript. Біблія користувача. / Гурман Д. , 2002. – 958 с.
11. Документація MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs>
12. Документація Next.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nextjs.org/docs>
13. Офіційна документація Нова Пошта API [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.novaposhta.ua>
14. Проєктування інформаційних систем: навч. посіб. / Сидоренко М. В. – Харків: ФОП Бровін О.В. , 2020. – 295 с.

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

15. Стаття про покращення якості сервісу. [Електронний ресурс] – Режим доступу до ресурсу: <https://service.yaware.com.ua/blog/yak-pokrashhyty-yakist-servisu/>

16. Практика розробки Full-Stack додатків на JavaScript. / Ткаченко А. А. – Харків: СумДУ, 2023. – 216 с.

17. ТОП 10 загроз кібербезпеці бізнесу у 2023 році. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bdo.ua/uk-ua/insights-2/information-materials/2023/top-10-cybersecurity-threats-to-businesses-in-2023>

18. Захист інформації в комп'ютерних системах: підручник. / Шилова, Ю. А. – Київ: Видавничо-поліграфічний центр “Київський університет”, 2016.

19. Розробка вебзастосунків з використанням фреймворку Next.js / Ярошук І.О. – Вісник НТУУ «КПІ». Інформатика та обчислювальна техніка. , 2022. – 53с.

20. “The Complete React Developer Course (Hooks and Redux)” / Andrew Mead. – Udemy , 2020.

21. API Укрпошти – документація [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ukrposhta.ua/ua/api>

22. Clientsidetechnologies. [Електронний ресурс] – Режим доступу до ресурсу: https://w3techs.com/technologies/overview/client_side_language

23. Comfy – Офіційний сайт [Електронний ресурс] – Режим доступу до ресурсу: <https://comfy.ua>

24. “Node.js Web Development: Server-side development with Node 10 made easy”. / David Herron. – Packt Publishing , 2018.

25. Douglas Crockford’s JavaScript. [Електронний ресурс] – Режим доступу до ресурсу: <http://crockford.com/javascript/>.

26. What Is A Web Server And How Does It Function? [Електронний ресурс] / Pravin Ganore. 2017. – Режим доступу до ресурсу: <https://www.milesweb.com/blog/hosting/web-server-function/>

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

27. HTML. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.w3.org/TR/html401>
28. HTML&CSS. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.w3.org/standards/webdesign/htmlcss>
29. Nextjs 15 Todo tutorial 2025 [Електронний ресурс] – Режим доступу до ресурсу: https://medium.com/@lior_amsalem/nextjs-15-todo-tutorial-2025-89b3f31546e0
30. ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.
31. MDN Web Docs. JavaScript Guide [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
32. Data Structures and Algorithms with JavaScript / Michael McMillan , 2014. – 246 с.
33. MongoDB. [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/nosql/mongodb/1.1.php>
34. Top 8 tech stacks. [Електронний ресурс] – Режим доступу до ресурсу: <https://fullscale.io/blog/top-5-tech-stacks/>
35. Most used languages stats. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>
36. MOYO – Офіційний сайт. [Електронний ресурс] – Режим доступу до ресурсу: <https://moyo.ua>
37. Rozetka – Офіційний сайт. [Електронний ресурс] – Режим доступу до ресурсу: <https://rozetka.com.ua>
38. Stripe API Documentation. [Електронний ресурс] – Режим доступу до ресурсу: <https://stripe.com/docs/api>

					ПЗ.КР.Б-121-25-ПЗ	Арк. 67
Змн.	Арк.	№ докум.	Підпис	Дата		

39. The Software Quality Company. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tiobe.com/tiobe-index/>

40. Trends in JS development. [Електронний ресурс] – Режим доступу до ресурсу: <https://trends.builtwith.com/docinfo/Javascript>

41. Yup doc. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jsdocs.io/package/yup>

42. UAA Team. Тренди та виклики українського ринку eCommerce у 2024 році. [Електронний ресурс] – Режим доступу до ресурсу: <https://uaateam.agency/blog/trendy-ta-vyklyky-ukrainskogo-rynku-ecommerce>

43. Web Server and its Types of Attacks 2012. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.greycampus.com/opencampus/ethical-hacking/web-server-and-its-types-of-attacks>

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

**Порівняльна характеристика популярних аналогів
інтернет-магазинів в Україні**

Платформа	Тип платформи	Ключовий функціонал	Переваги	Недоліки
Rozetka	Маркетплейс із широким асортиментом	Каталог із фільтрами, відеоогляди, 360° зображення, порівняння товарів, рейтинг, бонусна програма, мобільний застосунок, особистий кабінет	Широкий асортимент, висока довіра користувачів, стабільність роботи, зручна мобільна версія, швидка доставка	Перевантажений інтерфейс, надлишок рекламних блоків, складна конкуренція між продавцями
Comfy	Спеціалізований інтернет-магазин техніки та електроніки	Простий і зрозумілий каталог, онлайн-консультант, бонуси, швидке оформлення замовлення, інтеграція з оплатою частинами	Сучасний UX/UI, вигідні акції, швидкість обслуговування, орієнтація на якість	Обмежений вибір товарів, відсутність мультивендорної моделі, залежність від власного складу
MOYO	Онлайн-магазин електроніки з фокусом на сервіс	Відеоогляди, структуровані технічні характеристики, огляди, форма зворотного зв'язку, просте замовлення	Якісна подача товарів, орієнтація на клієнта, прозорі умови доставки, зрозумілий інтерфейс	Менш відомий бренд, обмежений асортимент, відсутність маркетплейс-функцій
Алло	Гібридна платформа: інтернет-магазин + маркетплейс	Широка товарна лінійка, відеоогляди, оплата частинами, підтримка 24/7, онлайн-чат, мобільний застосунок, просування через соцмережі	Сильна маркетингова присутність, зручна система оплати, мультимедійність платформи, гнучкий підхід до комунікації	Навігація іноді перевантажена, багато рекламних елементів, складна структура категорій

Джерело: створено автором на основі аналізу створено автором на основі [2, 25, 39, 40]

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

API-обробник оформлення замовлення

```

const handler = async (req: NextApiRequest, res: NextApiResponse) => {
  await connect();

  await authenticate(req, res, async () => {
    const authedReq = req as AuthRequest;

    const lang = (authedReq.query.lang as string)?.toUpperCase();
    if (!["UA", "EN"].includes(lang)) {
      return res.status(400).json({ message: "Unsupported language" });
    }

    if (authedReq.method !== "POST") {
      return res.status(405).json({ message: "Method Not Allowed" });
    }

    try {
      const orderData = authedReq.body;

      if (!orderData || typeof orderData !== "object" || !orderData.customer) {
        return res.status(400).json({ message: "Invalid order data" });
      }

      const {
        customer: {
          firstName,
          lastName,
          phone,
          email,
          address,
          comment,
          payment,
          delivery,
        },
        items,
        total,
        payed,
        date,
      } = orderData;

      const newOrder = new Order({
        firstName,
        lastName,
        phone,
        email,
        address,
        comment,
        payment,
        delivery,
      });
    }
  });
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    items,
    total,
    payed,
    date,
    lang,
    user: authedReq.user._id,
  });

  await newOrder.save();

  return res.status(200).json({
    message: "Order created successfully",
    id: newOrder._id,
  });
} catch (error) {
  console.error("✖ Error saving order:", error);
  return res.status(500).json({ message: "Failed to process order" });
}
});
};

export default ctrlWrapper(handler);

```

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

Компонент картки товару на сторінці каталогу (ProductCard.tsx)

```

interface ProductCardProps extends ComponentDetailsI {
  deleteButtonText: string;
  itemDeletedToast: string;
  handleDelete: () => void;
  descriptionPropertyNames: ProductDescription;
}

const ProductCard: React.FC<ProductCardProps> = ({
  id,
  title,
  price,
  images,
  quantity,
  description,
  category,
  slug,
  deleteButtonText,
  itemDeletedToast,
  handleDelete,
}) => {
  const { deleteCartItem } = useCartActionsContext();
  const router = useRouter();
  const lang = useLangFromPathname();

  const handleRemove = () => {
    deleteCartItem({ id, category, toastMessage: itemDeletedToast });

    const cart = localStorage.getItem("cartProducts");
    if (cart) {
      const parsed = JSON.parse(cart) as Record<
        "phone" | "laptop" | "component",
        { id: string }[]
      >;

      const isEmpty = Object.values(parsed).every(
        (items) => Array.isArray(items) && items.length === 0
      );

      if (isEmpty) {
        router.push(`/${lang}`);
      }
    }

    handleDelete();
  };

```

```

return (
  <li className={styles.card}>
    <Link href={`/${lang}/${slug}/${id}`}>
      <div className={styles.card__img_container}>
        <Image
          src={images[0]}
          alt={title}
          fill
          priority
          sizes="(min-width: 768px) 150px, 100px"
        />
      </div>
    </Link>

    <div className={styles.card__block}>
      <div className={styles.card__content}>
        <Link href={`/${lang}/${slug}/${id}`}>
          <Typography variant="bodyRegular" className={styles.card__title}>
            {title}
          </Typography>
        </Link>

        {description && (
          <Typography
            variant="bodyRegular"
            className={styles.card__description}
            color="var(--cl-gray-500)"
          >
            {description}
          </Typography>
        )}

        <Price price={price} />
      </div>

      <div className={styles.card__actions}>
        <ComponentQuantity
          id={id}
          qty={quantity}
          isCartQuantity
          category={category}
          className={styles.buttonGroup}
        />
        <button type="button" onClick={handleRemove}>
          <Typography variant="bodyS" className={styles.delete}>
            {deleteButtonText}
          </Typography>
        </button>
      </div>
    </div>
  </li>
);

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

Контекст управління кошиком (CartContext.tsx)

```
interface CartContextI {
  cartProducts: CartProducts;
  totalCartProducts: number;
  cartTotalPrice: number;
}

interface IToggleQuantityParams {
  id: string;
  value: typeof INCREMENT | typeof DECREMENT;
  category: keyof CartProducts;
}

interface IDeleteCartItemParams {
  id: string;
  category: keyof CartProducts;
  toastMessage: string;
}

interface CartActionsContextProps {
  addComponentsToCart: (params: ComponentCartParams) => void;
  addPhonesToCart: (params: PhoneOrLaptopCartParams) => void;
  addLaptopsToCart: (params: PhoneOrLaptopCartParams) => void;
  toggleQuantity: (params: IToggleQuantityParams) => void;
  deleteCartItem: (params: IDeleteCartItemParams) => void;
  clearCartProducts: () => void;
}

const CartContext = createContext<CartContextI | null>(null);
const CartActionsContext = createContext<CartActionsContextProps | null>(null);

export const CartContextProvider = ({ children }: { children: ReactNode }) => {
  const [cartProducts, setCartProducts] = useLocalStorage<CartProducts>(
    "cartProducts",
    getInitialCartProducts()
  );

  const getTotalQuantity = (items: CartItem[]) =>
    items.reduce((acc, item) => acc + item.quantity, 0);

  const getTotalPrice = (items: CartItem[]) =>
    items.reduce((acc, item) => acc + item.quantity * item.price, 0);

  const totalCartProducts =
    getTotalQuantity(cartProducts.component) +
    getTotalQuantity(cartProducts.phone) +
    getTotalQuantity(cartProducts.laptop);

  const cartTotalPrice =
    getTotalPrice(cartProducts.component) +
```

```

    getTotalPrice(cartProducts.phone) +
    getTotalPrice(cartProducts.laptop);

const addComponentsToCart = useCallback(
  ({
    id,
    slug,
    category,
    toastMessage,
    price,
    quantity = 1,
  }: ComponentCartParams) => {
    setCartProducts((prevItems) => {
      const isComponentInCart = prevItems.component.find(
        (item) => item.id === id && item.slug === slug
      );
      const updatedItems = isComponentInCart
        ? {
            ...prevItems,
            component: prevItems.component.map((item) =>
              item.id === id && item.slug === slug
                ? { ...item, quantity: item.quantity + quantity }
                : item
            ),
          }
        : {
            ...prevItems,
            component: [
              ...prevItems.component,
              { id, slug, category, quantity, price },
            ],
          };

      return updatedItems;
    });

    showToast(`${toastMessage}`);
  },
  [setCartProducts]
);

const addPhonesToCart = useCallback(
  ({
    id,
    slug,
    category,
    toastMessage,
    price,
    quantity = 1,
  }: PhoneOrLaptopCartParams) => {
    setCartProducts((prevItems) => {

```

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const isPhoneInCart = prevItems.phone.find((item) => item.id === id);
const updatedItems = isPhoneInCart
? {
  ...prevItems,
  phone: prevItems.phone.map((item) =>
    item.id === id
    ? { ...item, quantity: item.quantity + quantity }
    : item
  ),
}
: {
  ...prevItems,
  phone: [
    ...prevItems.phone,
    { id, slug, category, quantity, price },
  ],
};
return updatedItems;
});
showToast(`${toastMessage}`);
},
[setCartProducts]
);

```

```

const addLaptopsToCart = useCallback(
({
  id,
  slug,
  category,
  toastMessage,
  price,
  quantity = 1,
}: PhoneOrLaptopCartParams) => {
  setCartProducts((prevItems) => {
    const isLaptopInCart = prevItems.laptop.find((item) => item.id === id);
    const updatedItems = isLaptopInCart
    ? {
      ...prevItems,
      laptop: prevItems.laptop.map((item) =>
        item.id === id
        ? { ...item, quantity: item.quantity + quantity }
        : item
      ),
    }
    : {
      ...prevItems,
      laptop: [
        ...prevItems.laptop,
        { id, slug, category, quantity, price },
      ],
    };
  });
}

```

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return updatedItems;
    });

    showToast(`${toastMessage}`);
  },
  [setCartProducts]
);

const toggleQuantity = useCallback(
  ({ id, value, category }: IToggleQuantityParams) => {
    setCartProducts((prev) => {
      const updatedItems = prev[category].map((item) => {
        if (item.id !== id) return item;
        const newQuantity =
          value === INCREMENT
            ? item.quantity + 1
            : Math.max(item.quantity - 1, 1);
        return { ...item, quantity: newQuantity };
      });
      return { ...prev, [category]: updatedItems };
    });
  },
  [setCartProducts]
);

const deleteCartItem = useCallback(
  ({ id, category, toastMessage }: IDeleteCartItemParams) => {
    setCartProducts((prev) => {
      const updatedItems = prev[category].filter((item) => item.id !== id);
      return { ...prev, [category]: updatedItems };
    });
    showToast(toastMessage);
  },
  [setCartProducts]
);

const clearCartProducts = useCallback(() => {
  setCartProducts(getInitialCartProducts());
}, [setCartProducts]);

const contextValue = useMemo(
  () => ({
    cartProducts,
    totalCartProducts,
    cartTotalPrice,
  }),
  [cartProducts, totalCartProducts, cartTotalPrice]
);

return (
  <CartContext.Provider value={contextValue}>

```

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    <CartActionsContext.Provider
      value={{
        addComponentsToCart,
        addPhonesToCart,
        addLaptopsToCart,
        toggleQuantity,
        deleteCartItem,
        clearCartProducts,
      }}
    >
      {children}
    </CartActionsContext.Provider>
  </CartContext.Provider>
);
};

export const useCartContext = () => {
  const context = useContext(CartContext);
  if (!context) {
    throw new Error("useCartContext must be used within a CartContextProvider");
  }
  return context;
};

export const useCartActionsContext = () => {
  const context = useContext(CartActionsContext);
  if (!context) {
    throw new Error(
      "useCartActionsContext must be used within a CartContextProvider"
    );
  }
  return context;
};

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтеграція з платіжною системою Stripe

```

import Stripe from "stripe";
import type { NextApiRequest, NextApiResponse } from "next";

const stripe = new Stripe(process.env.STRIPE_SECRET_KEY as string);

export default async function handler(
  req: NextApiRequest,
  res: NextApiResponse
) {
  if (req.method !== "POST") {
    return res.status(405).json({ message: "Method Not Allowed" });
  }

  try {
    const { items } = req.body;

    const session = await stripe.checkout.sessions.create({
      payment_method_types: ["card"],
      mode: "payment",
      line_items: items.map((item: any) => ({
        price_data: {
          currency: "uah",
          product_data: {
            name: item.title || item.name || "Unnamed product",
          },
          unit_amount: item.price * 100,
        },
        quantity: item.quantity,
      })),
      success_url: `${req.headers.origin}/uk/success-order`,
      cancel_url: `${req.headers.origin}/uk/checkout/error`,
    });

    return res.status(200).json({ url: session.url, id: session.id });
  } catch (error: any) {
    console.error("Stripe session creation error:", error.message);
    return res.status(500).json({ message: "Stripe session creation failed" });
  }
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Е.1. Операції Redux для оновлення профілю користувача

```

export const register = createAsyncThunk<
  AuthResponse,
  RegisterParams,
  { rejectValue: RejectValue }
>("auth/register", async (credentials, { rejectWithValue }) => {
  try {
    const response = await apiRegister(credentials);
    Notiflix.Notify.success("Реєстрацію завершено успішно");
    return response;
  } catch (error: any) {
    if (
      error?.message?.includes("409") ||
      error?.message === "Registration failed"
    ) {
      Notiflix.Notify.failure("Користувач з таким email вже існує");
      return rejectWithValue({
        status: 409,
        message: "User already exists",
      });
    } else {
      Notiflix.Notify.failure("Помилка реєстрації");
      return rejectWithValue({
        status: 500,
        message: error.message || "Unknown registration error",
      });
    }
  }
});

export const login = createAsyncThunk<
  AuthResponse,
  LoginParams,
  { rejectValue: RejectValue }
>("auth/login", async (credentials, { rejectWithValue }) => {
  try {
    const response = await apiLogin(credentials);
    Notiflix.Notify.success("Вхід виконано успішно");
    return response;
  } catch (error: any) {
    Notiflix.Notify.failure("Помилка входу. Перевірте email або пароль");
    return rejectWithValue({
      status: 401,
      message: error.message || "Invalid login credentials",
    });
  }
});

```

					ПЗ.КР.Б-121-25-ПЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

export const logout = createAsyncThunk<
  void,
  string,
  { rejectValue: RejectValue }
>("auth/logout", async (token, { rejectWithValue }) => {
  try {
    await apiLogout(token);
    Notiflix.Notify.success("Вихід виконано успішно");
  } catch (error: any) {
    Notiflix.Notify.failure("Не вдалося вийти з облікового запису");
    return rejectWithValue({
      status: 403,
      message: error.message || "Logout failed",
    });
  }
});

```

```

export const current = createAsyncThunk<
  CurrentUserResponse,
  string,
  { rejectValue: RejectValue }
>("auth/current", async (token, { rejectWithValue }) => {
  try {
    const response = await getCurrent(token);
    console.log(response);
    return response;
  } catch (error: any) {
    return rejectWithValue({
      status: 403,
      message: error.message || "Unable to fetch user",
    });
  }
});

```

Додаток Е. 2. Слайс Redux для керування автентифікацією користувача

```

const authSlice = createSlice({
  name: "auth",
  initialState,
  reducers: {},
  extraReducers: (builder) => {
    builder
      .addCase(register.fulfilled, (state, { payload }) => {
        state.isLoading = false;
        state.user = payload.user;
        state.token = payload.accessToken;
        state.isLoggedIn = true;
        state.error = null;
      })
  }
});

```

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

.addCase(login.fulfilled, (state, { payload }) => {
  state.user = payload.user;
  state.token = payload.accessToken;
  state.isLoggedIn = true;
  state.error = null;
})
.addCase(current.pending, (state) => {
  state.isRefreshing = true;
  state.isLoading = true;
})
.addCase(current.fulfilled, (state, { payload }) => {
  state.user = payload;
  state.isLoggedIn = true;
  state.isRefreshing = false;
  state.error = null;
})
.addCase(register.rejected, (state) => {
  state.isLoading = false;
})
.addCase(logout.fulfilled, (state) => {
  state.user = null;
  state.token = null;
  state.isLoggedIn = false;
  state.error = null;
})
.addMatcher(
  isAnyOf(register.rejected, login.rejected, logout.rejected),
  (state, { payload }: any) => {
    state.error = payload || {
      status: 500,
      message: "Authentication error",
    };
    state.isLoading = false;
  }
);
},
});

export const authReducer = authSlice.reducer;

```

Додаток Е. 3. Конфігурація Redux Store

```

const authPersistConfig = {
  key: "auth",
  storage,
  whitelist: ["token"],
};

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

```
const persistedAuthReducer = persistReducer(authPersistConfig, authReducer);
```

```
export const store = configureStore({
  reducer: {
    auth: persistedAuthReducer,
    user: userReducer,
  },
  middleware: (getDefaultMiddleware) =>
    getDefaultMiddleware({
      serializableCheck: {
        ignoredActions: [
          FLUSH,
          REHYDRATE,
          PAUSE,
          PERSIST,
          PURGE,
          REGISTER,
          "user/updateUser/pending",
          "user/updateUser/fulfilled",
          "user/updateUser/rejected",
        ],
        ignoredActionPaths: ["meta.arg", "payload.user.userInfo"],
        ignoredPaths: ["user.user.userInfo"],
      },
    }),
});
```

```
export const persistor = persistStore(store);
```

```
export type RootState = ReturnType<typeof store.getState>;
```

```
export type AppDispatch = typeof store.dispatch;
```

```
export default store;
```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		