

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 в'язаного одягу»

Виконала студентка групи ПЗ-22-3
КОЛМИКОВА Амалія Сергіївна

Керівник роботи:
МОРОЗОВ Андрій Васильович

Рецензент:
ЛОБАНЧИКОВА Надія Миколаївна

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
«28» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **КОЛМИКОВА Амалия Сергіївна**

Керівник роботи: **МОРОЗОВ Андрій Васильович**

Тема роботи: **«Інтернет-магазин в'язаного одягу»**,

затверджена Наказом закладу вищої освіти від **«27» лютого 2026 р., № 92/с**

Вихідні дані для роботи: процес автоматизації роздрібної торгівлі та взаємодії між майстром та кінцевими споживачами, який використовує веб-платформу для візуалізації каталогу товарів, конфігурації індивідуальних параметрів виробів та підвищення ефективності збуту продукції.

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Морозов А.В.	01.03.2026р.	01.03.2026р
2	Морозов А.В.	01.03.2026р	01.03.2026р
3	Морозов А.В.	01.03.2026р	01.03.2026р

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з інтернет-магазину в'язаного одягу та пояснювальної записки. Пояснювальна записка до випускної роботи містить 89 сторінок, 40 ілюстрацій та 12 таблиць.

Метою роботи є розробка інтернет-магазину для автоматизації взаємодії між майстром та покупцями в'язаного одягу з використанням каталогу товарів та інструментів кастомізації виробів для візуалізації асортименту та оформлення індивідуальних замовлень у реальному часі.

Архітектура сервісу базується на поєднанні серверної частини на базі PHP та клієнтського рішення на бібліотеці React. Використання технології SSR забезпечує високу швидкість завантаження сторінок товарів та ефективну SEO-оптимізацію, що є критичним для просування бренду.

Додаток має інтуїтивно зрозумілий інтерфейс із застосуванням сучасних UI-патернів, що дозволяє користувачам зручно переглядати каталог товарів та створювати замовлення.

КЛЮЧОВІ СЛОВА: E-COMMERCE, ІНТЕРНЕТ-МАГАЗИН, CLOUD STORAGE, POSTGRESQL, ОНЛАЙН-ОПЛАТА, В'ЯЗАНИЙ ОДЯГ.

					ІПЗ.КР.Б – 121 – 26 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Інтернет-магазин в'язаного одягу	Літ.	Арк.	Аркушів
Розроб.		Колмикова А.С.						
Перевір.		Морозов А.В.					3	89
Керівник						Житомирська політехніка, група ІПЗ-22-3		
Н. контр.								
Зав. каф.		Вакалюк Т.А.						

ABSTRACT

This bachelor's thesis consists of an online store for knitted clothing and an explanatory note. The explanatory note accompanying the thesis comprises 89 pages, 40 illustrations, and 12 tables.

The aim of the work is to develop an online store to automate the interaction between the artisan and buyers of knitted clothing, utilizing a product catalog and product customization tools to visualize the assortment and place individual orders in real time.

The service architecture is based on a combination of a PHP-based backend and a client-side solution using the React library. The application of SSR technology ensures high loading speeds for product pages and effective SEO optimization, which is critical for brand promotion.

The application features an intuitive interface employing modern UI patterns, allowing users to conveniently browse the product catalog and place orders.

KEYWORDS: E-COMMERCE, ONLINE STORE, CLOUD STORAGE, POSTGRESQL, ONLINE PAYMENT, KNITTED CLOTHING

					ІПЗ.КР.Б – 121 – 26 – ІПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Колмикова А.С.			Інтернет-магазин в'язаного одягу			Літ.	Арк.	Аркушів	
Перевір.		Морозов А.В.								4	89
Керівник								Житомирська політехніка, група ІПЗ-22-3			
Н. контр.											
Зав. каф.		Вакалюк Т.А.									

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИБІР ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ.....	10
1.1. Постановка задачі	10
1.2. Аналіз аналогів програмного продукту.....	12
1.3. Обґрунтування вибору інструментальних засобів	18
Висновки до першого розділу	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ ІНТЕРНЕТ-МАГАЗИНУ В'ЯЗАНОГО ОДЯГУ	23
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи.....	23
2.2. Розробка бази даних системи	27
2.3. Проєктування та реалізація алгоритмів роботи системи.....	44
2.4. Реалізація функціоналу	48
Висновки до другого розділу.....	52
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ІНТЕРНЕТ-МАГАЗИНОМ В'ЯЗАНОГО ОДЯГУ	54
3.1. Порядок розгортання та налаштування параметрів платформи.....	54
3.2. Структура інтерфейсу магазину.....	57
3.3. Тестування вебплатформи	70
Висновки до третього розділу	75
ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	77
ДОДАТКИ.....	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API - прикладний програмний інтерфейс

БД - база даних

ПЗ - програмне забезпечення

СУБД - система управління базами даних

HTTP / HTTPS - протокол передавання гіпертексту

JSON - текстовий формат обміну даними

SEO - пошукова оптимізація

SSR - рендеринг на стороні сервера

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми зумовлена тим, що на сьогоднішній день брендів одягу та різноманітних пропозицій стало дуже багато, через що створення власного вебсайту є важливим кроком для роботи онлайн та залучення більшої кількості клієнтів. У сучасних умовах покупці витрачають на пошук якісного трикотажного одягу занадто багато часу, тому людям простіше орієнтуватися на пряму взаємодію з виробником, унікальність та чесні умови, а не на стандартний мас-маркет. Проте знайти та безпечно купити таку річ зараз дуже важко. Популярні маркетплейси та соціальні мережі зазвичай перевантажені нав'язливими рекламними банерами, мають заплутаний інтерфейс, а замовлення в директі часто губляться через складність комунікації. Крім того, більшість сторонніх платформ є просто незручними для вибору індивідуальних параметрів, що перетворює процес покупки на виснажливий процес.

У цих реаліях розробка мого інтернет-магазину є критично важливою, оскільки мій сайт дозволяє створити нову, сучасну та просту вебплатформу, де люди можуть легко знайти в'язаний одяг, який їм справді сподобається. Створення цього сайту дозволяє повністю автоматизувати процес взаємодії покупців та виробника: замість довгих ручних листувань у месенджерах, користувач отримує зрозумілий інструмент, де можна швидко обрати потрібні характеристики й оформити покупку в кілька кліків. Мій сайт забезпечує повну незалежність від мінливих алгоритмів соціальних мереж чи сторонніх сервісів, а інтеграція надійних онлайн-платежів гарантує абсолютну безпеку фінансових взаємодій. Таким чином, розробка такого інтернет-магазину є вкрай актуальною, адже кардинально змінює підхід до купівлі одягу й перетворює цей процес із виснажливого пошуку на приємний, швидкий та прозорий досвід, побудований на довірі та простоті.

Метою випускної кваліфікаційної роботи бакалавра є розробка програмного комплексу для представлення, вибору та купівлі в'язаних

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

виробів машинної роботи із використанням ефективних архітектурних підходів.

Для розв'язання поставлених завдань та розробки платформи застосовуються сучасні методи системного аналізу, які досліджують вимоги до продукту, принципи об'єктно-орієнтованого програмування для побудови архітектури, а також методи візуального моделювання для проєктування бази даних і логіки роботи вебзастосунку.

Завданням для досягнення поставленої мети проєкту необхідно виконати комплекс завдань, що передбачає проведення детального аналізу ринку сервісів електронної комерції у сегменті в'язаних товарів з метою виявлення основних тенденцій та формування чітких вимог до функціональності майбутньої системи. На основі отриманих даних важливо здійснити обґрунтований вибір засобів розробки, довівши доцільність та ефективність використання обраного сучасного технологічного стеку для створення стабільного вебпродукту. Завершальним і ключовим етапом роботи є безпосереднє проєктування та програмна реалізація самого інтернет-магазину разом із повноцінною системою управління (адмін-панеллю), яка дозволить зручно контролювати всі процеси взаємодії з клієнтами та керувати асортиментом в'язаного одягу.

Об'єктом дослідження є процес створення сайту для продажу трикотажного одягу.

Предметом дослідження є функціональні можливості, архітектура та алгоритми роботи інтернет-магазину, які дозволяють виробнику виставляти продукцію на продаж, а покупцям зручно шукати, обирати та оформлювати замовлення онлайн.

Дослідження також охоплює питання масштабованості платформи та забезпечення високої швидкості оновлення даних про доступні товари, їхні характеристики та статуси замовлень для кінцевих користувачів.

Практичне значення полягає у створенні повністю працездатного інтернет-магазину в'язаного одягу, який автоматизує весь цикл взаємодії між

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

виробником та покупцем від вибору індивідуальних параметрів товару до фіксації замовлення. Розроблена вебплатформа надає готове практичне рішення для локального бренду, дозволяючи відмовитися від ручної обробки замовлень у месенджерах, інтегрувати надійні платіжні онлайн-сервіси для безпечних фінансових взаємодій та незалежно просувати свій продукт на ринку електронної комерції. Окрім безпосереднього комерційного використання, описані архітектурні підходи, структура бази даних та реалізовані модулі управління магазином можуть слугувати технологічною основою для розробки аналогічних систем у суміжних сферах цифрового роздрібного бізнесу.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИБІР ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ

1.1. Постановка задачі

У межах кваліфікаційної роботи необхідно дослідити предметну область електронної комерції у сфері в'язаного одягу та сформулювати задачу створення інтернет-магазину трикотажних виробів із використанням ефективних архітектурних підходів. Постановка задачі передбачає визначення цілей розробки вебплатформи для автоматизації процесів продажу й реклами, формування її основних функціональних можливостей та архітектурних обмежень, а також аналіз потреб цільової аудиторії та інструментів для унікального позиціонування бренду в інтернеті. На основі аналізу потреб бізнесу було сформульовано вимоги до можливостей майбутнього інтернет-магазину. Він має забезпечувати повний цикл взаємодії з покупцем, починаючи від інтерактивного представлення каталогу товарів із детальним описом їхніх характеристик і закінчуючи автоматизованим оформленням та безпечною оплатою замовлень через надійні платіжні системи.

Програма має забезпечувати зручний перегляд товарів за допомогою інтерактивного каталогу, де кожен трикотажний виріб містить детальний опис, фотографії та основні характеристики. Покупець може самостійно додавати речі до кошика, швидко заповнювати дані для доставки та переходити до автоматизованого оформлення покупки. Система повинна підтримувати інтеграцію з платіжними сервісами для здійснення безпечної онлайн-оплати замовлень у реальному часі.

Для власника бізнесу необхідно реалізувати захищену панель адміністрування, яка дозволяє повністю контролювати роботу інтернет-магазину. Через цей інтерфейс адміністратор може додавати нові в'язані вироби, змінювати ціни, стежити за залишками продукції на складі та переглядати профілі зареєстрованих покупців. Також система має

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечувати керування статусами замовлень та відображати аналітику щодо прибутку і найбільш популярних товарів.

Окремим важливим функціоналом сайту є індивідуальні замовлення, які дозволяють врахувати специфіку виготовлення в'язаного одягу. В адміністративній панелі власник повинен мати можливість самостійно конфігурувати форми для персональних замовлень, додаючи потрібні поля для введення даних, вибору типу пряжі чи візерунку. Покупці, своєю чергою, отримують зручний інструмент на сайті, де вони можуть заповнювати ці форми за власними параметрами, вказуючи індивідуальні мірки та особливі побажання до майбутнього виробу перед відправкою заявки в обробку.

Код і внутрішня структура сайту мають бути оптимізовані для пошукових двигунів [3], що дозволить забезпечити успішне SEO-просування інтернет-магазину в мережі [1]. Важливою вимогою є створення унікального дизайну та повністю адаптивного інтерфейсу користувача. Вебплатформа повинна однаково зручно працювати на смартфонах, планшетах і комп'ютерах, забезпечуючи при цьому швидке завантаження контенту [2].

З боку технічної архітектури система має гарантувати надійне збереження персональних даних користувачів та захист фінансових транзакцій. Інтернет-магазин повинен демонструвати високу стабільність роботи та стійкість до відмов навіть під час високих навантажень або великої кількості одночасних відвідувачів.

Таким чином, у результаті проведеного дослідження та аналізу предметної області було чітко визначено архітектурні межі та ключовий функціонал майбутнього додатка. Основними функціями інтернет-магазину є унікальне брендоване представлення трикотажної продукції, автоматизована обробка замовлень із безпечною оплатою, можливість гнучкого налаштування та заповнення форм індивідуальних замовлень, а також комплексне адміністрування з аналітикою для ефективного управління бізнес-процесами. Сформовані нефункціональні вимоги гарантують високу швидкість роботи, адаптивність під мобільні пристрої, готовність до

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

пошукового просування та надійний захист персональних даних користувачів.

1.2. Аналіз аналогів програмного продукту

Для аналізу було обрано веб-сайти, що спеціалізуються на продажі в'язаних виробів та матеріалів для їх створення. Відповідно до тематики роботи, ці ресурси мають функціонал особистих кабінетів, систему відгуків (коментарів) та розділи з порадами або блогами, де користувачі можуть ставити запитання. Під цей опис підходять такі платформи, як «Etsy», «Wool and the Gang» та «LoveCrafts».

Etsy - найбільший міжнародний маркетплейс товарів ручної роботи. Головна мета платформи Etsy полягає у створенні глобальної спеціалізованої екосистеми, яка об'єднує незалежних майстрів, дизайнерів та колекціонерів вінтажу з покупцями, що шукають унікальні, автентичні товари та альтернативу масовому маркетплейс-виробництву. Платформа прагне монетизувати творчість, надаючи локальним ремісникам прямий вихід на міжнародний ринок без значних стартових капіталовкладень. У контексті індустрії рукоділля Etsy виконує роль цифрового бізнес-інкубатора, де головний фокус зміщено з масштабів виробництва на цінність людської праці, унікальність кожної речі та персоналізований купівельний досвід.

Для досягнення цієї мети маркетплейс надає широкі функціональні можливості як для продавців, так і для аудиторії покупців. Майстри з в'язання та інших ремесел отримують інструменти для розгортання повноцінних брендів сторінок: системи управління замовленнями, аналітичні панелі для відстеження трафіку, інструменти внутрішньої пошукової оптимізації (SEO) та інтегровані рекламні кабінети. Покупцям платформа пропонує гнучкі алгоритми пошуку за специфічними тегами та категоріями, безпечну платіжну інфраструктуру з підтримкою міжнародних транзакцій, а також інтерактивні інструменти для комунікації й кастомізації замовлень безпосередньо через інтерфейс сайту.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Серед ключових переваг аналізованої платформи чітко виділяється високий рівень інтерактивності та соціального доказу, зокрема через систему деталізованих фото-відгуків, яка суттєво нівелює бар'єри недовіри до якості дистанційного в'язання. Опції глибокої персоналізації через систему особистих повідомлень дозволяють покупцеві та майстру детально узгодити параметри виробу, як-от точні заміри чи склад і гіпоалергенність пряжі, що створює ефект індивідуального ательє. Крім того, сучасний, інтуїтивно зрозумілий та ергономічний дизайн інтерфейсу забезпечує легку навігацію, утримує увагу користувача та стимулює повторні конверсії завдяки зручним алгоритмам рекомендацій.

Поряд із перевагами, Etsy має суттєві недоліки, які ускладнюють масштабування бізнесу для незалежних майстрів. По-перше, багаторівнева система комісій помітно знижує маржинальність і з'їдає прибуток. По-друге, платформа має жорсткі обмеження щодо кастомізації дизайну магазину, через що продавцям важко відбудувати унікальну візуальну ідентичність бренду в межах стандартного шаблону. Нарешті, останніми роками маркетплейс потерпає від критичного перенасичення ринку та напливу прихованого масового перепродажу товарів фабричного виробництва, що створює демпінг цін та ускладнює органічне просування для авторів справжнього hand-made контенту. Інтерфейс маркетплейсу представлений на рис. 1.1 та рис. 1.2.

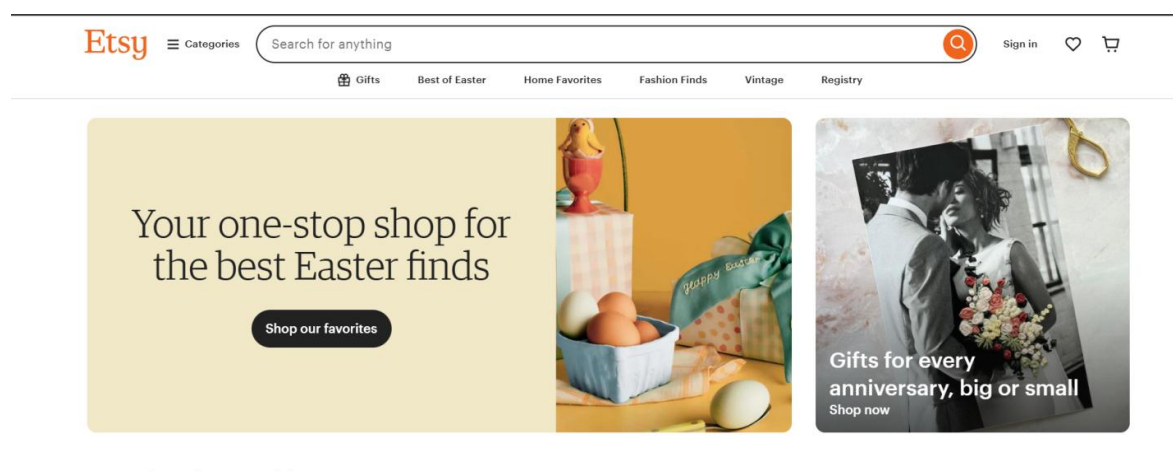


Рис. 1.1. Головна сторінка веб-сайту

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

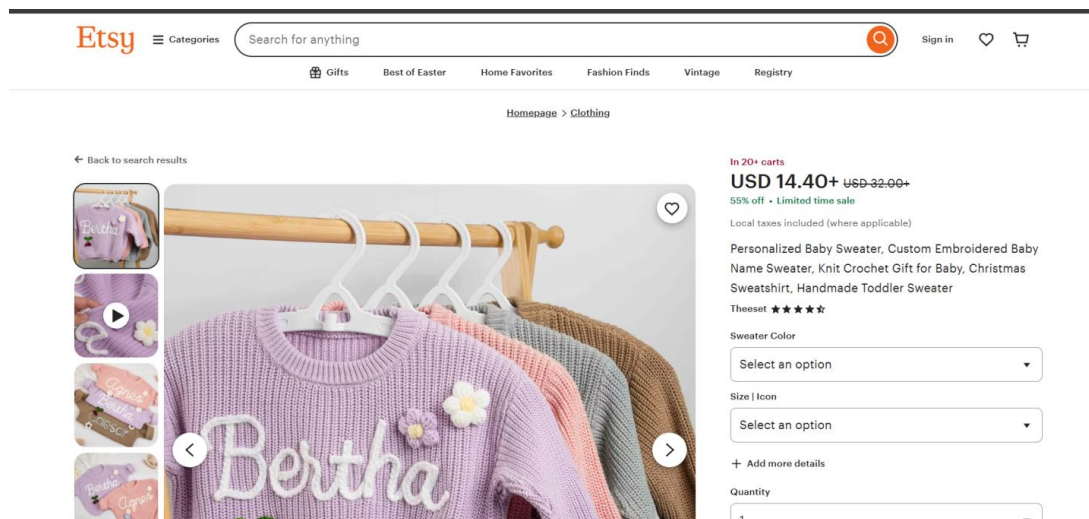


Рис. 1.2. Продукт

Wool and the Gang - спеціалізована e-commerce платформа. Головною метою міжнародного інтернет-магазину та бренду Wool and the Gang є популяризація культури ручного в'язання серед сучасної молодіжної аудиторії через концепцію стійкої моди та комерціалізацію готової продукції й унікальних трикотажних наборів. Платформа прагне змінити сприйняття рукоділля, перетворюючи його на трендовий, екологічний та соціально інтегрований процес, що дозволяє користувачам створювати дизайнерський одяг власноруч і мінімізувати текстильні відходи.

Для успішної реалізації цієї концепції веб-сайт надає користувачам інтегровані комерційні та освітні можливості. Клієнти мають доступ до гнучкого інструментарію підбору матеріалів, де разом із деталізованими схемами в'язання можна кастомізувати набори за типом, кольором та кількістю пряжі, а також за рівнем складності роботи (від початківця до експерта). Особливе місце посідає потужний соціально-освітній блок: платформа пропонує розгалужену базу безкоштовних відеоуроків, інтерактивні інструкції та цифрові комерційні простори для обміну досвідом всередині глобальної спільноти майстрів під фірмовими хештегами.

Ключовою перевагою Wool and the Gang є ультрасучасний, мінімалістичний та візуально привабливий дизайн інтерфейсу, який чітко диференціює бренд на тлі традиційних та застарілих ресурсів для рукоділля. Наявність якісного, структурованого навчального контенту та інтерактивних

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

розділів для запитань суттєво знижує поріг входження для новачків, формуючи високу лояльність до бренду. Проте, суттєвим недоліком платформи є її вузький асортимент, обмежений виключно власною монобрендовою продукцією, що позбавляє покупця можливості порівнювати ціни чи обирати товари інших виробників, програючи в цьому аспекті масштабним міжнародним маркетплейсам.

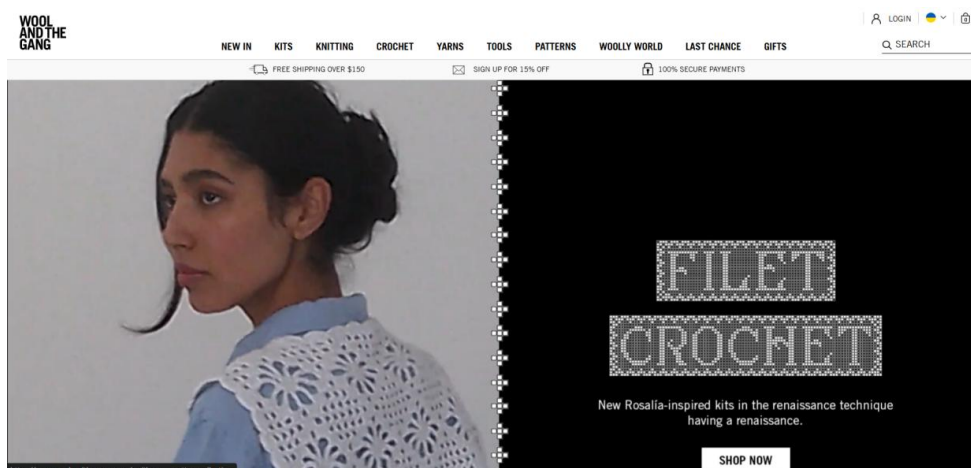


Рис. 1.3. Головна сторінка веб-сайту.

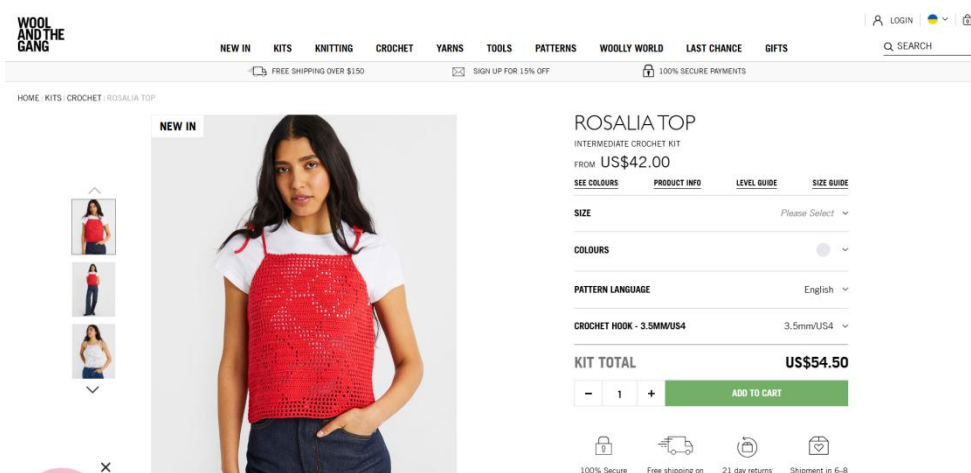


Рис. 1.4. Продукт.

LoveCrafts - інтегрована e-commerce платформа та соціальна мережа. Головною метою міжнародного ресурсу LoveCrafts є створення глобальної цифрової екосистеми, яка об'єднує функції масштабного інтернет-магазину товарів для рукоділля та спеціалізованої соціальної мережі для майстрів. Платформа прагне забезпечити безперервний клієнтський досвід: від пошуку натхнення та схем у спільноті до миттєвої купівлі необхідних матеріалів

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІПЗ	Арк.
		Морозов А.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

(пряжі, інструментів), формуючи у такий спосіб стійку та залучену навколо бренду комерційну аудиторію.

Для реалізації цієї мети веб-сайт надає користувачам розширені можливості для взаємодії з контентом і продукцією. Клієнтам доступний персоналізований профіль майстра, де вони можуть публікувати власні проєкти, вести блоги, коментувати роботи інших учасників та безкоштовно або на комерційній основі поширювати схеми для в'язання чи вишивання. Комерційний блок сайту пропонує інтегровану систему кошика, яка дозволяє в один клік купувати всі матеріали, зазначені автором у специфікації до конкретного візуального проєкту чи схеми.

Суттєвою перевагою LoveCrafts є розвинений функціонал соціальної спільноти, який забезпечує високий рівень органічного залучення користувачів і стимулює обмін досвідом. Крім того, платформа має надзвичайно детальну систему фільтрації у каталозі, яка дозволяє оптимізувати пошук за типом виробу, рівнем складності схеми, брендом, складом та навіть товщиною вовни. Водночас вагомим недоліком ресурсу є помітне інформаційне перенасичення інтерфейсу: через надмірну концентрацію різномірних елементів (комерційного магазину, блогів, форумів та схем) на одній сторінці, навігація сайту стає складною, що може дезорієнтувати покупця та знизити конверсію швидких покупок.

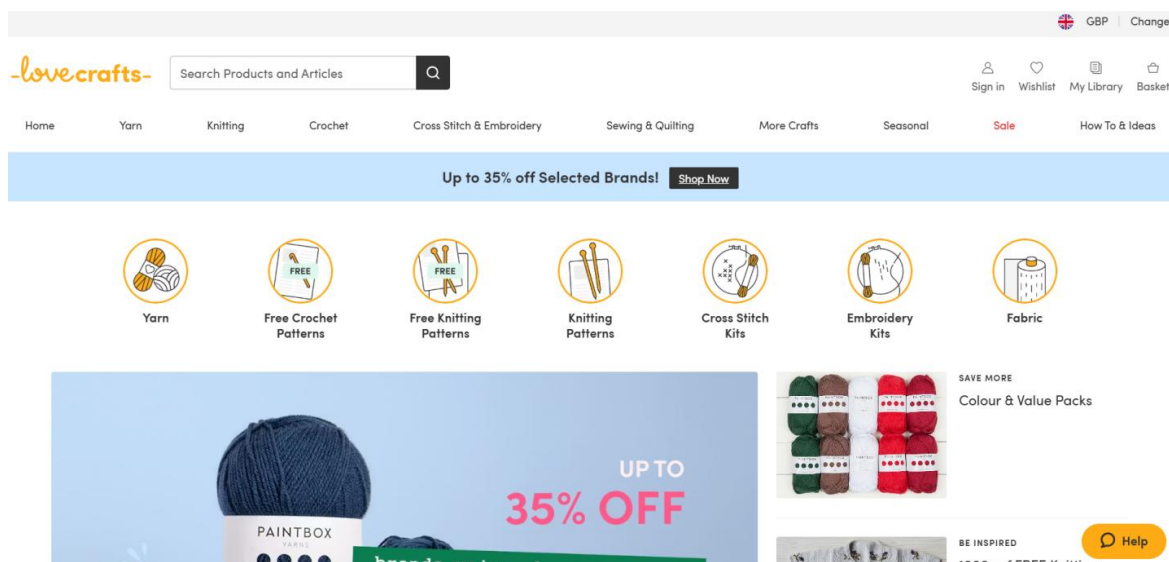


Рис. 1.5. Головна сторінка

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

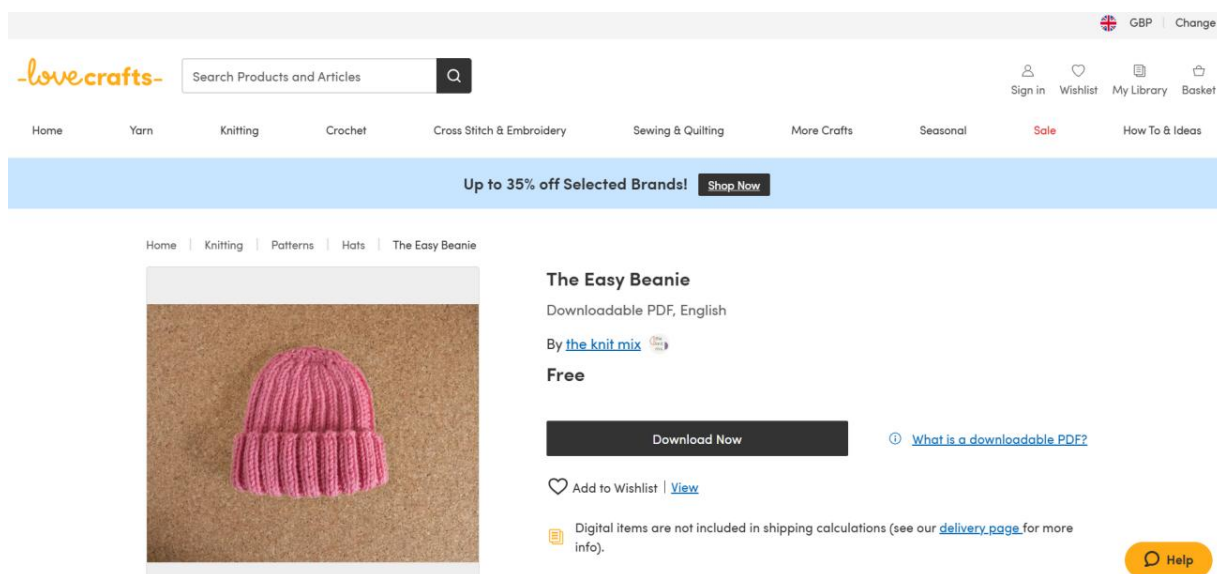


Рис. 1.6. Сторінка продукт

Порівняння рішень електронної комерції наведено у табл. 1.1.

Таблиця порівняння платформ (таблиця 1.1):

Характеристика	Etsy	Wool and the Gang	LoveCrafts	Розроблюваний інтернет-магазин
Просування власного бренду	+	+	+	+
Відсутність комісій маркетплейса	-	-	-	+
Система онлайн-оплати	+	+	+	+
Швидкий пошук та фільтрація	+	+	+	+

продовж. табл. 1.1

Висока швидкість завантаження	-	+	-	+
Пряма взаємодія з майстром	+	-	-	+
SEO-оптимізація товарів для Google	-	+	-	+
Масштабованість під потреби бізнесу	-	-	-	+

Порівняльний аналіз програмних аналогів свідчить, що існуючі рішення часто мають обмеження у вигляді високих комісій, перевантажених інтерфейсів або недостатньої швидкості роботи, що перешкоджає ефективному розвитку власного бренду [4]. Розроблювана система забезпечує кращу SEO-оптимізацію, високу швидкість завантаження сторінок та пряму взаємодію з покупцями без посередників. Інтернет-магазин поєднує в собі переваги професійного маркетплейса з повною свободою у просуванні авторського стилю, що робить її найбільш вигідним інструментом для автоматизації продажів та зростання бренду в мережі [5].

1.3. Обґрунтування вибору інструментальних засобів

Для реалізації розроблюваного інтернет-магазину було сформовано сучасний, збалансований та високоефективний технологічний стек. Комплекс обраних інструментів повністю відповідає критеріям швидкодії, безпеки, гнучкості розгортання та здатності витримувати високі навантаження під час пікових продажів.

Основою серверної логіки інтернет-магазину обрано мову програмування PHP у поєднанні з сучасними архітектурними підходами її

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

екосистеми. На сьогоднішній день РНР демонструє високу швидкість обробки HTTP-запитів, низькі накладні витрати на ініціалізацію та оптимізоване керування пам'яттю, що є критично важливим для систем із динамічним каталогом товарів [6].

Платформа проєктована як модульна структура з чітким розділенням зон відповідальності. Використання сучасного пакетного менеджера Composer забезпечує суворе керування залежностями, швидку інтеграцію сертифікованих бібліотек та автоматизований аудит безпеки сторонніх пакунків [7]. Серверна частина містить вбудовані інструменти для автентифікації користувачів, гнучкого розмежування прав доступу та надійного захисту від поширених вебвразливостей.

Для розробки користувацького інтерфейсу вебплатформи обрано бібліотеку React [8]. Цей вибір зумовлений необхідністю створення динамічного, чуйного та повністю адаптивного UI, що однаково стабільно функціонує на десктопних та мобільних пристроях.

Завдяки концепції Virtual DOM [9], React мінімізує витрати на оновлення сторінок, забезпечуючи миттєву реакцію інтерфейсу на дії користувача. Компонентний підхід дозволяє створити унікальний, брендований дизайн із перевикористовуваними елементами, що спрощує підтримку та оптимізує швидкість завантаження сторінок. Крім того, архітектура frontend частини адаптована для інтеграції інструментів серверного рендерингу [10], що повністю задовольняє нефункціональні вимоги щодо SEO-оптимізації інтернет-магазину для пошукових систем.

Централізованим сховищем структурованої інформації обрано об'єктно-реляційну СУБД PostgreSQL [11]. Вона є беззаперечним лідером серед рішень із відкритим вихідним кодом за показниками надійності, масштабованості та суворого дотримання принципів ACID, що є критичним для безпечного ведення фінансових транзакцій, обліку замовлень та збереження персональних даних покупців [12].

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

Висока швидкість обробки запитів досягається завдяки ефективному механізму індексування. Гнучкість PostgreSQL дозволяє успішно поєднувати сувору реляційну структуру із напівструктурованими даними за допомогою типу JSONB [13]. Це оптимізує збереження специфічних, динамічних характеристик в'язаного одягу без ускладнення схеми бази даних та втрати продуктивності.

Для збереження та дистрибуції статичного медіаконтенту застосовано об'єктне сховище, сумісне зі стандартом S3 [15].

Винесення важких медіафайлів за межі сервера додатків та реляційної бази даних запобігає надмірному розростанню дискового простору СУБД та суттєво прискорює створення резервних копій. Об'єктне сховище забезпечує високу доступність, безпечний доступ до ресурсів за тимчасово підписаними посиланнями та швидку віддачу контенту користувачам, що безпосередньо впливає на загальну плавність роботи інтерфейсу інтернет-магазину.

З метою розширення можливостей обраної бази даних та спрощення процесу розгортання до інфраструктури інтегровано платформу Supabase. Використання цього інструменту дозволяє зручно розробляти вебдодаток і гнучко керувати міграціями без необхідності локального розгортання та адміністрування бази даних і об'єктного сховища на власному пристрої розробника. Натомість усі необхідні компоненти функціонують у хмарній екосистемі, забезпечуючи швидкий доступ до даних, автоматичне масштабування та спрощену синхронізацію між середовищами розробки та тестування.

З метою зниження часу відгуку системи та мінімізації навантаження на основну базу даних, до інфраструктури інтегровано високошвидкісне сховище в оперативній пам'яті Redis [14].

Redis функціонує як ефективний шар кешування для даних, що часто запитуються, але рідко змінюються або отримуються з сторонніх ресурсів. Також інструмент використовується для швидкого та надійного збереження

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

сесій користувачів, що гарантує безперебійну роботу кошика покупця навіть у разі високої інтенсивності трафіку або короткочасного оновлення сервера.

Для забезпечення ізоляції компонентів системи, стандартизації середовища розробки та спрощення процесу розгортання інтернет-магазину використано технологію контейнеризації Docker. Завдяки пакуванню сервісів додатку, залежностей та конфігурацій у легковагові ізольовані контейнери, Docker повністю нівелює проблему розбіжностей між локальними машинами розробників і продуктовим сервером. Це гарантує ідентичність поведінки коду на всіх етапах життєвого циклу проекту, дозволяє швидко масштабувати окремі вузли архітектури під час пікових навантажень та суттєво прискорює процеси безперервної інтеграції та доставки.

Проектування архітектури платформи базується на сучасному технологічному стеку, спроможному забезпечити високу швидкість обробки даних, гнучке масштабування та стабільну роботу системи під навантаженням.

Мінімальні вимоги для розгортання системи:

Для забезпечення стабільної роботи серверної складової вебплатформи, що координує взаємодію інтерпретатора PHP, клієнтських запитів React-додатка, реляційної бази даних PostgreSQL, кеш-сервера Redis та інтерфейсів взаємодії з об'єктним S3-сховищем, розгортання інфраструктури потребує обчислювальної платформи на базі чотириядерного процесора з архітектурою x86-64 чи ARM64 та тактовою частотою від 2.4 ГГц. З метою уникнення затримок під час обробки складних реляційних запитів і підтримки високої швидкодії шару кешування в оперативній пам'яті, сервер має бути оснащений щонайменше 8 Гігабайтами оперативної пам'яті. Для збереження індексів бази даних, логів системи та тимчасових файлів необхідно виділити від 40 Гігабайт вільного простору на швидкісному твердотільному накопичувачі, а також організувати мережевий канал із пропускнуою здатністю від 100 Мегабіт за секунду. Базовим програмним середовищем для розгортання компонентів виступає операційна система

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

Linux, де обов'язковою технічною умовою є наявність встановленого інструментарію контейнеризації Docker та утиліти Docker Compose для ізолюваного керування та оркестрації всіх сервісів платформи [16].

Висновки до першого розділу

У першому розділі поставлено задачу на розробку вебплатформи для продажу трикотажних виробів, визначено ключові цілі та обмеження проєкту, а також сформульовано основні функціональні й нефункціональні вимоги до майбутньої системи та її адміністративної частини.

Виконано порівняльний аналіз популярних міжнародних аналогів і маркетплейсів у сфері рукоділля. Виявлено їхні головні переваги й недоліки, що дозволило обґрунтувати необхідність створення власного незалежного інтернет-магазину для ефективного просування бренду без додаткових комісій.

Окреслено загальні архітектурні вимоги до системи та визначено мінімальні параметри серверної інфраструктури, необхідні для стабільної та ізолюваної роботи всіх модулів платформи під навантаженням.

На основі аналізу потреб бізнесу та вимог до швидкодії й пошукової оптимізації обґрунтовано вибір сучасного технологічного стеку. Для розробки серверної логіки та управління базою даних обрано надійні серверні інструменти, для створення адаптивного й динамічного інтерфейсу використано сучасні клієнтські технології, а для підвищення швидкості роботи системи та збереження медіафайлів інтегровано ефективні рішення з кешування та хмарного збереження даних.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ ІНТЕРНЕТ-МАГАЗИНУ В'ЯЗАНОГО ОДЯГУ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

На етапі проєктування інтернет-магазину в'язаного одягу було визначено ключові прецеденти, які детально описують принципи взаємодії користувачів із вебплатформою. Логіка розподілу варіантів використання формувалася на основі специфіки електронної комерції та зафіксованих функціональних вимог до системи.

Архітектура додатка чітко розмежовує права доступу для різних типів користувачів. Клієнтська сторона орієнтована на покупця, який шукає унікальні речі, керує особистим профілем, взаємодіє з кошиком та здійснює фінансові розрахунки. Адміністративна частина розроблена для власника бізнесу, який контролює наповнення каталогу, відстежує статуси покупок, модерує облікові записи та аналізує комерційну статистику [17]. Ця структура взаємозв'язків наочно відображена на загальній діаграмі варіантів використання.

Процес взаємодії з платформою починається з вивчення асортименту. Покупець підбирає одяг за допомогою пошуку, фільтрації та сортування. Кожна позиція містить детальний опис та варіації виконання, де можна вказати потрібний колір і розмірну сітку. Користувач додає та переглядає товари додані до кошика, для оформлення замовлення користувач вказує відділення Нової Пошти для доставки та проводить оплату. Сервіс автоматично перевіряє актуальні залишки продукції на складі та не дозволяє створення замовлень, що не є в наявності. Також для замовлення індивідуальних товарів користувач може заповнити форму з потрібними параметрами товару, який може буде підтверджений адміністратором.

Варіанти використання системи покупцем відображено на діаграмі прецедентів (рис. 2.1).

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				23
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.1. Варіанти використання системи клієнтом

Адміністратор платформи виконує ключову роль у забезпеченні стабільного функціонування системи та обробці специфічних запитів клієнтів. Його діяльність починається з моніторингу надходження нових заявок на індивідуальні товари. Коли покупець надсилає заповнену форму індивідуального товару, адміністратор ретельно перевіряє вказані параметри, оцінює технічну можливість створення такого одягу та за результатами перевірки підтверджує або відхиляє це замовлення.

Паралельно з цим адміністратор здійснює повний контроль за матеріально-технічною базою платформи. Він безпосередньо керує актуальними залишками на складі та вносить зміни в базу даних, що дозволяє автоматичній системі валідації вчасно блокувати оформлення замовлень на позиції, яких немає в наявності. Також адміністратор повністю відповідає за візуальну та інформаційну складову платформи, регулярно модеруючи каталог одягу. Це включає систематичне оновлення детальних описів, додавання актуальних розмірних сіток та редагування варіацій для кожної позиції, нових кольорів та тканин.

Варіанти використання системи адміністратором відображено на діаграмі прецедентів (рис. 2.2).

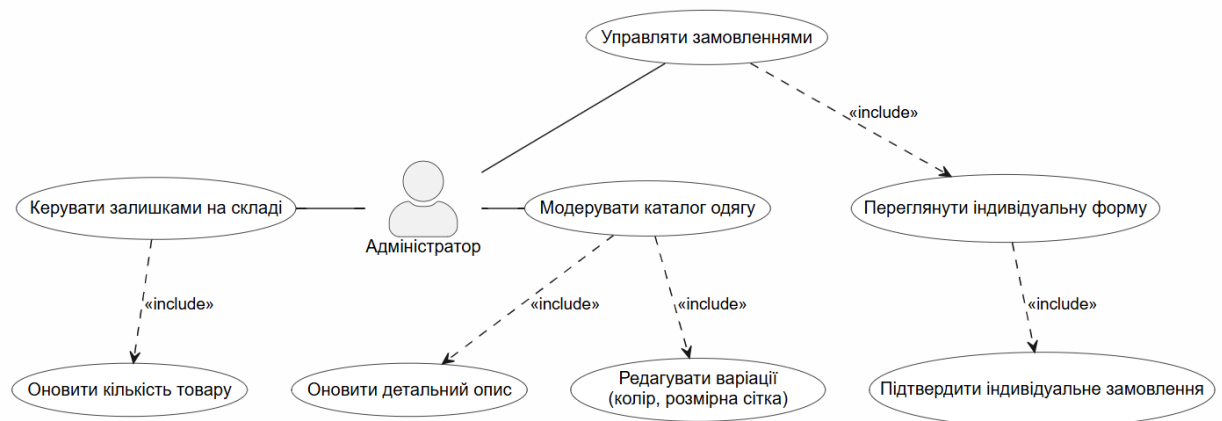


Рис. 2.2. Варіанти використання системи адміністратором.

Сформований перелік прецедентів визначає ключові сутності, функціонал яких забезпечуватиме система. Загальну архітектуру та взаємозв'язки елементів предметної області візуалізовано за допомогою діаграми класів (рис. 2.3). Вона визначає вимоги до структури доменної моделі, залишаючись незалежною від технологічного стека чи конкретної програмної реалізації.

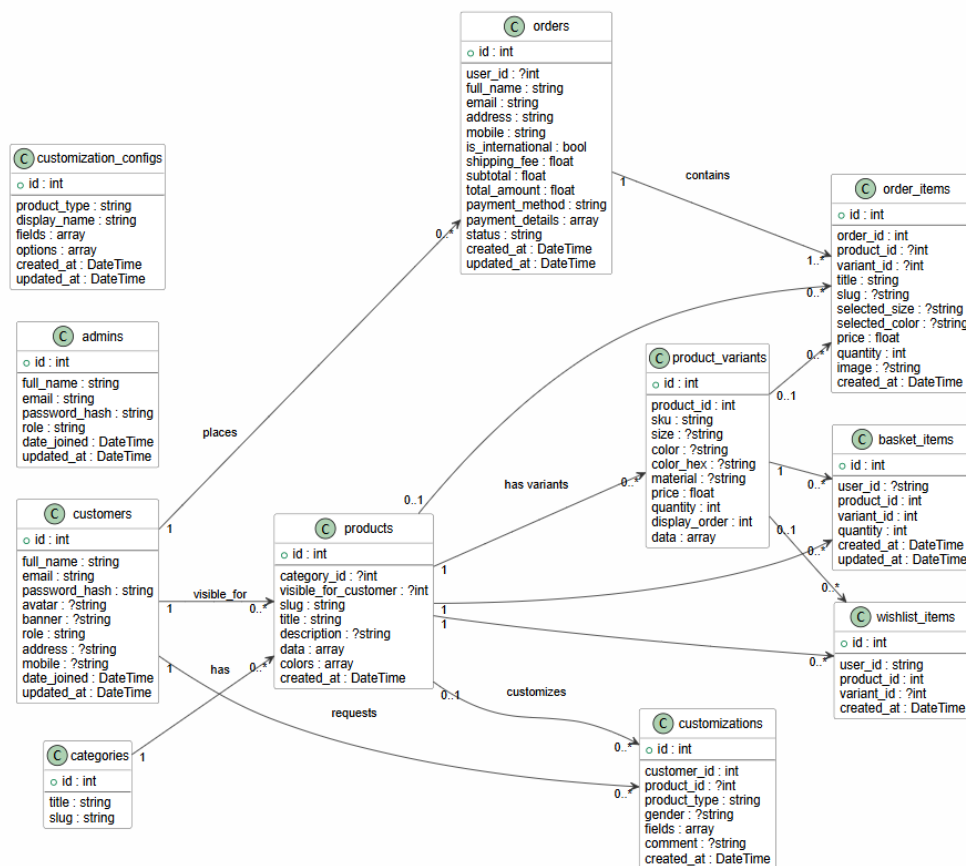


Рис. 2.3. Діаграма класів

Базою для збереження персональних профілів клієнтів виступає сутність покупці. Вона містить автентифікаційні дані, зокрема пароль та електронну пошту, а також контактну інформацію, мобільний телефон, аватар, банер і дефолтну адресу, що спрощує подальшу взаємодію з платформою та оформлення замовлень.

Повністю відокремленим типом профілю для співробітників платформи функціонує сутність адміністратори. З її допомогою ізолюються системні ролі та забезпечується безпечний доступ до панелі модерації каталогу, керування замовленнями й індивідуальними формами.

За тематичне структурування асортименту одягу на сайті відповідає сутність категорії. Вона містить назву та унікальний людинозрозумілий шлях, що дозволяє групувати товари й будувати зручну навігацію для покупця.

Головною карткою для моделі одягу в каталозі виступає сутність товари. Тут зберігається загальна інформація, така як назва, детальний опис, посилання на категорію, прапорець видимості для клієнта, а також масиви доступних кольорів і загальних характеристик.

До конкретних фізичних одиниць на складі базову картку деталізує сутність варіанти товарів. Саме в ній фіксуються унікальні комбінації для кожної позиції: артикул, конкретний розмір, колір та його гексакод, матеріал, ціна й актуальна кількість залишків.

Збереження та перегляд продукції, яку покупець вибрав для потенційного замовлення, забезпечує сутність елементи кошика. Вона динамічно пов'язана з конкретною моделькою та її точним варіантом виконання за розміром чи кольором, фіксуючи потрібну кількість.

Відкладати товари на майбутнє покупцям дозволяє сутність елементи списку бажаного. Вона фіксує інтерес користувача до певної моделі або конкретного варіанту одягу, зберігаючи дату додавання без резервування залишків на складі.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

Головним документом фінансової та логістичної звітності є сутність замовлення. Вона акумулює дані про покупця, точну адресу доставки, вартість відправлення, загальну суму до сплати, обраний спосіб оплати, платіжні реквізити та поточний статус обробки транзакції.

Роль історичного зрізу транзакції виконує сутність елементи замовлення. Вона деталізує кожну покупку, жорстко фіксуючи назву товару, його обраний колір, розмір, ціну та кількість на момент оплати, що захищає дані від майбутніх змін у загальному каталозі.

Функціонал створення одягу за персональними параметрами клієнта реалізує сутність індивідуальні замовлення. Вона реєструє запит покупця, вибраний тип виробу, стать, додаткові коментарі, а також динамічний масив індивідуальних мірок і специфічних побажань.

Внутрішнім інструментом налаштування платформи є сутність конфігурації індивідуальних замовлень. З її допомогою визначаються правила, за якими будується форма індивідуального товару, задаються доступні типи продукції, назви полів для заповнення та готові опції, які пропонуються покупцю.

2.2. Розробка бази даних системи

Для забезпечення збереження та обробки даних у системі замовлення одягу, перегляду каталогу та створення індивідуальних замовлень було спроектовано реляційну базу даних, структура якої відображає основні бізнес-процеси платформи. База даних реалізує зберігання інформації про покупців та адміністраторів, каталог товарів з їхніми конкретними розмірними й колірними варіантами, елементи кошика та списків бажаного, оформлені замовлення з деталізованим переліком придбаних позицій, а також індивідуальні запити користувачів разом із гнучкими конфігураціями полів для їхнього створення.

Проектування схеми бізнес-модуля виконувалося з урахуванням принципів нормалізації, що дозволяє зменшити надлишковість даних та забезпечити цілісність інформації. Кожна предметна сутність подається

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

окремою таблицею, а зв'язки між ними реалізовано через зовнішні ключі для підтримання референційної цілісності на рівні системи керування базою даних PostgreSQL. Загальну структуру спроектованої схеми бази даних бізнес-модулю системи подано на рис. 2.4.

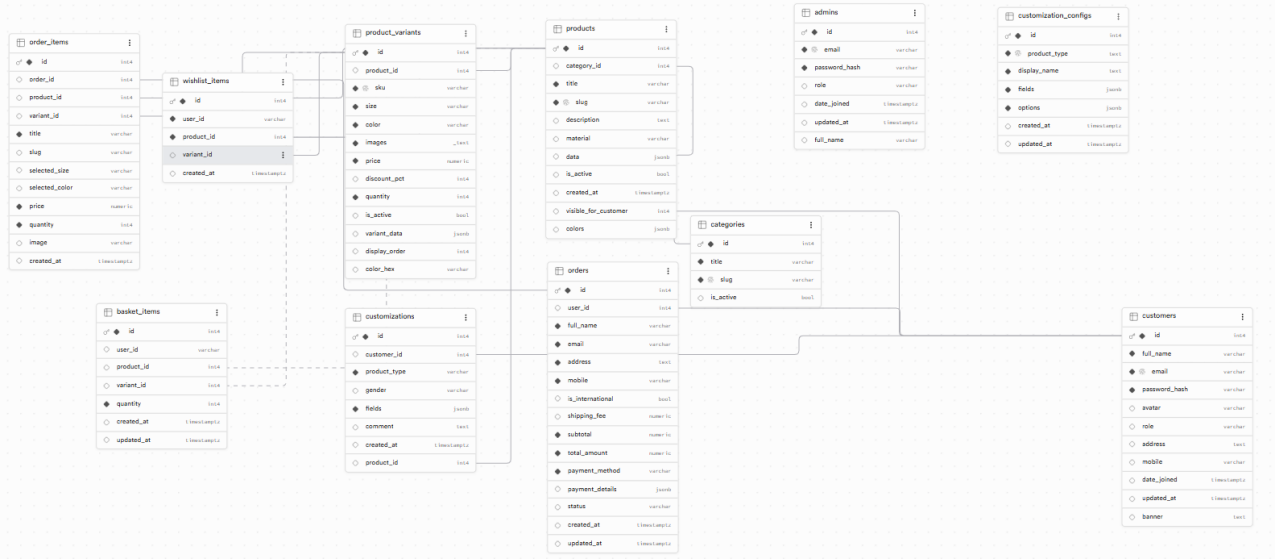


Рис. 2.4. Схема бази даних.

Сутність покупці є ключовим сховищем облікових записів та персональної інформації клієнтів. Вона забезпечує роботу модулів автентифікації, безпеки та персоналізації інтерфейсу платформи. Детальний опис структури полів та типів даних для зберігання інформації про покупців представлено в табл. 2.1.

Таблица 2.1

Опис таблиці customers

Назва поля	Тип даних	Р К	Ф К	Опис
id	SERIAL	+		Унікальний ідентифікатор покупця.
full_name	VARCHAR(255)			Прізвище та ім'я клієнта.
email	VARCHAR(255)			Електронна пошта для входу (унікальна).

продовж. табл. 2.1

password_hash	VARCHAR(255)			Хеш-значення пароля користувача.
avatar	VARCHAR(255)			Шлях до зображення профілю.
banner	VARCHAR(255)			Шлях до фонового баннера кабінету.
role	VARCHAR(50)			Системна роль користувача ('USER').
address	TEXT			Основна адреса доставки одягу.
mobile	VARCHAR(255)			Номер мобільного телефону клієнта.
date_joined	TIMESTAMP WITH TIME ZONE			Дата та час реєстрації профілю.
updated_at	TIMESTAMP WITH TIME ZONE			Час останньої модифікації даних.

Сутність адміністратори повністю відокремлює облікові записи персоналу платформи від клієнтської бази. Детальний опис структури полів та типів даних для зберігання інформації про адміністраторів представлено в табл. 2.2.

Таблиця 2.2

Опис структури полів таблиці admins

Назва поля	Тип даних	PK	FK	Опис
id	SERIAL	+		Ідентифікатор адміністратора.

продовж. табл. 2.3

full_name	VARCHAR(255)			Повне ім'я співробітника.
email	VARCHAR(255)			Робоча електронна пошта (унікальна).
password_hash	VARCHAR(255)			Хеш-значення пароля адміна.
role	VARCHAR(50)			Рівень доступу ('ADMIN').
date_joined	TIMESTAMP WITH TIME ZONE			Час створення облікового запису.
updated_at	TIMESTAMP WITH TIME ZONE			Час останнього оновлення прав чи даних.

Сутність категорії відповідає за ієрархічне проектування дерева товарів платформи. Детальний опис структури полів та типів даних для зберігання інформації про категорії каталогу представлено в табл. 2.3.

Таблиця 2.3

Опис структури полів таблиці categories

Назва поля	Тип даних	PK	FK	Опис
id	SERIAL	+		Ідентифікатор категорії.

продовж. табл. 2.3

title	VARCHAR(100)			Назва категорії (наприклад, «Светри»).
slug	VARCHAR(100)			Символьний аліас для формування URL.

Сутність товари описує фундаментальні характеристики моделей одягу в системі. Детальний опис структури полів та типів даних для зберігання інформації про моделі товарів представлено в табл. 2.4.

Таблиця 2.4

Опис структури полів таблиці products

Назва поля	Тип даних	PK	FK	Опис
id	SERIAL	+		Ідентифікатор моделі одягу.
category_id	INTEGER		+	Зв'язок із таблицею categories (SET NULL).

продовж. табл. 2.4

visible_for_customer	INTEGER		+	Зв'язок із customers для приватного показу.
slug	VARCHAR(255)			Символьний аліас картки товару для URL.
title	VARCHAR(255)			Комерційна назва виробу.
description	TEXT			Маркетинговий та технічний опис моделі.
data	JSONB			Динамічні характеристики у форматі JSON.
colors	JSONB			Масив доступних за замовчуванням кольорів.
created_at	TIMESTAMP WITH TIME ZONE			Дата та час додавання позиції до системи.

Сутність варіанти товарів розбиває загальну модель одягу на унікальні фізичні комбінації параметрів на складі. Детальний опис структури полів та типів даних для зберігання інформації про варіації одягу представлено в табл. 2.5.

Таблиця 2.5

Опис таблиці product_variants

Назва поля	Тип даних	PK	FK	Опис
id	SERIAL	+		Ідентифікатор конкретної варіації.
product_id	INTEGER		+	Зв'язок із карткою моделі products (CASCADE).
sku	VARCHAR(100)			Унікальний товарний артикул складського обліку.
size	VARCHAR(50)			Розмір одягу (S, M, L тощо).
color	VARCHAR(50)			Текстова назва кольору варіанта.
color_hex	VARCHAR(20)			Шістнадцятковий код кольору для фронтенду.

продовж. табл. 2.5

material	VARCHAR(100)			Склад тканини цієї модифікації.
price	DECIMAL(10, 2)			Роздрібна ціна товарної одиниці.
quantity	INTEGER			Доступна кількість штук на складі.
display_order	INTEGER			Порядок сортування варіантів у списку.
data	JSONB			Додаткові метадані конкретної партії.

Сутність елементи кошика тимчасово відстежує товари, обрані користувачем для подальшої купівлі. Детальний опис структури полів та типів даних для зберігання інформації про елементи кошика покупця представлено в табл. 2.6.

Таблиця 2.6

Опис структури полів таблиці basket_items

Назва поля	Тип даних	PK	FK	Опис
id	SERIAL	+		Ідентифікатор позиції в кошику.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

продовж. табл. 2.6

user_id	VARCHAR(255)			Ідентифікатор сесії або ID покупця.
product_id	INTEGER		+	Зв'язок із батьківською моделлю products.
variant_id	INTEGER		+	Зв'язок із точним складом product_variants.
quantity	INTEGER			Кількість обраних одиниць товару.
created_at	TIMESTAMP WITH TIME ZONE			Час додавання елемента до кошика.
updated_at	TIMESTAMP WITH TIME ZONE			Час останньої зміни запису.

Сутність елементи списку бажаного реалізує функціонал збереження моделей одягу «на майбутнє». Детальний опис структури полів та типів даних для зберігання інформації про список вподобань представлено в табл. 2.7.

Таблиця 2.7

Опис структури полів таблиці wishlist_items

Назва поля	Тип даних	PK	FK	Опис
id	SERIAL	+		Ідентифікатор елемента у списку бажаного.

продовж. табл. 2.7

user_id	VARCHAR(255)			Токен або унікальний ідентифікатор покупця.
product_id	INTEGER		+	Посилання на модель виробу з products.
variant_id	INTEGER		+	Опціональне посилання на точну варіацію.
created_at	TIMESTAMP WITH TIME ZONE			Дата фіксації товару у списку.

Сутність замовлення містить підсумкову логістичну та платіжну інформацію про купівлю. Детальний опис структури полів та типів даних для зберігання інформації про оформлені замовлення представлено в табл. 2.8.

Таблиця 2.8

Опис таблиці orders

Назва поля	Тип даних	PK	FK	Опис
id	SERIAL	+		Головний номер замовлення платформи.

продовж. табл. 2.8

user_id	INTEGER		+	Зв'язок із профілем клієнта customers (SET NULL).
full_name	VARCHAR(255)			Прізвище та ім'я отримувача посилки.
email	VARCHAR(255)			Електронна пошта для надсилення сповіщень.
address	TEXT			Дані доставки (відділення або поштомот).
mobile	VARCHAR(255)			Контактний телефон одержувача.
is_international	BOOLEAN			Ознака міжнародної доставки вантажу.
shipping_fee	DECIMAL(10, 2)			Нарахована вартість доставки.

subtotal	DECIMAL(10, 2)			Вартість усіх товарів без доставки.
total_amount	DECIMAL(10, 2)			Підсумкова сума до сплати (разом із доставкою).
payment_method	VARCHAR(50)			Обраний метод розрахунку (CARD, CASH).
payment_details	JSONB			Логи та відповіді від платіжного шлюзу.
status	VARCHAR(50)			Поточний статус обробки ('PENDING', 'PAID').
created_at	TIMESTAMP WITH TIME ZONE			Дата та точний час створення транзакції.
updated_at	TIMESTAMP WITH TIME ZONE			Час останньої зміни статусу менеджером.

Замовлення містять перелік продуктів що були придбані на сайті. Продукти додані до замовлення відображені сутністю елементу замовлення. Структуру відповідної сутності супроводжує детальне виокремлення атрибутів, що наведені в табл. 2.9.

Таблиця 2.9

Опис таблиці order_items

Назва поля	Тип даних	PK	FK	Опис
id	SERIAL	+		Ідентифікатор товарного рядка замовлення.
order_id	INTEGER		+	Зв'язок із батьківським замовленням orders.
product_id	INTEGER		+	М'яка прив'язка до моделі одягу products.
variant_id	INTEGER		+	М'яка прив'язка до product_variants.
title	VARCHAR(255)			Копія назви товару на момент оплати.

продовж. табл. 2.9

slug	VARCHAR(255)			Шлях картки товару на момент покупки.
selected_size	VARCHAR(50)			Зафіксований придбаний розмір одягу.
selected_color	VARCHAR(50)			Зафіксована назва придбаного кольору.
price	DECIMAL(10, 2)			Фактична ціна за одиницю на момент транзакції.
quantity	INTEGER			Кількість одиниць викупленого одягу.
image	VARCHAR(255)			Посилання на фото товару для чека.
created_at	TIMESTAMP WITH TIME ZONE			Дата формування запису про товар.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

Сутність індивідуальні замовлення обробляє заявки клієнтів на пошиття одягу за унікальними мірками. Детальний опис структури полів та типів даних для зберігання заявок на індивідуальне пошиття представлено в табл. 2.10.

Таблиця 2.10

Опис таблиці customizations

Назва поля	Тип даних	РК	FK	Опис
id	SERIAL	+		Номер індивідуального проекту пошиття.
customer_id	INTEGER		+	Зв'язок із профілем автора з customers.
product_id	INTEGER		+	Посилання на прототип виробу з products.
product_type	VARCHAR(50)			Категорійний ключ виробу (наприклад, 'sweater').

продовж. табл. 2.10

gender	VARCHAR(10)			Цільова стаття для розробки легал.
fields	JSONB			Набір персональних мірок у форматі JSON.
comment	TEXT			Додаткові побажання чи нотатки клієнта.
created_at	TIMESTAMP WITH TIME ZONE			Час надсилання форми адміністратору.

Сутність конфігурації індивідуальних замовлень виступає динамічним конструктором форм для адмін-панелі платформи. Детальний опис структури налаштувань індивідуальних форм представлено в табл. 2.11.

Таблиця 2.11

Опис таблиці customization_configs

Назва поля	Тип даних	PK	FK	Опис
id	SERIAL	+		Внутрішній ключ шаблону конфігуратора.

продовж. табл. 2.11

product_type	TEXT			Унікальний системний ідентифікатор типу одягу.
display_name	TEXT			Назва виробу для клієнта (наприклад, «Штани»).
fields	JSONB			Структура та правила валідації полів мірок.
options	JSONB			Доступні пресети матеріалів та колірна палітра.
created_at	TIMESTAMP WITH TIME ZONE			Час створення шаблону форми в базі.
updated_at	TIMESTAMP WITH TIME ZONE			Час останнього редагування структури форми.

Проектована структура бази даних повністю задовольняє вимоги ключових модулів системи. Вона забезпечує стабільну роботу поточного функціоналу платформи та закладає надійний фундамент для її подальшого масштабування.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

2.3. Проектування та реалізація алгоритмів роботи системи

Процес купівлі готового одягу починається з того, що покупець відправляє запит на пошук речей за допомогою фільтрів чи ключових слів. Модуль каталогу звертається до бази даних, витягує інформацію про моделі з урахуванням наявних кольорів і розмірних сіток та відображає підібраний асортимент користувачу. Визначившись із конкретною модифікацією речі, клієнт додає її до кошика, що фіксується системою у відповідній таблиці збережених елементів.

На етапі переходу до оформлення замовлення покупець обирає місто та конкретне відділення Нової Пошти, після чого модуль логістики повертає список точок видачі та автоматично розраховує вартість транспортування. Далі клієнт підтверджує намір здійснити покупку, вказуючи адресу й метод оплати. Модуль замовлень миттєво перевіряє актуальну кількість товару на складі й за умови його наявності записує фінальні дані про транзакцію в базу даних. Останнім кроком модуль замовлень передає запит платіжному шлюзу [18], який через зовнішній API Monobank генерує інвойс та повертає покупцеві пряме посилання на платіжну сторінку банку для завершення розрахунку.

Діаграма послідовності демонструє повний процес оформлення замовлення (рис. 2.5).

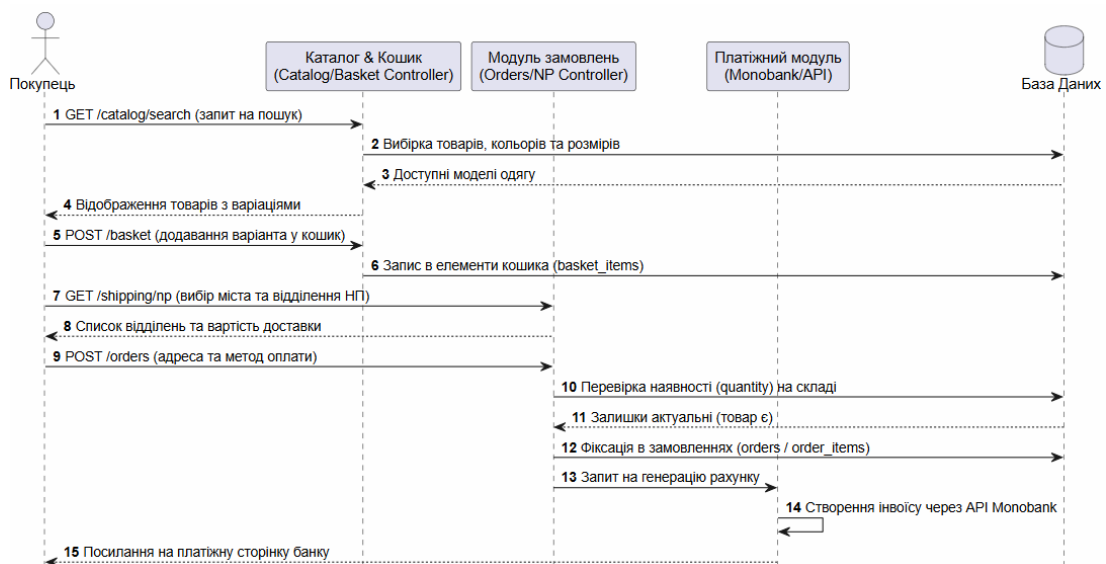


Рис. 2.5 Діаграма послідовності оформлення замовлення

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

Процес створення одягу за персональними параметрами починається із запиту покупця на отримання структури індивідуальної форми для обраного типу виробу. Контролер кастомізації завантажує з бази даних спеціальну JSON-конфігурацію полів, яка містить вимоги до мірок, а також пресети доступних ексклюзивних матеріалів і палітри кольорів. На основі цих даних інтерфейс сайту рендерить форму, де клієнт вказує свою стать, зріст, індивідуальні заміри, залишає текстовий коментар і відправляє заявку на сервер, де вона зберігається в базі із початковим статусом очікування.

Далі до роботи долучається адміністратор платформи, який через панель керування робить запит на перегляд нових індивідуальних форм. Контролер кастомізації підтягує дані мірок клієнтів із бази даних і виводить їх на екран адмін-панелі для модерації. Менеджер оцінює технічну можливість пошиття речі, вибирає дію підтвердження форми, і система автоматично змінює статус заявки на схвалений. Фінальний крок ініціює сам покупець, відправляючи запит на оплату підтвердженого замовлення, після чого платіжний модуль перевіряє успішний статус модерації в базі даних та повертає посилання на онлайн-оплату від Monobank.

Діаграма послідовності демонструє повний процес оформлення замовлення (рис. 2.6).

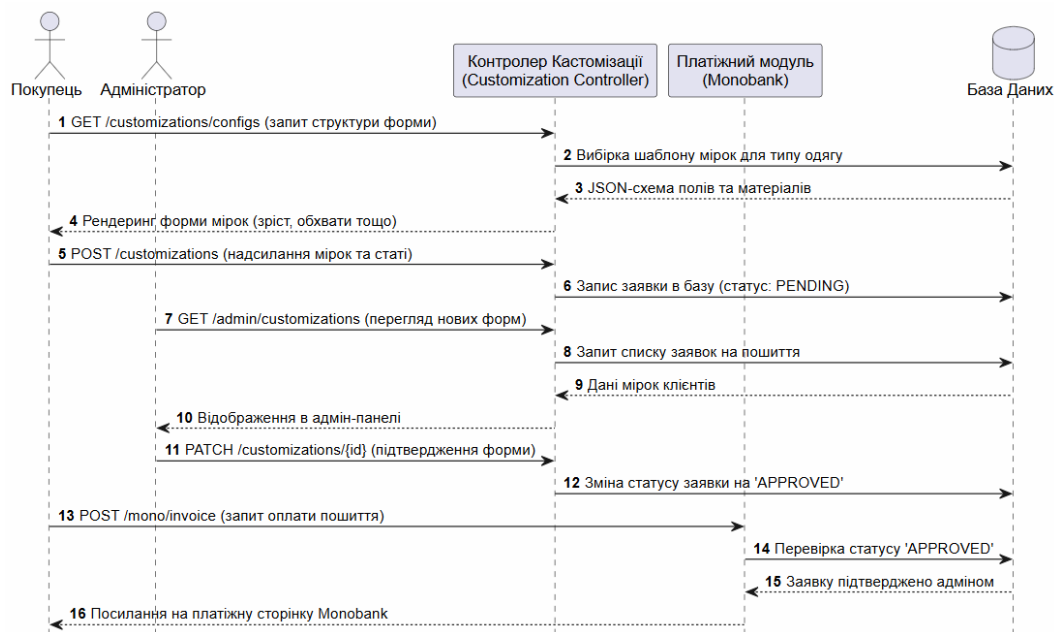


Рис. 2.6 Діаграма послідовності оформлення індивідуального замовлення

Життєвий цикл замовлення розпочинається у стані очікування PENDING, коли покупець підтверджує намір придбати одяг і система фіксує нову транзакцію, перевіряючи складські залишки. Наступний етап настає тоді, коли клієнт успішно перераховує кошти через платіжний шлюз, внаслідок чого транзакція верифікується банком, а саме замовлення автоматично набуває статусу оплаченого PAID. У цьому стані менеджер платформи бере замовлення в роботу, формує посилку на складі та передає її логістичній службі.

Після того як товар фізично відправлено одержувачу Новою Поштою та створено експрес-накладну, замовлення переходить у стан відправленого SHIPPED. Фінал взаємодії настає у момент, коли клієнт забирає посилку з відділення або поштомату і транспортна компанія підтверджує успішне вручення вантажу. Після цієї фіксації замовлення набуває фінального статусу доставленого DELIVERED, що ознаменовує повне завершення угоди та закриття циклу покупки.

Діаграма станів замовлення представлена на рис. 2.7.

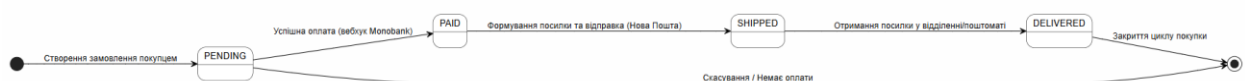


Рис. 2.7 Діаграма станів замовлення

Процес інтеграції з логістичним сервісом Нова Пошта починається з того, що покупець на сторінці оформлення замовлення вказує місто доставки. Інтерфейс сайту миттєво перехоплює цю дію та відправляє HTTP-запит до сервера. Отримавши цей запит, сервер системи спочатку перевіряє внутрішній кеш на наявність збереженої інформації для цього міста. У разі успішного знаходження даних у кеші сервер миттєво витягує готовий список точок видачі та повертає його інтерфейсу, що дозволяє уникнути зайвого навантаження на мережу та прискорити відгук сайту. Якщо ж у кеші інформація відсутня або термін її дії закінчився, сервер ініціює пряме звернення через зовнішнє API Нової Пошти, відправляючи запит на отримання актуальних складів. Отриману відповідь сервер паралельно

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

записує у свій кеш із встановленим часом актуальності для подальшого повторного використання і повертає список відділень та поштоматів на екран користувача.

Після того як покупець визначається з конкретною точкою видачі, система переходить до етапу обчислення вартості транспортування. Інтерфейс передає серверу пункт призначення, а сервер відправляє прямий запит до АРІ Нової Пошти для розрахунку ціни доставки. Логістичний сервіс повертає точну вартість посилки, яку сервер приймає і передає на фронтенд для демонстрації покупцеві.

Діаграма станів послідовності розрахунку вартості доставки зображена на рис. 2.8.

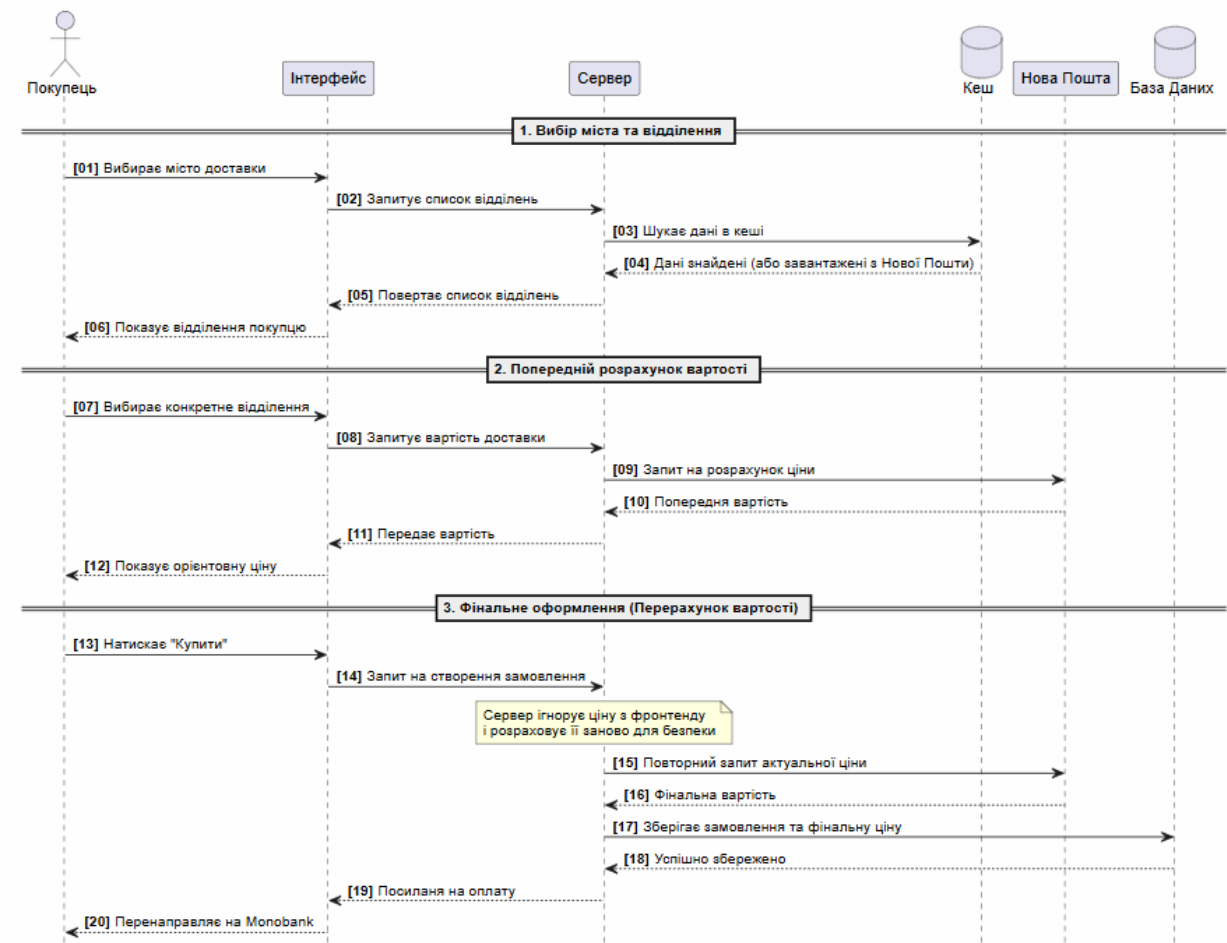


Рис. 2.8 Діаграма станів послідовності розрахунку вартості доставки

2.4. Реалізація функціоналу

Процес взаємодії з платіжною системою Monobank складається з двох автономних, але логічно пов'язаних етапів: ініціалізації інвойсу під час оформлення замовлення користувачем та асинхронного приймання результатів транзакції через вебхук [19]. Обидві операції спроектовані з урахуванням суворих вимог до цілісності даних у реляційній базі даних та вимагають обов'язкової верифікації стану об'єктів.

Процедура створення платіжної сесії починається з ідентифікації клієнта через JWT-токен або спеціалізований заголовок авторизації. Після успішної аутентифікації система відкриває транзакцію бази даних, щоб гарантувати узгодженість операцій. Першочергово виконується завантаження поточного вмісту кошика користувача за допомогою SQL-запиту [20], який об'єднує таблиці елементів кошика, базових продуктів та їхніх конкретних модифікацій.

```
$sql = <<<SQL
SELECT
    b.quantity,
    p.id AS "productId",
    p.title,
    p.slug,
    v.id AS "variantId",
    v.size AS "selectedSize",
    v.color AS "selectedColor",
    v.price,
    v.images[1] AS "image"
FROM basket_items b
JOIN products p ON b.product_id = p.id
JOIN product_variants v ON b.variant_id = v.id
WHERE b.user_id = :user_id
SQL;
```

Якщо кошик порожній, транзакція негайно скасовується. За наявності товарів система переходить до калькуляції фінансових показників. Базова вартість розраховується шляхом підсумовування добутків ціни та кількості для кожної позиції. Після цього, у разі використання сервісу доставки «Нова

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

Пошта», відбувається динамічний запит до логістичного контролера для обчислення вартості транспортування на основі ідентифікатора міста, базової ваги та оціночної вартості. Сумарна сума замовлення формується як поєднання базової вартості та отриманого логістичного тарифу.

Наступним кроком є фіксація замовлення в базі даних зі статусним маркером PENDING. Система виконує параметризований запит додавання запису до таблиці замовлень, використовуючи механізм RETURNING id для миттєвого отримання згенерованого первинного ключа в межах поточної сесії. Після успішного збереження заголовка замовлення, у циклі виконується наповнення таблиці специфікації, де фіксуються історичні ціни та обрані характеристики товарів на момент покупки. Як тільки специфікація сформована, система повністю очищає кошик користувача і фіксує транзакцію бази даних за допомогою методу commit.

Перед відправкою запиту до банку система трансформує масив товарів у формат, що вимагається специфікацією Monobank API. Особливістю цієї трансформації є переведення всіх фінансових значень з плаваючою крапкою у цілі числа, де одиницею виміру виступають копійки. Масив кошика збагачується текстовими назвами модифікацій, кодами варіантів та одиницями виміру. Контролер формує фінальне корисне навантаження запиту, яке містить загальну суму в копійках, код валюти гривні, URL-адресу перенаправлення для клієнта, адресу вебхуку для асинхронних сповіщень та час дії інвойсу.

```
$invoicePayload = [  
    'amount' => (int) round($totalAmount * 100),  
    'ccy' => 980,  
    'redirectUrl' => $frontendBase . '/checkout/success?orderId=' . $orderId,  
    'webhookUrl' => $webhookBase . '/mono/webhook',  
    'validity' => 3600,  
    'paymentType' => 'debit',  
    'basket' => $monoBasket,  
];
```

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

Запит надсилається за допомогою cURL до API мерчанта Monobank із обов'язковим додаванням секретного токена авторизації X-Token у заголовки. Якщо зовнішній шлюз повертає HTTP-статус, відмінний від 200, або у відповіді відсутній унікальний ідентифікатор інвойсу, система переходить у режим обробки помилок. Оскільки кошик на цьому етапі вже очищено, замовлення не видаляється, проте його стан оновлюється — у поле платіжних деталей записується весь сирий текст помилки від банку, а клієнту повертається помилка з кодом 502 [21]. При успішній відповіді банку отриманий invoiceId та посилання на платіжну сторінку returnUrl записуються в базу даних, а структура з цими даними повертається на фронтенд для перенаправлення користувача.

Вебхук виступає в ролі ізольованої точки доступу, яку платіжна система Monobank викликає автоматично після будь-якої зміни статусу виставленого інвойсу. Надійна обробка цього запиту є критичною, оскільки клієнт на момент оплати вже може закрити сторінку магазину. Процедура обробки починається з первинної перевірки структури вхідного JSON-пакета: якщо тіло запиту порожнє або не є масивом, система реєструє помилку в логах сервера та повертає статус 400.

З отриманого пакета виділяється унікальний ідентифікатор invoiceId. Замість потенційно вразливої процедури локальної перевірки криптографічного підпису запиту, архітектура системи використовує безпечніший патерн прямого зворотного підтвердження. Система ініціює самостійний запит до API Monobank, запитуючи актуальний та гарантовано правдивий статус інвойсу безпосередньо у банку. Якщо банківський сервіс недоступний або повертає помилку, обробка негайно переривається з кодом 400, запобігаючи несанкціонованій зміні статусів у базі даних.

Після отримання верифікованого статусу з боку банку здійснюється пошук відповідного замовлення в базі даних. Пошук виконується за допомогою SQL-оператора, який аналізує збережені JSON-дані платіжних

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

деталей за допомогою оператора стрілки, оптимізованого під індекси jsonb в PostgreSQL.

```
SELECT id, status FROM orders WHERE payment_details::jsonb->>'invoiceId' = :invoice_id LIMIT 1
```

Якщо замовлення з таким ідентифікатором інвойсу не знайдено в системі, контролер реєструє попередження в лог-файлах і повертає успішну відповідь платіжній системі, щоб банк припинив повторні відправки цього сповіщення. Якщо ж запис знайдено, система аналізує поточний статус замовлення та новий статус від банку. Для переведення банківських статусів у внутрішні бізнес-статуси застосовується конструкція порівняння match.

Банківський маркер success трансформується у внутрішній стан замовлення PAID, маркер failure перетворюється на статус FAILED, а статус reversed відповідає стану повернення коштів REFUNDED. Будь-які інші проміжні або невідомі статуси ігноруються, повертаючи нульове значення. Модифікація бази даних відбувається лише за умови, що новий розрахований статус не є порожнім і водночас відрізняється від поточного статусу замовлення — це захищає систему від повторної обробки однакових пакетів.

```
if ($newStatus !== null && $order['status'] !== $newStatus) {  
    error_log(sprintf('Monobank Webhook: Altering order %s status to %s', $order['id'], $newStatus));  
  
    $updateStmt = $pdo->prepare(  
        "UPDATE orders SET status = :status, payment_details = :pd WHERE id = :id"  
    );  
    $updateStmt->execute([  
        'status' => $newStatus,  
        'pd' => json_encode($invoiceData),  
        'id' => $order['id'],  
    ]);  
    error_log(sprintf('Monobank Webhook: Database update committed for order %s.', $order['id']));  
}
```

Під час оновлення запису в базу даних записується не лише новий текстовий статус, а й повністю перезаписується поле payment_details, куди вноситься актуальний зліпок даних від банку, включаючи фінансові мітки часу та параметри авторизації транзакцій. Якщо процес оновлення пройшов успішно, або якщо модифікація не була потрібна через збіг статусів,

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

контролер завершує роботу, відправляючи банку структуру `JsonResponse::ok()`. Будь-які критичні збої, що виникають під час парсингу чи взаємодії з базою даних, перехоплюються загальним блоком `catch`, який детально документує стек викликів помилки та повертає платіжному шлюзу HTTP-код 500, сигналізуючи про необхідність повторити відправку вебхуку пізніше.

Висновки до другого розділу

У межах дослідження було виконано комплексний аналіз предметної області інтернет-магазину в'язаних handmade-виробів із застосуванням методології об'єктно-орієнтованого проєктування. Завдяки цьому вдалося чітко окреслити функціональні вимоги до платформи, визначити ключових суб'єктів взаємодії покупця та адміністратора, а також детально розписати їхні базові сценарії роботи з вебдодатком.

Для наочного представлення внутрішньої логіки та архітектури системи було сформовано діаграми прецедентів і концептуальних класів. Це дало змогу побудувати гнучку модель розмежування прав доступу на основі JWT-авторизації, ізолювати дані про асортимент товарів та їхні фізичні параметри, а також оптимізувати зв'язки між кошиком покупця та підсумковими замовленнями.

Інформаційну модель реалізовано у вигляді реляційної бази даних під керуванням СУБД PostgreSQL. Для підвищення гнучкості системи та збереження детальної історії фінансових операцій було задіяно технологію JSONB, яка дозволяє зберігати неструктуровані зліпки банківських відповідей.

Динамічні аспекти та внутрішні стани системи зафіксовані за допомогою діаграм активності, послідовності та станів. Особливу увагу було приділено оптимізації критичних бізнес-процесів: алгоритму формування замовлення з інтеграцією логістичного сервісу Нова Пошта для розрахунку вартості доставки, життєвому циклу транзакцій, а також логіці виставлення інвойсів та обробки вебхуків від платіжного шлюзу Monobank.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

Програмну реалізацію серверної частини платформи виконано мовою РНР у вигляді модульного та швидкодійного API. Взаємодію із зовнішнім банківським сервісом організовано через ізольований платіжний сервіс за допомогою запитів до API мерчанта. Захист від підміни даних та фінансових маніпуляцій забезпечено завдяки архітектурному патерну зворотного підтвердження статусів та механізму ізоляції транзакцій БД.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ІНТЕРНЕТ-МАГАЗИНОМ В'ЯЗАНОГО ОДЯГУ

3.1. Порядок розгортання та налаштування параметрів платформи

Розгортання та налаштування параметрів системи передбачає підготовку серверного середовища, запуск архітектурних компонентів платформи електронної комерції та забезпечення взаємодії між вебдодатками й базами даних. Вебплатформа функціонує в межах монолітної архітектури виконуються всередині спільного серверного процесу.

До складу інфраструктури інтернет-магазину входить серверний API-додаток на базі мови PHP. Користувацьку частину інтерфейсу реалізовано у вигляді сучасного вебдодатка на базі React із підтримкою технології серверного рендерингу, що дозволяє оптимізувати початкове завантаження сторінок та покращити індексацію пошуковими роботами та SEO.

Для збереження комерційних даних та інформації про асортимент задіяно реляційну СУБД PostgreSQL. Інфраструктура також включає високопродуктивне сховище Redis, яке виконує функції розподіленого кешування та менеджменту сесій для зниження навантаження на базу даних. Збереження статичного контенту покладено на об'єктне хмарне сховище S3, а обробка платежів та перевірка транзакцій реалізована через інтеграцію з API-інтерфейсом банку Monobank.

Для спрощення розгортання, ізоляції компонентів та швидкого локального тестування вся архітектура платформи описана за допомогою декларативних конфігурацій інструменту оркестрації Docker Compose. Це дає змогу розгортати та масштабувати інфраструктуру у вигляді пов'язаних контейнерів, гарантуючи ідентичність поведінки коду під час розробки, тестування та фінального релізу на продуктивному сервері.

Процес розгортання вебплатформи складається з кількох послідовних етапів. Спочатку на цільовому сервері під керуванням операційної системи

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

Linux Ubuntu 22.04 налаштовується середовище виконання Docker Daemon та утиліта Docker Compose [22]. На наступному кроці актуальна версія вихідного коду завантажується з віддаленого репозиторію через систему контролю версій Git. Далі створюються та заповнюються локальні файли конфігурації оточення.

Окремим важливим етапом автоматизації розгортання є синхронізація структури реляційної бази даних, яка реалізована за допомогою інструментарію командного рядка Supabase CLI. Усі зміни в схемі даних, такі як створення таблиць асортименту, замовлень і користувачів, а також налаштування зв'язків, індексів та тригерів, фіксуються у вигляді послідовних файлів міграцій у локальному репозиторії проєкту.

Під час ініціалізації інфраструктури через Docker Compose спеціальний службовий контейнер із встановленим Supabase CLI авторизується у віддаленій хмарі Supabase за допомогою унікального токена проєкту та сервісного ключа доступу. Перед запуском основних бізнес-компонентів платформи, а саме серверного додатка на PHP та клієнтського на React, утиліта автоматично аналізує стан віддаленої бази даних PostgreSQL. Вона порівнює його з локальною історією міграцій і послідовно застосовує лише нові файли через команду `supabase db push`. Це гарантує повну цілісність даних та виключає ризик помилок через різницю в структурі таблиць між комп'ютером розробника та живим сервером.

На фінальному етапі ініціюється команда збірки та підняття контейнерів Docker Compose, яка автоматично створює ізольовану віртуальну мережу, запускає сервери Redis, сервіс React SSR та PHP. Зворотний проксі-сервер Nginx починає маршрутизувати вхідний трафік, розподіляючи запити між клієнтським SSR-додатком та серверною PHP-частиною.

Для представлення процесу розгортання було наведено діаграму на рис. 3.1.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

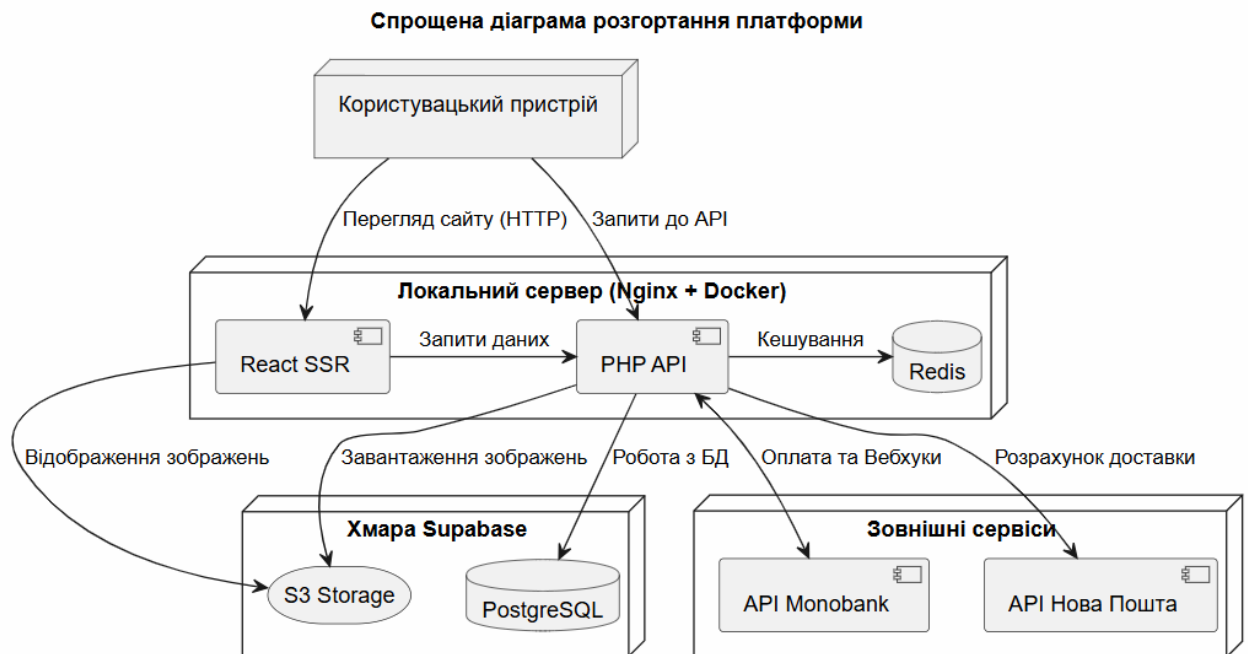


Рис. 3.1. Діаграма розгортання

Структура розгортання містить вебсервер Nginx, який виступає єдиною точкою входу для зовнішніх HTTP-запитів. Він перенаправляє запити користувачів на контейнер React SSR для рендерингу клієнтського інтерфейсу або на контейнер PHP для обробки бізнес-логіки. Внутрішній шар системи формують локальний контейнер Redis та підключені зовнішні сервіси: логістичне API Нова Пошта, платіжний шлюз Monobank, а також хмарна інфраструктура Supabase, що хостить базу даних PostgreSQL та об'єктне S3-сховище [23].

До параметрів конфігурації вебплатформи належать адреси підключення до віддаленої бази даних Supabase, авторизаційні ключі до хмарного S3-сховища, налаштування безпеки JWT, секретні токени зовнішніх систем, а також ключі платежів[24]. Усі параметри розділені на дві категорії. До першої належать публічні конфігураційні змінні: URL-адреси шлюзів, рівні логування помилок, правила CORS для безпечного крос-доменного доступу та часові таймаути [25]. Другу категорію складають чутливі секретні дані: паролі до БД Supabase, приватні ключі шифрування токенів, API-ключі Monobank та Нової Пошти.

Публічні параметри та базові ліміти задаються у файлах конфігурації всередині вихідного коду додатків, тоді як усі секретні параметри та рядки підключення до хмари з міркувань безпеки передаються в контейнери динамічно через змінні середовища операційної системи під час ініціалізації Docker Compose [26].

3.2. Структура інтерфейсу магазину

Клієнтська частина платформи складається з двох основних інтерфейсів: інтернет-магазину та панелі адміністратора. Наведено основні екрани інтерфейсу з описом використання вебсайту.

На рис. 3.2 представлено інтерфейс авторизації користувачів, виконаний у мінімалістичному стилі з використанням монохромної палітри кольорів. У верхній частині сторінки розміщено уніфіковану навігаційну панель із логотипом бренду, пошуковим рядком, іконкою кошика та функціональним меню. Центральну частину займає форма входу, яка містить текстові поля для введення електронної пошти та паролю, посилання для відновлення доступу, а також кнопку авторизації із правосторонньою стрілкою.

Рис. 3.2. Сторінка входу

На рис. 3.3 зображено інтерфейс створення нового облікового запису. Форма містить три основні текстові поля для заповнення персональних даних користувача: повне ім'я, електронна пошта та пароль. На знімку екрана

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

продемонстровано роботу клієнтської валідації введених даних: система підсвічує обов'язкові для заповнення поля червоними маркерами попередження, блокуючи відправку форми в разі некоректного або неповного введення інформації перед реєстрацією в системі.

Рис. 3.3. Сторінка реєстрації

На рис. 3.4 представлено головну сторінку вебсайту, що виконує роль презентаційного та навігаційного хабу системи. Функціональна структура сторінки містить фіксовану верхню панель, яка забезпечує глобальну навігацію між основними розділами сайту, інтегровану пошукову стрічку для швидкого знаходження позицій за ключовими словами, модуль кошика користувача, а також блоки авторизації та реєстрації. Центральну частину займає банерна hero-секція, яка поєднує текстовий маркетинговий блок з кнопкою швидкого переходу «Купити зараз».

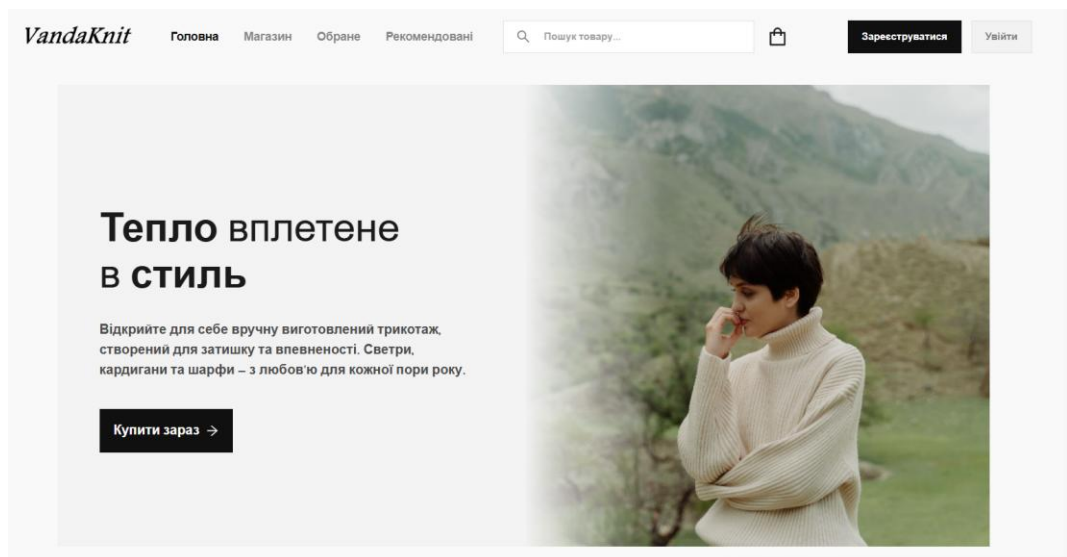


Рис. 3.4. Головна сторінка

На рис. 3.5 зображено інтерфейс сторінки магазину із реалізованою товарною сіткою. Окрім збереження наскрізної навігаційної панелі сайту, функціонал сторінки розширено спеціалізованими інструментами електронної комерції: додано кнопку «Фільтри» для сортування та відбору продукції за технічними характеристиками, а також інтегровано пошуковий рядок безпосередньо над товарними картками. Інтерфейс демонструє картки товарів, кожна з яких є інтерактивним функціональним вузлом. У межах однієї картки реалізовано: виведення графічного зображення виробу, назву та матеріал, селектор вибору доступних кольорів у вигляді круглих інтерактивних іконок, динамічне відображення ціни та кнопку безпосереднього додавання позиції до кошика.

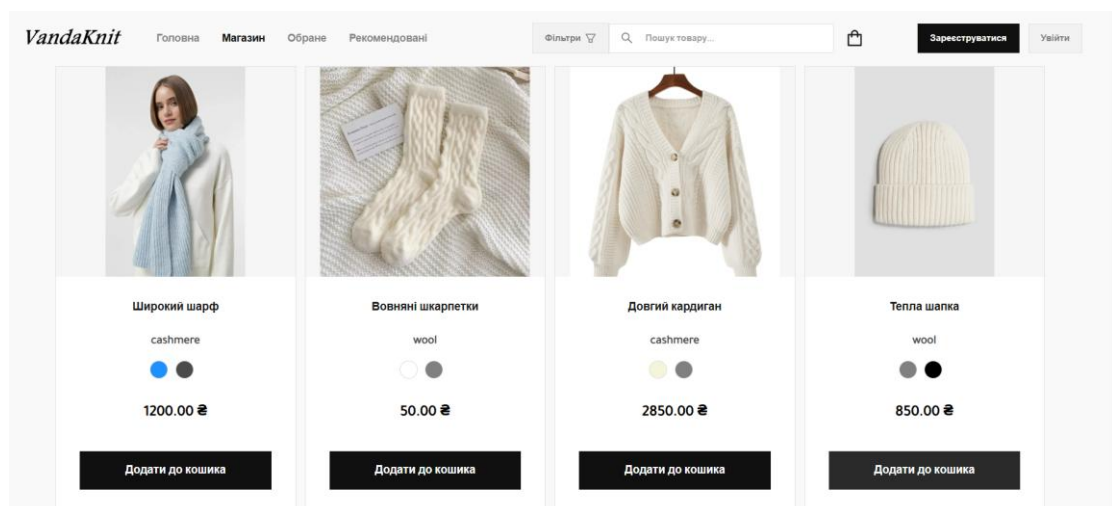


Рис. 3.5. Магазин товарів

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

На рис. 3.6 продемонстровано роботу системи динамічного зворотного зв'язку вебсайту під час взаємодії користувача з каталогом. Функціонал іконки кошика у верхній панелі автоматично оновлюється: на ній з'являється числовий індикатор червоного кольору «1», що сигналізує про успішне додавання одиниці товару без перезавантаження сторінки. Паралельно у верхній правій частині екрана реалізовано спливаюче інтерактивне сповіщення з текстовим підтвердженням «Товар додано до кошика». Візуально це повідомлення оформлене в м'яких зелених тонах, що відповідає загальноприйнятим стандартам UX-дизайну для успішних операцій та допомагає користувачеві миттєво зафіксувати результат своєї дії.

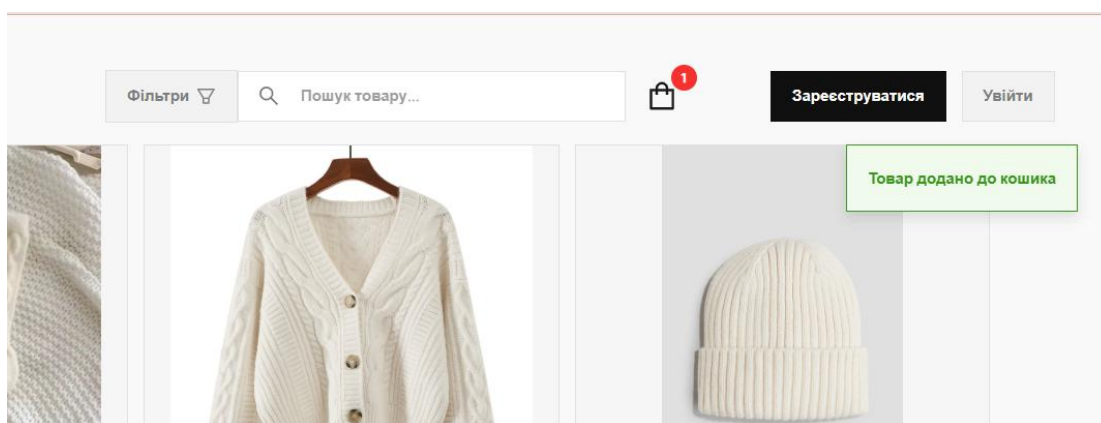


Рис. 3.6. Доданий товар

На рис. 3.7 відображено функціонал інтерактивного кошика, реалізованого у вигляді висувної бічної панелі, яка перекриває основний контент каталогу з правого боку. У верхній частині панелі розміщено керуючі елементи «Закрити» та «Очистити кошик». Функціонал картки обраного товару в кошику містить: мініатюру зображення виробу, його назву, автоматично визначений розмір, маркер обраного кольору, ціну, а також кнопку швидкого видалення позиції. Також додано кнопки зміни кількості товару в кошику. У нижній частині панелі виводиться автоматично обчислювана сума замовлення та кнопка оформлення замовлення, що ініціює перехід до наступного кроку воронки продажів.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

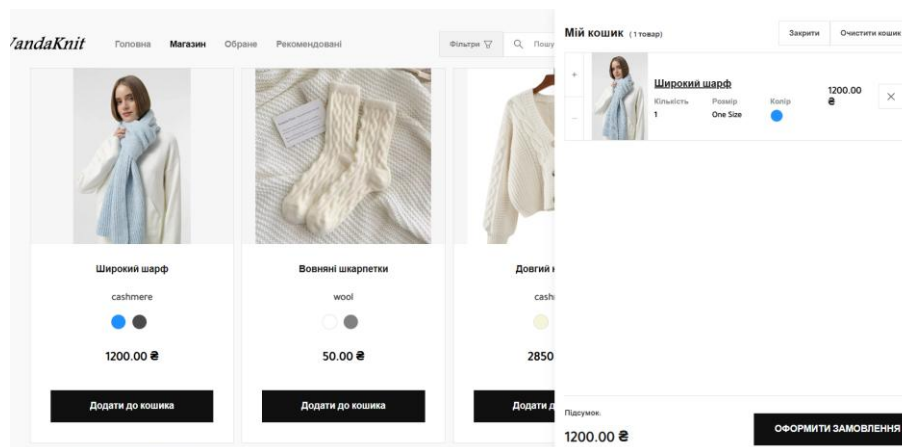


Рис. 3.7. Кошик

На рис. 3.8 показано роботу механізму автентифікації користувача на етапі оформлення покупки за допомогою модального вікна. Під час спроби неавторизованого клієнта перейти до оплати, система блокує фоновий інтерфейс та виводить у фокус діалогове вікно з попередженням. Це вікно містить два альтернативні функціональні сценарії у вигляді кнопок: «Продовжити покупки» та «Увійти для оформлення».

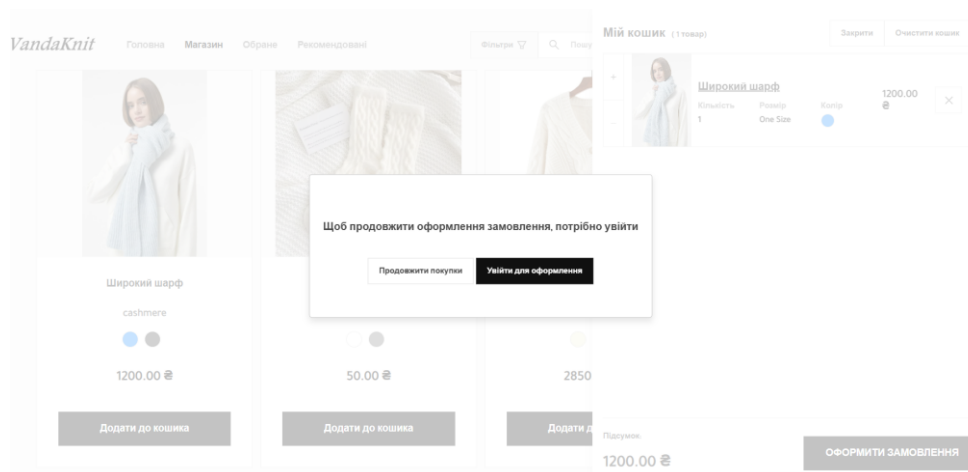


Рис. 3.8. Автентифікація на етапі оформлення покупки

На рис. 3.9 реалізовано функціонал кошика та першого кроку оформлення замовлення, де клієнт може детально перевірити свій вибір перед покупкою. Інтерфейс відображає список доданих товарів із їхніми ключовими атрибутами фотографією, назвою, кількістю, розміром та кольором, а також дозволяє змінювати обсяг замовлення або видаляти позиції. Система автоматично підраховує проміжну та фінальну суму,

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

надаючи користувачеві вибір: повернутися до каталогу товарів чи перейти до наступного етапу оформлення.

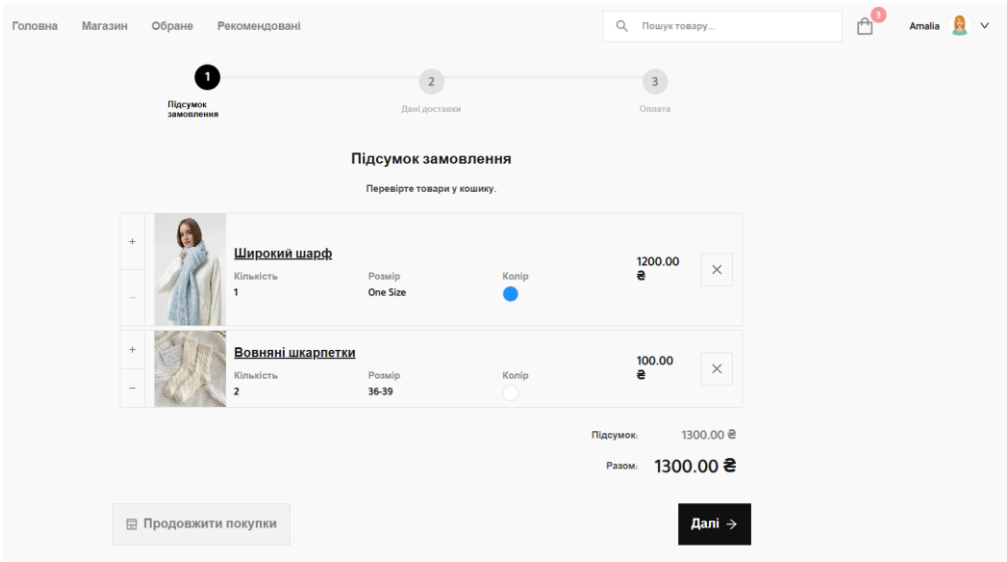


Рис. 3.9. Перший крок оформлення замовлення

На рис. 3.10 представлено функціонал блоку доставки, який забезпечує збір контактних даних та інтеграцію з логістичним оператором. Користувач заповнює обов'язкові поля з іменем та номером телефону, після чого обирає населений пункт. Головною особливістю цього кроку є динамічний випадаючий список, який підтягує актуальну базу відділень та поштоматів Нової Пошти для обраного міста, дозволяючи клієнту швидко та безпомилково вказати точну адресу для отримання посилки.

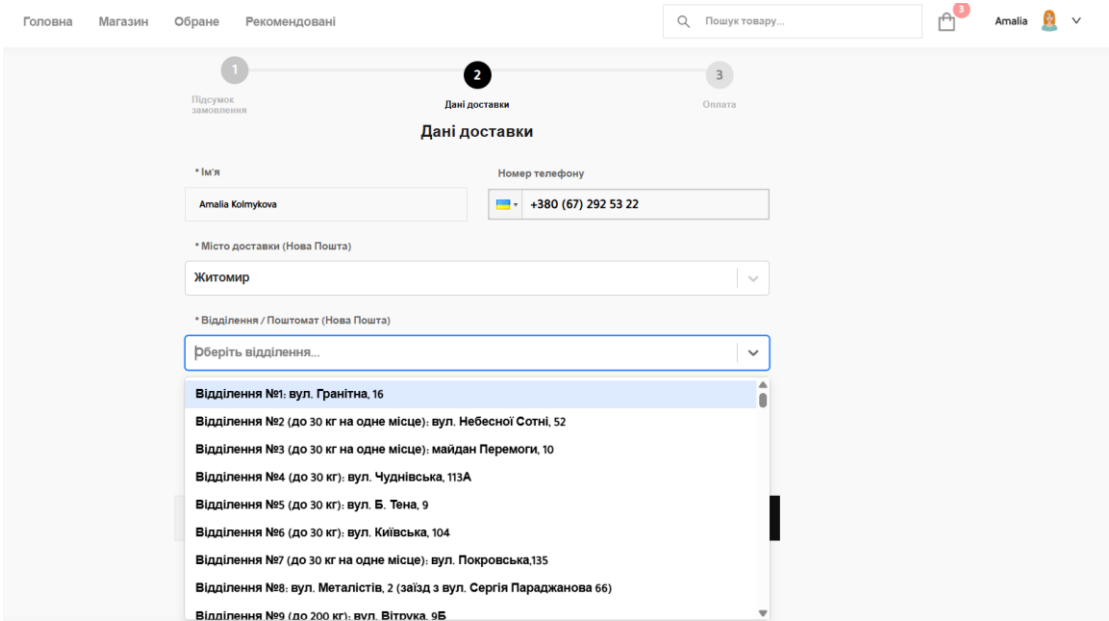


Рис. 3.10. Блок оформлення доставки

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

На рис. 3.11 зображено заповнену форму даних доставки після того, як користувач обрав конкретне відділення логістичного оператора. Функціонал системи на цьому етапі автоматично розраховує та додає вартість доставки до загальної вартості замовлення, оновлюючи фінальну суму до оплати. Також стає активною навігація, що дозволяє користувачеві повернутися на попередній крок або перейти до фінального кроку.

Рис. 3.11. Розрахунок доставки

На рис. 3.12 реалізовано фінальний крок оплати, який виконує функцію перевірки перед транзакцією та інтеграції з платіжним шлюзом. Інтерфейс відображає всю введену раніше інформацію: обрані товари, дані про доставку, а також повну вартість. Крок завершується кнопкою оплати з Monobank.

Рис. 3.12. Крок оплати

На рис. 3.13 реалізовано сторінку профілю клієнта. Цей функціонал призначений для відображення основної інформації про користувача в режимі читання. Тут виводяться завантажений аватар, повне ім'я, електронна пошта, збережена адреса доставки, номер телефону, а також дата реєстрації аккаунта. Навігація зверху дозволяє перемикатися між персональними даними, списком бажань та історією замовлень.

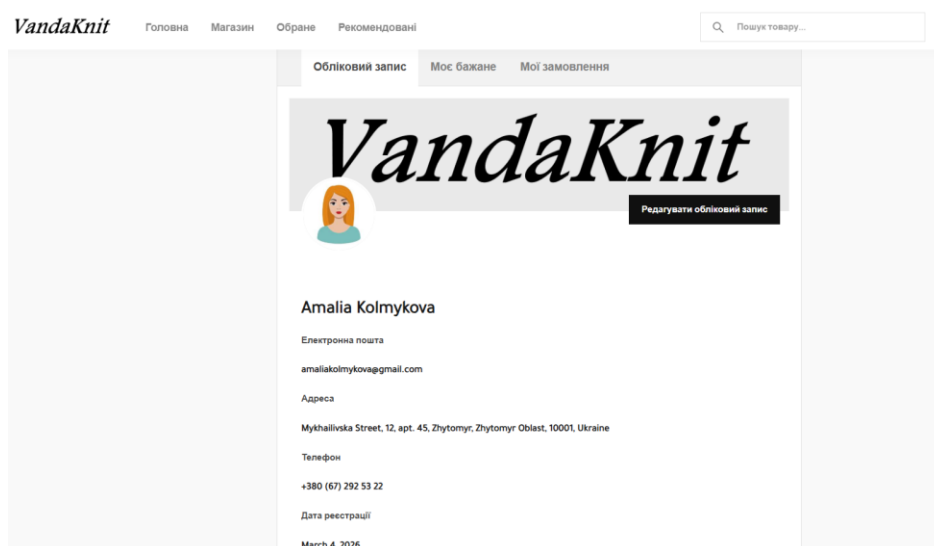


Рис. 3.13. Обліковий запис користувача

На рис. 3.14 показано роботу форми редагування профілю користувача. Функціонал цього кроку дозволяє клієнту оновлювати свої персональні та контактні дані у відповідних полях, які надалі автоматично заповнюватимуть форми під час оформлення замовлень. Крім того, реалізовано можливість інтерактивної зміни зображення аватара та фонового банера за допомогою іконок редагування.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

VandaKnit

Amalia Kolmykova

* Повне ім'я

* Електронна пошта

amaliakolmykova@gmail.com

Адреса (використовується при оформленні)

Mykhailivska Street, 12, Apt. 45, Zhytomyr, Zhytomyr Oblast, 10001, Ukraine

Номер телефону (використовується при оформленні)

+380 (67) 292 53 22

← Назад до профілю

✓ Оновити профіль

Рис. 3.14. Редагування профілю

На рис. 3.15 показано роботу сторінку замовлень користувача. Список відображає дату, суму та статус замовлень з можливістю перейти до перегляду їх деталей.

Головна Магазин Обране Рекомендовані

Пошук товару...

Обліковий запис Моє бажане **Мої замовлення**

Мої замовлення

Замовлення №	Дата	Статус	Сума
Замовлення №17	June 20, 2025	PENDING	1370.00 €
Замовлення №16	June 19, 2025	PAID	120.00 €
Замовлення №15	June 19, 2025	PAID	120.00 €
Замовлення №14	June 19, 2025	PENDING	120.00 €
Замовлення №13	June 19, 2025	PENDING	120.00 €

Рис. 3.15. Перегляд здійснених замовлень

На рис. 3.16 реалізовано панель керування, яка слугує головним центром аналітики для моніторингу бізнесу. У верхній частині виведені інформаційні картки з ключовими кількісними показниками системи: загальна кількість клієнтів, товарів у базі та оформлених замовлень. Основну

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

площу займають модулі візуалізації даних: динамічна стовпчикова діаграма, що відображає помісячний дохід із функцією інтерактивних підказок, а також кругова діаграма, яка класифікує замовлення за їхніми поточними статусами: оплачено, відправлено, доставлено або очікується.

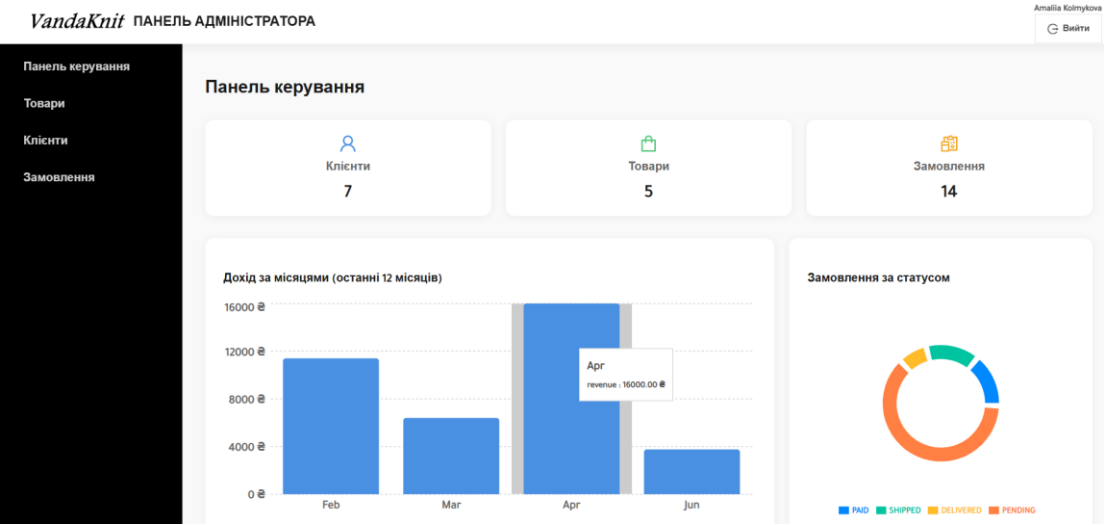


Рис. 3.16. Адмін панель керування

На рис. 3.17 показано інтерфейс вкладки товарів, призначений для безпосереднього CRUD-управління асортиментом магазину. Зверху розміщено інструменти швидкого пошуку, фільтрації та кнопку додавання нових товарів. Основна частина функціонала реалізована у вигляді інтерактивної таблиці, де виводиться список наявної продукції з зображеннями та ключовими атрибутами: назвою, доступними кольоровими маркерами, типом матеріалу, ціною та залишками на складі. Для кожного рядка передбачено інструменти оперативного редагування та видалення товарних позицій.

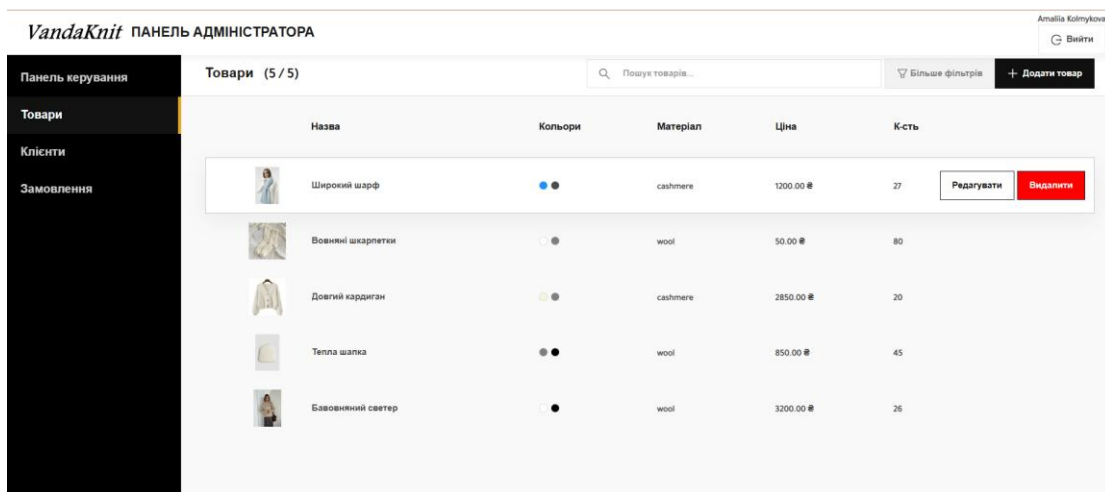


Рис. 3.17. Сторінка додавання та редагування товарів

На рис. 3.18 реалізовано інтерфейс вкладки клієнтів, який відображає таблицю зареєстрованих користувачів системи. Функціонал дозволяє адміністратору переглядати повний список покупців із їхніми аватарами, іменами, прив'язаними електронними адресами та точними датами реєстрації в системі. Навпроти кожного запису інтегровано кнопку редагування, що забезпечує швидкий перехід до профілю клієнта для зміни його даних або керування обліковим записом.

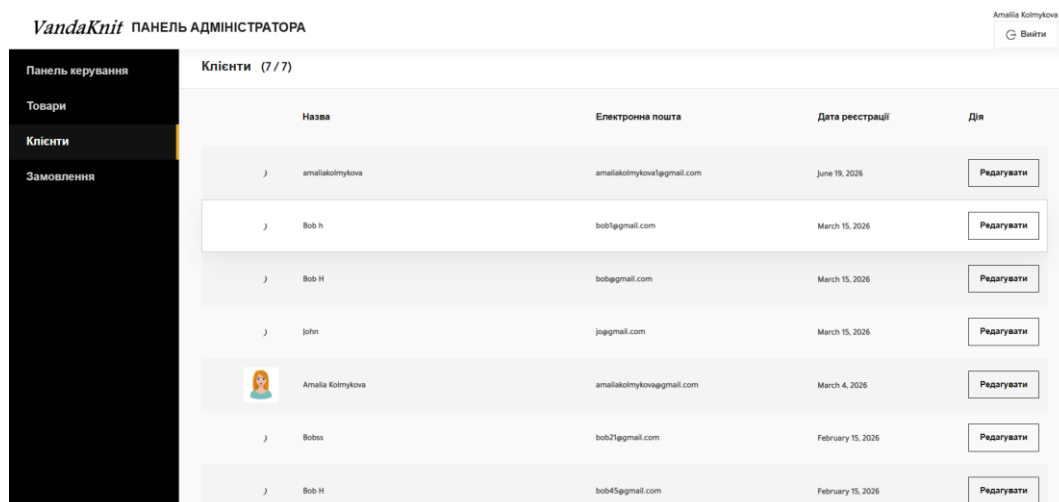


Рис. 3.18. Перегляд користувачів

На рис. 3.19 показано роботу системи управління замовленнями. Функціонал цієї вкладки призначений для трекінгу та адміністрування купівельної активності. Таблиця відображає ключову інформацію: унікальний ID замовлення, ім'я клієнта, дату створення, загальну суму чека

та поточний статус. Також зверху додано інструмент фільтрації за статусами для сортування заявок, а для менеджменту кожного окремого замовлення передбачено кнопку керування.

VandaKnit ПАНЕЛЬ АДМІНІСТРАТОРА Amalia Kolmyk [Вийти](#)

Панель керування

Товари

Клієнти

Замовлення

Замовлення (14 / 14) Усі статуси

ID замовлення	Клієнт	Дата	Сума	Статус	Дія
#17	Amalia Kolmykova	June 20, 2026	1370.00 ₴	РЕЗОНАНС	Керувати
#16	Amalia Kolmykova	June 19, 2026	120.00 ₴	ПАР	Керувати
#15	Amalia Kolmykova	June 19, 2026	120.00 ₴	ПАР	Керувати
#14	Amalia Kolmykova	June 19, 2026	120.00 ₴	РЕЗОНАНС	Керувати
#13	Amalia Kolmykova	June 19, 2026	120.00 ₴	РЕЗОНАНС	Керувати
#12	Amalia Kolmykova	June 19, 2026	120.00 ₴	РЕЗОНАНС	Керувати
#11	Amalia Kolmykova	June 19, 2026	520.00 ₴	РЕЗОНАНС	Керувати
#10	Amalia Kolmykova	June 19, 2026	1270.00 ₴	РЕЗОНАНС	Керувати

Рис. 3.19. Перегляд стадій виконання замовлення

На рис. 3.20 відображений внутрішній інтерфейс деталей замовлення. Екран дозволяє адміністратору бачити контактні дані покупця, логістичні параметри, фінансові підсумки та специфікацію замовленої продукції з конкретними характеристиками, такими як колір та розмір.

VandaKnit ПАНЕЛЬ АДМІНІСТРАТОРА Amalia Kolmyko [Вийти](#)

Панель керування

Товари

Клієнти

Замовлення

Кастомізації

Налаштування кастомізацій

Деталі замовлення №18 Статус: Очікує

Інформація про клієнта Ім'я: Amalia Kolmykova Електронна пошта: no-email@vanda-knit.com Телефон: 380672925322 Адреса: Місто: Житомир, Відділення: Відділення №4 (до 30 кг): вул. Чуднівська, 113А Міжнародна: Ні		Інформація про оплату Спосіб: monobank Дата замовлення: June 25, 2026 Підсумок: 350.00 ₴ Доставка: 70.00 ₴ Загальна сума: 420.00 ₴	
--	--	--	--

Замовлені товари

Товар	Варіант	Ціна	К-сть	Разом
 Вовняні шкарпетки	white / 36-39	50.00 ₴	7	350.00 ₴

Рис. 3.20. Перегляд деталей замовлення

На рис. 3.21 відображена інтерактивна сторінка кастомізації для клієнта. Інтерфейс містить покрокову форму, де покупець може самостійно

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

змоделювати одяг, вказавши власні антропометричні дані для подальшого надсилання індивідуальної заявки на виробництво.

Головна Магазин Обране Рекомендовані Кастомізація

Пошук товару...

Amalia

Створіть свій унікальний виріб

Оберіть тип виробу та вкажіть ваші параметри

* Тип виробу
Светр

* Зріст (см)
175

* Розмір (S, M, L...)
M

Надіслати запит

Рис. 3.21. Кастомізація замовлення покупцем

На рис. 3.22 відображена сторінка списку змодельованих продуктів. Інтерфейс відображає список спеціальних замовлень з можливістю створення продуктів на основі описів.

VandaKnit ПАНЕЛЬ АДМІНІСТРАТОРА

Панель керування	Запити на кастомізацію (4)
Товари	
Клієнти	
Замовлення	
Кастомізації	
Налаштування кастомізації	

Клієнт	Тип виробу	Параметри	Дата
Amalia amaliagmail.com	Кордиган	Колір: Червоний Стать: Жіноча	21 Jun 2026, 12:38
Додати товар			
Переглянути товар			
Amalia amaliagmail.com	Світр	Розмір (S, M, L...): 111 Зріст (см): 11	20 Jun 2026, 23:00
Додати товар			
Переглянути товар			
Amalia amaliagmail.com	Кордиган	Колір: Червоний Стать: Жіноча	20 Jun 2026, 22:52
Додати товар			

Рис. 3.22. Перегляд запитів на кастомізацію

На рис. 3.23 відображена сторінка налаштування однієї з форм для моделювання продукту. На сторінці можна налаштувати вид, назву та чергу полів форми для заповнення користувачами.

VandaKnit

ПАНЕЛЬ АДМІНІСТРАТОРА

Amalia Kolmykova

Вийти

Панель керування

Товари

Клієнти

Замовлення

Кастомізації

Налаштування кастомізацій

←

Редагувати Кардиган

Код типу (англійською, без пробілів)

cardigan

Відображувана назва

Кардиган

Поля (Характеристики)

Додайте поля, які будуть відображатися на формі кастомізації. Типи «Колір», «Матеріал» та «Стать» мають власні спеціальні елементи вводу.

Назва поля	Код (name)	Тип	Placeholder	Обов'язкове
Стать	gender	Стать		✓
Колір	color	Колір (з палітри)		✓
Додаткові побажання	additionalComment	Текстове поле	Ваші побажання	✓

+ Додати поле

Рис. 3.23. Конфігуратор форми для моделювання

3.3. Тестування вебплатформи

Перевірку відповідності розроблених програмних рішень вихідному технічному завданню було реалізовано за допомогою створення комплексу автоматизованих тестів у середовищі виконання РНР. Написані тести прикладного рівня спрямовані на сувору верифікацію логіки базових сервісів платформи та методів обробки HTTP-запитів, які відповідають за виконання ключових користувацьких та адміністративних сценаріїв. Такий підхід дозволив забезпечити стабільність кодової бази, суттєво спростивши пошук і локалізацію помилок на ранніх етапах розробки [27].

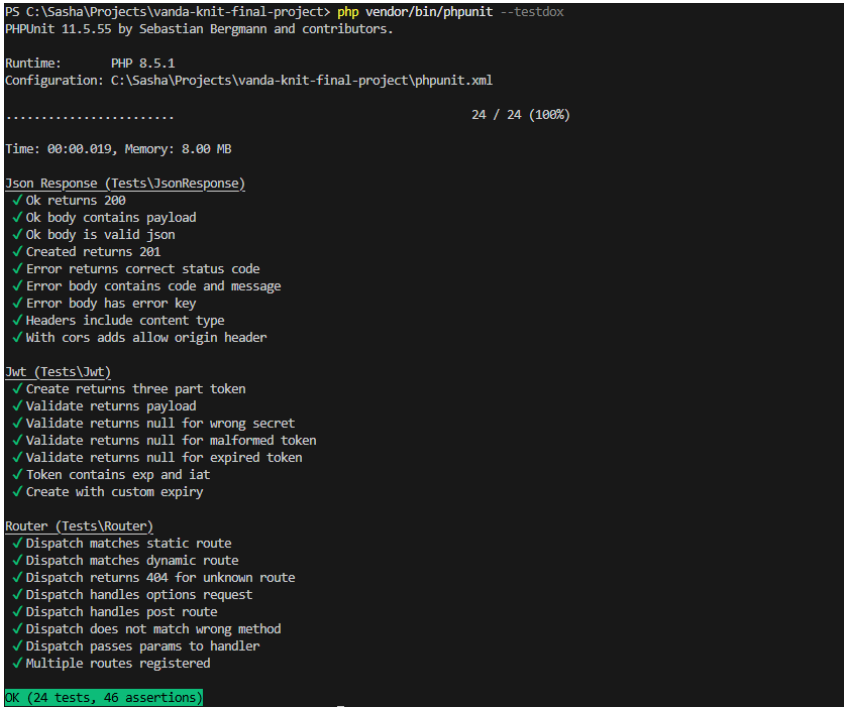
Під час проєктування тестової моделі вагомий акцент було зроблено на перевірці трьох системних компонентів серверної частини: підсистеми формування відповідей, механізму безпеки та модуля маршрутизації API. Перший блок тестів класу JsonResponse повністю покриває логіку генерації стандартних відповідей сервера, гарантуючи правильність повернення HTTP-статусів, коректність структури корисного навантаження, валідацію вихідного JSON-формату та автоматичне додавання заголовків безпеки, включаючи CORS-походження для безпечної крос-доменної взаємодії [28].

Наступний рівень автоматизованої перевірки охоплює критично важливі компоненти безпеки, зосереджені у модулі Jwt. Тестові сценарії

забезпечують повний контроль над процесами автентифікації користувачів через генерацію та валідацію трикомпонентних токенів доступу. Набір тестів успішно верифікує відхилення скомпрометованих запитів із неправильним секретним ключем, блокування деформованих пакетів та нівелювання авторизації у разі закінчення терміну дії токена на основі системних міток exp та iat. Своєю чергою, тести компонента Router підтверджують точність диспетчеризації статичних і динамічних маршрутів та коректність передачі параметрів безпосередньо до задіяних RHP-контролерів.

Повністю завершений та успішно виконаний цикл автоматизованого тестування, який продемонстрував стовідсоткове проходження всіх запланованих тестових сценаріїв за мінімальний проміжок часу, підтверджує високу надійність і швидкість архітектурної побудови серверної частини платформи. Результати свідчать про точність роботи ядра системи та безпеку обробки даних.

Загальний хід виконання тестів продемонстровано на рис. 3.24.



```
PS C:\Sasha\Projects\vanda-knit-final-project> php vendor/bin/phpunit --testdox
PHPUnit 11.5.55 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.5.1
Configuration: C:\Sasha\Projects\vanda-knit-final-project\phpunit.xml

.....
24 / 24 (100%)

Time: 00:00.019, Memory: 8.00 MB

Json Response (Tests\JsonResponse)
 ✓ Ok returns 200
 ✓ Ok body contains payload
 ✓ Ok body is valid json
 ✓ Created returns 201
 ✓ Error returns correct status code
 ✓ Error body contains code and message
 ✓ Error body has error key
 ✓ Headers include content type
 ✓ With cors adds allow origin header

Jwt (Tests\Jwt)
 ✓ Create returns three part token
 ✓ Validate returns payload
 ✓ Validate returns null for wrong secret
 ✓ Validate returns null for malformed token
 ✓ Validate returns null for expired token
 ✓ Token contains exp and iat
 ✓ Create with custom expiry

Router (Tests\Router)
 ✓ Dispatch matches static route
 ✓ Dispatch matches dynamic route
 ✓ Dispatch returns 404 for unknown route
 ✓ Dispatch handles options request
 ✓ Dispatch handles post route
 ✓ Dispatch does not match wrong method
 ✓ Dispatch passes params to handler
 ✓ Multiple routes registered

OK (24 tests, 46 assertions)
```

Рис. 3.24. Виконання тестування

Верифікація клієнтської частини платформи здійснювалася шляхом ручного тестування відображення та функціонування мобільної версії сайту.

Оскільки взаємодія з інтерфейсом користувача на екранах смартфонів має

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				71
Змн.	Арк.	№ докум.	Підпис	Дата		

свою специфіку, основна увага приділялася візуальному контролю адаптивності верстки, коректності відмальовування графічних компонентів та стабільності логіки при роботі на пристроях із різною роздільною здатністю екрана. Процес тестування охоплював симуляцію поведінки покупця за допомогою інструментів розробника в браузері та безпосередню перевірку інтерфейсу на реальних смартфонах для підтвердження гнучкості дизайну під різні моделі мобільних пристроїв [29].

Під час перевірки компонентів відображення каталогу handmade-товарів, карток виробів та кошика покупця було підтверджено, що всі текстові блоки, кнопки та зображення масштабуються коректно. Інтерфейс успішно адаптується до зміни орієнтації екрана з портретної на ландшафтну, при цьому елементи керування залишаються повністю доступними, не перекривають один одного і не виходять за межі видимої області. Також було візуально проконтрольовано роботу динамічних елементів, зокрема коректність зміни кількості товарів у кошику, випадаючих списків вибору кольору та розміру, а також адаптацію форм введення даних.

Перевірка додатку у мобільному форматі наведена на рис. 3.25 та рис. 3.26.



Рис. 3.25. Каталог у мобільному вигляді

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				72
Змн.	Арк.	№ докум.	Підпис	Дата		

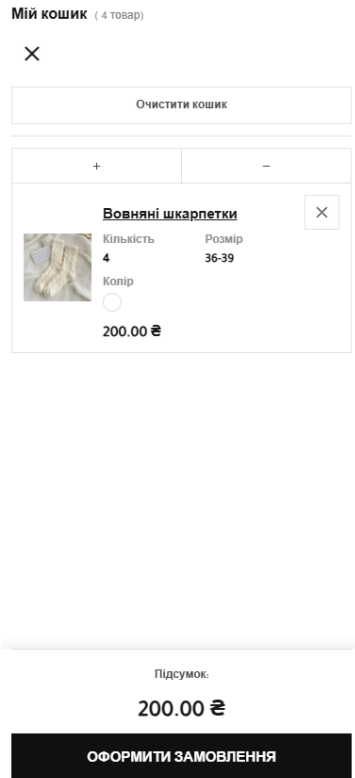


Рис. 3.26. Кошик у мобільному вигляді

Оцінку граничних можливостей та стійкості серверної частини розробленого інтернет-магазину було виконано за допомогою проведення нагрузочного тестування API. Практичним інструментом для симуляції мережевого трафіку виступив програмний комплекс Postman. У процесі дослідження було організовано відправку інтенсивних потоків вхідних запитів до найбільш навантажених ендпоінтів платформи, що дозволило змодельовати поведінку системи у моменти критичного напливу покупців. Основна увага зосереджувалася на моніторингу стабільності ядра PHP-API та швидкості взаємодії із сесіями Redis під впливом аномальної кількості одночасних підключень [30].

Методологія експерименту базувалася на лінійному нарощуванні інтенсивності трафіку, коли кількість запитів за секунду збільшувалася поетапно. Ключовими метриками, які фіксувалися під час тестування, виступали швидкість формування відповідей сервером, тривалість затримок та відсоткове співвідношення успішних операцій до збоїв. Розроблений бекенд успішно впорався із заданим екстремальним навантаженням, не

зафіксувавши жодної помилки авторизації чи втрати пакетів при роботі з базою даних, що підтверджує запас міцності архітектури (рис. 3.27).

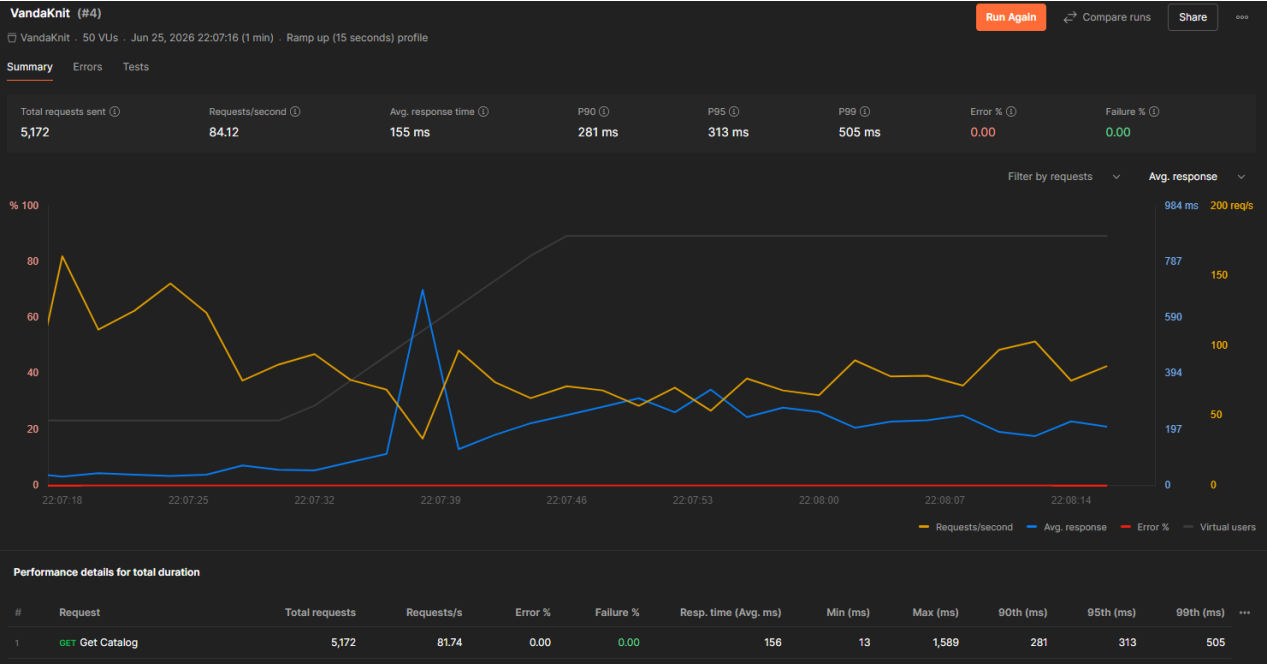


Рис. 3.27. Стрес тестування

Виконаний комплекс діагностичних заходів, що поєднав автоматизовану верифікацію ядра системи, ручний контроль адаптивності інтерфейсу та стрес-тестування під навантаженням, дозволяє підтвердити високу надійність та відмовостійкість розробленого програмного забезпечення. Успішне проходження всіх етапів перевірки гарантує стабільне функціонування інтернет-магазину в умовах інтенсивного користувацького трафіку та забезпечує повну готовність платформи до розгортання в продуктивному середовищі для взаємодії з реальними покупцями. Ретельна всебічна перевірка спроможності розробленого сервісу мінімізує ризики виникнення технічних збоїв під час експлуатації, що безпосередньо вплине на формування позитивної репутації бренду та підвищення лояльності кінцевих користувачів.

Висновки до третього розділу

У третьому розділі представлено практичні аспекти підготовки до роботи, налаштування та експлуатації розробленого вебмагазину, створеного для реалізації handmade-продукції.

Описано процедуру оркестрації та керування інфраструктурними компонентами платформи за допомогою інструменту Docker Compose. Це забезпечило швидку збірку та повну ізоляцію контейнерів серверного API на PHP, локального сховища Redis та додатку React, які успішно взаємодіють із зовнішньою хмарною екосистемою Supabase, де розміщено базу даних PostgreSQL та об'єктне S3-сховище. Впровадження конфігураційних файлів оточення та динамічних змінних середовища дозволило надійно відокремити публічні параметри від чутливих секретних даних, гарантуючи безпеку та передбачуваність функціонування системи на етапі її релізу.

Висвітлено особливості реалізації інтерфейсу користувача з підтримкою серверного рендерингу для десктопних пристроїв та його адаптації під екрани смартфонів. Детально продемонстровано клієнтський шлях покупця, починаючи від вибору параметрів в'язаних виробів у каталозі до оформлення замовлення та здійснення оплати, а також інтерфейси адміністративної панелі для керування замовленнями й контентом.

Стабільність коду та точність виконання бізнес-логіки було доведено за допомогою створення комплексу автоматизованих юніт-тестів у середовищі PHPUnit. Своєю чергою, ручна верифікація відображення мобільної версії сайту підтвердила високу якість адаптивної верстки, гнучкість UI/UX-дизайну та коректність взаємодії з пристроями різної роздільної здатності. Підсумкова оцінка граничних можливостей системи, проведена шляхом інтенсивного стрес-тестування API в інструментарії Postman, продемонструвала відсутність помилок і підтвердила відмовостійкість архітектури під час пікового напливу відвідувачів.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У межах виконання цієї бакалаврської роботи було успішно спроектовано та реалізовано сучасну вебплатформу для продажу в'язаних виробів, що забезпечує швидкий пошук, зручний вибір та безпечне оформлення замовлень клієнтами. На старті дослідження було всебічно вивчено поточний стан ринку електронної комерції у сфері саморобних товарів і проаналізовано функціонал наявних конкурентних рішень. Це дозволило виявити їхні ключові переваги й недоліки, обґрунтувати актуальність створення унікального інтернет-магазину та сформувати вичерпний перелік технічних вимог до майбутнього програмного продукту.

Етап проєктування охоплював детальний аналіз архітектурних паттернів та побудову оптимальної моделі бази даних. Завдяки застосуванню модульного принципу побудови системи вдалося чітко розмежувати бізнес-логіку, ізолювати окремі компоненти та налаштувати гнучкі інтерфейси взаємодії між ними. Такий підхід забезпечив масштабованість вебдодатка, спростив процес його розгортання та мінімізував ризик виникнення критичних помилок при подальшому масштабуванні коду. Практична частина роботи завершилася повноцінною розробкою серверної та клієнтської частин онлайн-магазину.

Стабільність та коректність функціонування вебресурсу було підтверджено за допомогою багаторівневого тестування, яке поєднувало автоматизовану перевірку програмних модулів, симуляцію користувацьких сценаріїв, а також ручне інспектування інтерфейсу.

Результати дипломного проєкту свідчать про повне виконання поставлених завдань та досягнення фінальної мети. Створений інтернет-магазин є ефективним інструментом для популяризації та автоматизації продажів в'язаного одягу, який суттєво покращує досвід онлайн-шопінгу для кінцевого споживача.

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		76

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SEO Learning Center : Ahrefs. URL: <https://ahrefs.com/seo> (дата звернення: 03.06.2026)
2. Responsive web design basics : web.dev. URL: <https://web.dev/articles/responsive-web-design-basics> (дата звернення: 14.06.2026)
3. How do search engines work: BBC. URL: <https://www.bbc.co.uk/bitesize/articles/ztbjq6f> (дата звернення: 22.06.2026)
4. Ecommerce vs Marketplace : VTEX Blog. URL: <https://www.vtex.com/en-us/blog/ecommerce-vs-marketplace/> (дата звернення: 08.06.2026)
5. What is Branding? : WooCommerce. URL: <https://woocommerce.com/posts/what-is-branding/> (дата звернення: 19.06.2026)
6. PHP for Beginners (2023 Edition) : Laracasts. URL: <https://laracasts.com/series/php-for-beginners-2023-edition> (дата звернення: 01.06.2026)
7. Using Composer PHP's Dependency Manager : Medium. URL: <https://zubairidrisaweda.medium.com/using-composer-phps-dependency-manager-73883d7826e> (дата звернення: 11.06.2026)
8. React Getting Started : MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Frameworks_libraries/React_getting_started (дата звернення: 25.06.2026)
9. The Virtual DOM React : LogRocket Blog. URL: <https://blog.logrocket.com/the-virtual-dom-react/> (дата звернення: 05.06.2026)
10. Server Side Rendering Simplified : Medium. URL: <https://medium.com/geekculture/server-side-rendering-simplified-fd708d5520ba> (дата звернення: 17.06.2026)

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Intro to Postgres : Database School. URL: <https://databaseschool.com/series/intro-to-postgres/videos/204> (дата звернення: 12.06.2026)
12. Cybersecurity ACID : VPN Unlimited. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/acid?srsId=AfmBOOpLkEqxK5h_MPL_rsXUpD2IqsrHdyrLwmQSUSv9nCsAwp3fFnR (дата звернення: 02.06.2026)
13. PostgreSQL JSONB Powerful Storage : Architecture Weekly. URL: <https://www.architecture-weekly.com/p/postgresql-jsonb-powerful-storage> (дата звернення: 20.06.2026)
14. What is Redis? The In-Memory Data Store That Makes Your App Faster : dev.to. URL: <https://dev.to/wadethomastt/what-is-redis-the-in-memory-data-store-that-makes-your-app-faster-291p> (дата звернення: 15.06.2026)
15. Storage Technology Explained: What is S3 and what is it good for : Computer Weekly. URL: <https://www.computerweekly.com/feature/Storage-technology-explained-What-is-S3-and-what-is-it-good-for> (дата звернення: 07.06.2026)
16. What is Docker Compose and why do you need it as DevOps Engineer : Medium. URL: <https://medium.com/devops-technical-notes-and-manuals/what-is-docker-compose-and-why-do-you-need-it-as-devops-engineer-adf544d892f7> (дата звернення: 24.06.2026)
17. What is an Admin Panel : Refine Blog. URL: <https://refine.dev/blog/what-is-an-admin-panel/> (дата звернення: 09.06.2026)
18. What's a Payment Gateway and How Does It Work : Montonio Blog. URL: <https://www.montonio.com/blog/whats-a-payment-gateway-and-how-does-it-work> (дата звернення: 18.06.2026)
19. Webhooks Explained: What They Are, How They Work, and Why Developers Love Them : Medium. URL: <https://timkimutai.medium.com/webhooks-explained-what-they-are-how->

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				78
Змн.	Арк.	№ докум.	Підпис	Дата		

- [they-work-and-why-developers-love-them-3ad8f45db14d](#) (дата звернення: 13.06.2026)
20. A Comprehensive Guide to Master SQL Querying : dev.to. URL: <https://dev.to/fullstackdev/a-comprehensive-guide-to-master-sql-querying-12ai> (дата звернення: 21.06.2026)
21. Understanding HTTP Status Codes: A Complete Guide : Trustico Blogs. URL: <https://shop.trustico.com/blogs/stories/understanding-http-status-codes-a-complete-guide?srsId=AfmBOopRmw06VBWjnFwkOcUbql4UT3t-SdJT2RkAxC5RlrhR7-AOfx7> (дата звернення: 04.06.2026)
22. Ubuntu Server Operating System : Hosting.com. URL: <https://hosting.com/blog/ubuntu-server-operating-system/> (дата звернення: 23.06.2026)
23. What is Supabase : MindStudio Blog. URL: <https://www.mindstudio.ai/blog/what-is-supabase> (дата звернення: 10.06.2026)
24. What is JWT Token and How Works in Modern Web Apps : Duende Software. URL: <https://duendesoftware.com/learn/what-is-jwt-token-and-how-works-in-modern-web-apps> (дата звернення: 16.06.2026)
25. What is CORS (Cross-Origin Resource Sharing) : KongHQ. URL: <https://konghq.com/blog/learning-center/what-is-cors-cross-origin-resource-sharing> (дата звернення: 06.06.2026)
26. What Every Developer Should Know About Environment Variables : Medium. URL: <https://medium.com/@rowleks/what-every-developer-should-know-about-environment-variables-91ba07aeb521> (дата звернення: 26.06.2026)
27. Unit Testing : Coursera Articles. URL: <https://www.coursera.org/articles/unit-testing> (дата звернення: 11.06.2026)
28. Unit Testing : Coursera Articles. URL: <https://www.coursera.org/articles/unit-testing> (дата звернення: 25.06.2026)

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				79
Змн.	Арк.	№ докум.	Підпис	Дата		

29. Responsive Design : Kern IT. URL: <https://www.kern-it.be/en/definitions/responsive-design/> (дата звернення: 02.06.2026)
30. Testing an API with Postman : CircleCI Blog. URL: <https://circleci.com/blog/testing-an-api-with-postman/> (дата звернення: 19.06.2026)

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Колмикова А.С.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		81

Лістинг програмного коду

Лістинг А.1:

```

<?php

declare(strict_types=1);

namespace App\Controllers;

use App\Database;
use App\Http\Request;
use App\Http\Response\JsonResponse;
use PDO;
use Throwable;

final class MonobankController
{
    private const MONO_API = 'https://api.monobank.ua';

    private function get_owner_id(Request $request): ?string
    {
        $auth = $request->headers['authorization'] ?? null;
        if ($auth && str_starts_with($auth, 'Bearer ')) {
            $token = substr($auth, 7);
            $secret = $_ENV['JWT_SECRET'] ?? 'default_secret_key_vanda_knit';
            $payload = \App\Jwt::validate($token, $secret);
            if ($payload && isset($payload['userId'])) {
                return (string) $payload['userId'];
            }
        }
        return $request->headers['x-user-id'] ?? null;
    }

    private function mono_token(): string
    {
        $token = $_ENV['MONO_TOKEN'] ?? '';
        if ($token === '') {
            throw new RuntimeException('MONO_TOKEN not configured');
        }
        return $token;
    }

    private function curl(string $method, string $url, ?array $payload = null, array $extra_headers = []): array
    {
        $ch = curl_init($url);
        $headers = array_merge([
            'X-Token: ' . $this->mono_token(),
            'Content-Type: application/json',
        ], $extra_headers);

        curl_setopt_array($ch, [
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_TIMEOUT => 15,
            CURLOPT_CUSTOMREQUEST => strtoupper($method),
            CURLOPT_HTTPHEADER => $headers,
        ]);

        if ($payload !== null) {

```

		Колмикова А.С.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($payload));
    }

    $body = (string) curl_exec($ch);
    $status = (int) curl_getinfo($ch, CURLINFO_HTTP_CODE);
    curl_close($ch);

    return ['status' => $status, 'body' => $body];
}

public function createInvoice(Request $request): JsonResponse
{
    $userId = $this->get_owner_id($request);
    if (!$userId) {
        return JsonResponse::error('unauthorized', 'User identification required.', 401);
    }

    $pdo = Database::connection();

    try {
        $pdo->beginTransaction();

        $sql = <<<SQL
        SELECT
            b.quantity,
            p.id AS "productId",
            p.title,
            p.slug,
            v.id AS "variantId",
            v.size AS "selectedSize",
            v.color AS "selectedColor",
            v.price,
            v.images[1] AS "image"
        FROM basket_items b
        JOIN products p ON b.product_id = p.id
        JOIN product_variants v ON b.variant_id = v.id
        WHERE b.user_id = :user_id
        SQL;

        $stmt = $pdo->prepare($sql);
        $stmt->execute(['user_id' => $userId]);
        $items = $stmt->fetchAll();

        if (empty($items)) {
            $pdo->rollBack();
            return JsonResponse::error('bad_request', 'Basket is empty.', 400);
        }

        $subtotal = 0.0;
        foreach ($items as $item) {
            $subtotal += $item['price'] * $item['quantity'];
        }

        $shipping = $request->body['shipping'] ?? [];

        $shippingFee = 0.0;
        if (isset($shipping['npCity']['value'])) {
            try {
                $npController = new NovaPoshtaController();
                $feeResult = $npController->calculateFee($shipping['npCity']['value'], 1.0, 300.0);
                $shippingFee = (float) ($feeResult['cost'] ?? 0.0);
            } catch (Throwable $e) {
                $shippingFee = 0.0;
            }
        }
    }
}

```

		Колмикова А.С.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

    }
}

$totalAmount = $subtotal + $shippingFee;

$address = isset($shipping['npCity']['label'], $shipping['npWarehouse']['label'])
    ? 'Місто: ' . $shipping['npCity']['label'] . ', Відділення: ' . $shipping['npWarehouse']['label']
    : ($shipping['address'] ?? 'No address');

$insertSql = <<<SQL
INSERT INTO orders (
    user_id, full_name, email, address, mobile,
    is_international, shipping_fee, subtotal, total_amount,
    payment_method, payment_details, status
) VALUES (
    :user_id, :full_name, :email, :address, :mobile,
    :is_international, :shipping_fee, :subtotal, :total_amount,
    :payment_method, :payment_details, :status
) RETURNING id
SQL;

$stmt = $pdo->prepare($insertSql);
$stmt->execute([
    'user_id' => is_numeric($userId) ? (int) $userId : null,
    'full_name' => $shipping['fullname'] ?? 'Guest',
    'email' => 'no-email@vanda-knit.com',
    'address' => $address,
    'mobile' => (string) ($shipping['mobile'] ?? ''),
    'is_international' => 0,
    'shipping_fee' => $shippingFee,
    'subtotal' => $subtotal,
    'total_amount' => $totalAmount,
    'payment_method' => 'monobank',
    'payment_details' => json_encode(['type' => 'monobank']),
    'status' => 'PENDING',
]);

$order = $stmt->fetch();
$orderId = $order['id'];

$itemSql = <<<SQL
INSERT INTO order_items (
    order_id, product_id, variant_id, title, slug,
    selected_size, selected_color, price, quantity, image
) VALUES (
    :order_id, :product_id, :variant_id, :title, :slug,
    :selected_size, :selected_color, :price, :quantity, :image
)
SQL;

$itemStmt = $pdo->prepare($itemSql);
foreach ($items as $item) {
    $itemStmt->execute([
        'order_id' => $orderId,
        'product_id' => $item['productId'],
        'variant_id' => $item['variantId'],
        'title' => $item['title'],
        'slug' => $item['slug'],
        'selected_size' => $item['selectedSize'],
        'selected_color' => $item['selectedColor'],
        'price' => $item['price'],
        'quantity' => $item['quantity'],
        'image' => $item['image'],
    ]);
}

```

		Колмикова А.С.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				84
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    });
}

$pdo->prepare("DELETE FROM basket_items WHERE user_id = :user_id")
->execute(['user_id' => $userId]);

$pdo->commit();

$monoBasket = array_map(static function (array $item): array {
    return [
        'name' => $item['title'] . ' (' . $item['selectedSize'] . ', ' . $item['selectedColor'] . ')',
        'qty' => (int) $item['quantity'],
        'sum' => (int) round($item['price'] * $item['quantity'] * 100),
        'unit' => 'шт.',
        'code' => (string) $item['variantId'],
    ];
}, $items);

$frontendBase = $_ENV['MONO_FRONTEND_URL'] ?? 'http://localhost:5173';
$webhookBase = $_ENV['MONO_WEBHOOK_URL'] ?? 'http://localhost:8080';

$invoicePayload = [
    'amount' => (int) round($totalAmount * 100),
    'ccy' => 980, // UAH
    'redirectUrl' => $frontendBase . '/checkout/success?orderId=' . $orderId,
    'webhookUrl' => $webhookBase . '/mono/webhook',
    'validity' => 3600,
    'paymentType' => 'debit',
    'basket' => $monoBasket,
];

$res = $this->curl('POST', self::MONO_API . '/api/merchant/invoice/create', $invoicePayload);
$data = json_decode($res['body'], true);

if ($res['status'] !== 200 || empty($data['invoiceId'])) {
    $pdo->prepare("UPDATE orders SET payment_details = :pd WHERE id = :id")
    ->execute([
        'pd' => json_encode(['error' => $data, 'raw' => $res['body']]),
        'id' => $orderId,
    ]);

    return JsonResponse::error(
        'mono_error',
        'Monobank invoice creation failed: ' . ($data['errText'] ?? $res['body']),
        502
    );
}

$pdo->prepare("UPDATE orders SET payment_details = :pd WHERE id = :id")
->execute([
    'pd' => json_encode(['invoiceId' => $data['invoiceId'], 'type' => 'monobank']),
    'id' => $orderId,
]);

return JsonResponse::created([
    'orderId' => (string) $orderId,
    'invoiceId' => $data['invoiceId'],
    'pageUrl' => $data['pageUrl'],
]);
} catch (Throwable $e) {
    $pdo->rollBack();
    return JsonResponse::error('server_error', 'Invoice creation failed: ' . $e->getMessage(), 500);
}

```

		Колмикова А.С.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		85

```

    }

    public function webhook(Request $request): JsonResponse
    {
        error_log('Monobank Webhook: Endpoint triggered.');
```

```

        try {
            $data = $request->body;
            if (!is_array($data)) {
                error_log('Monobank Webhook Error: Request body structure invalid.');
```

```

                return JsonResponse::error('bad_request', 'Invalid array format', 400);
            }

            $invoiceId = $data['invoiceId'] ?? null;
            if (!$invoiceId) {
                error_log('Monobank Webhook: Missing invoiceId. Ignoring.');
```

```

                return JsonResponse::ok(['status' => 'ignored']);
            }

            error_log('Monobank Webhook: Fetching true status for invoiceId: ' . $invoiceId);
            $res = $this->curl('GET', self::MONO_API . '/api/merchant/invoice/status?invoiceId=' . $invoiceId);

            if ($res['status'] !== 200) {
                error_log('Monobank Webhook Error: Could not fetch invoice status for ' . $invoiceId);
                return JsonResponse::error('bad_request', 'Could not verify invoice', 400);
            }

            $invoiceData = json_decode($res['body'], true);
            $status = $invoiceData['status'] ?? null;

            error_log(sprintf('Monobank Webhook: Invoice ID [%s], Status from API [%s]', $invoiceId, $status ??
'NULL'));

            if (!$status) {
                return JsonResponse::ok(['status' => 'ignored']);
            }

            error_log('Monobank Webhook: Querying order...');
            $pdo = Database::connection();

            $stmt = $pdo->prepare(
                "SELECT id, status FROM orders WHERE payment_details::jsonb->>'invoiceId' = :invoice_id
LIMIT 1"
            );
            $stmt->execute(['invoice_id' => $invoiceId]);
            $order = $stmt->fetch(PDO::FETCH_ASSOC);

            if (!$order) {
                error_log(sprintf('Monobank Webhook Warning: Order not found for invoiceId: %s', $invoiceId));
                return JsonResponse::ok(['status' => 'order_not_found']);
            }

            error_log(sprintf('Monobank Webhook: Found order ID [%s], current status [%s]', $order['id'],
$order['status']));

            $newStatus = match ($status) {
                'success' => 'PAID',
                'failure' => 'FAILED',
                'reversed' => 'REFUNDED',
                default => null,
            };

            if ($newStatus !== null && $order['status'] !== $newStatus) {

```

		Колмикова А.С.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		86

```

        error_log(sprintf('Monobank Webhook: Altering order %s status to %s', $order['id'], $newStatus));

        $updateStmt = $pdo->prepare(
            "UPDATE orders SET status = :status, payment_details = :pd WHERE id = :id"
        );
        $updateStmt->execute([
            'status' => $newStatus,
            'pd' => json_encode($invoiceData),
            'id' => $order['id'],
        ]);
        error_log(sprintf('Monobank Webhook: Database update committed for order %s.', $order['id']));
    } else {
        error_log('Monobank Webhook: No database modifications necessary.');
    }

    return JsonResponse::ok(['status' => 'ok']);

} catch (Throwable $e) {
    error_log('!!! CRITICAL MONOBANK WEBHOOK EXCEPTION !!!');
    error_log('Message: ' . $e->getMessage());
    error_log('File: ' . $e->getFile() . ' Line: ' . $e->getLine());

    return JsonResponse::error('server_error', 'Internal server error: ' . $e->getMessage(), 500);
}
}
}

```

Лістинг А.2:

<?php

```
declare(strict_types=1);
```

```
namespace App\Controllers;
```

```
use App\Http\Request;
```

```
use App\Http\Response\JsonResponse;
```

```
use Throwable;
```

```
final class NovaPoshtaController
```

```
{
```

```
    private const API_URL = 'https://api.novaposhta.ua/v2.0/json/';
```

```
    private function callApi(string $model, string $method, array $properties = []): array
```

```
{
```

```
        $apiKey = $_ENV['NOVA_POSHTA_API_KEY'] ?? '';
```

```
        if (empty($apiKey)) {
```

```
            throw new RuntimeException('Nova Poshta API Key is not configured in the environment variables.');
```

```
        }
```

```
        $payload = [
```

```
            'apiKey' => $apiKey,
```

```
            'modelName' => $model,
```

```
            'calledMethod' => $method,
```

```
            'methodProperties' => $properties
```

```
        ];
```

```
        $ch = curl_init(self::API_URL);
```

```
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
        curl_setopt($ch, CURLOPT_POST, true);
```

```
        curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($payload));
```

```
        curl_setopt($ch, CURLOPT_HTTPHEADER, ['Content-Type: application/json']);
```

| | | | | | | |
|------|------|----------------|--------|------|---------------------------|------|
| | | Колмикова А.С. | | | ІІЗ.КР.Б – 121 – 26 – ІІЗ | Арк. |
| | | Морозов А.В. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 87 |

```

$response = curl_exec($ch);
$error = curl_error($ch);
curl_close($ch);

if ($error) {
    throw new \RuntimeException('Nova Poshta API curl error: ' . $error);
}

$responseData = json_decode($response, true);

if (!isset($responseData['success']) || $responseData['success'] !== true) {
    $errorMessage = $responseData['errors'][0] ?? 'Unknown NP API Error';
    throw new \RuntimeException($errorMessage);
}

return $responseData['data'];
}

public function searchCities(Request $request): JsonResponse
{
    try {
        $rawSearch = $request->queryParams['search'] ?? $_GET['search'] ?? '';
        $search = trim(urldecode((string) $rawSearch));

        $cities = $this->callApi('Address', 'getCities', [
            'FindByString' => $search,
            'Limit' => 20
        ]);

        $result = array_map(static fn($city) => [
            'ref' => $city['Ref'],
            'name' => $city['Description'] ?? $city['DescriptionUa']
        ], $cities);

        return JsonResponse::ok($result);
    } catch (Throwable $e) {
        return JsonResponse::error('server_error', $e->getMessage(), 500);
    }
}

public function getWarehouses(Request $request): JsonResponse
{
    try {
        $cityRef = $request->queryParams['cityRef'] ?? $_GET['cityRef'] ?? '';

        if (empty($cityRef)) {
            return JsonResponse::error('bad_request', 'City reference (cityRef) parameter is required.', 400);
        }

        $warehouses = $this->callApi('Address', 'getWarehouses', [
            'CityRef' => (string) $cityRef
        ]);

        $result = array_map(static fn($wh) => [
            'ref' => $wh['Ref'],
            'name' => $wh['Description'] ?? $wh['DescriptionUa'],
            'number' => $wh['Number'],
            'type' => $wh['TypeOfWarehouse']
        ], $warehouses);

        return JsonResponse::ok($result);
    } catch (Throwable $e) {
        return JsonResponse::error('server_error', $e->getMessage(), 500);
    }
}

```

| | | | | | | |
|------|------|----------------|--------|------|---------------------------|------|
| | | Колмикова А.С. | | | ІІЗ.КР.Б – 121 – 26 – ІІЗ | Арк. |
| | | Морозов А.В. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 88 |

```

    }
}

public function calculateFee(string $cityRecipient, float $weight = 1.0, float $cost = 300.0, string $serviceType
= 'WarehouseWarehouse'): array
{
    $citySender = $_ENV['NOVA_POSHTA_SENDER_CITY_REF'] ?? '';

    if (empty($citySender)) {
        throw new \RuntimeException('Sender city reference is not configured on the server.');
```

```

    }

    if (empty($cityRecipient)) {
        throw new \InvalidArgumentException('Recipient city reference (cityRecipient) parameter is required.');
```

```

    }

    $pricingData = $this->callApi('InternetDocument', 'getDocumentPrice', [
        'CitySender' => (string) $citySender,
        'CityRecipient' => (string) $cityRecipient,
        'ServiceType' => (string) $serviceType,
        'Weight' => $weight,
        'Cost' => $cost,
    ]);

    $feeEstimation = $pricingData[0] ?? null;

    if (!$feeEstimation) {
        throw new \RuntimeException('Could not parse cost data from Nova Poshta.');
```

```

    }

    return [
        'cost' => $feeEstimation['Cost'] ?? 0,
        'assessedCost' => $feeEstimation['AssessedCost'] ?? 0,
        'deliveryDate' => $feeEstimation['DeliveryDate'] ?? null,
    ];
}

public function getShippingFee(Request $request): JsonResponse
{
    try {
        $cityRecipient = $request->queryParams['cityRecipient'] ?? $_GET['cityRecipient'] ?? '';
        $weight = (float) ($request->queryParams['weight'] ?? $_GET['weight'] ?? 1.0);
        $cost = (float) ($request->queryParams['cost'] ?? $_GET['cost'] ?? 300.0);
        $serviceType = $request->queryParams['serviceType'] ?? $_GET['serviceType'] ??
'WarehouseWarehouse';

        $result = $this->calculateFee($cityRecipient, $weight, $cost, $serviceType);

        return JsonResponse::ok($result);
    } catch (Throwable $e) {
        return JsonResponse::error('server_error', $e->getMessage(), 500);
    }
}
}

```

| | | | | | | |
|------|------|----------------|--------|------|--------------------------|------|
| | | Колмикова А.С. | | | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
| | | Морозов А.В. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 89 |