

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Державне некомерційне підприємство
«Державний університет» Київський авіаційний інститут»**

Факультет комп'ютерних наук та технологій

Кафедра інженерії програмного забезпечення

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____ Олена ГРІНЕНКО

« _____ » _____ 2025 р.

**КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Методика та застосунок багатоплатформенної системи онлайн-магазину

Виконавець: Хоменко Юрій Юрійович

Керівник: к. т. н., доцент, доцент Шибицька Наталія Миколаївна

Нормоконтролер: к. т. н., доцент, доцент Шибицька Наталія Миколаївна

Київ 2025

**Державне некомерційне підприємство
«Державний університет» Київський авіаційний інститут»**

Факультет комп'ютерних наук та технологій
Кафедра інженерії програмного забезпечення
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Олена ГРІНЕНКО

« ____ » _____ 2025 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи студента
Хоменка Юрія Юрійовича

1. Тема кваліфікаційної роботи: «Методика та застосунок багатоплатформенної системи онлайн-магазину»
затверджена наказом ректора № 2450/ст від 17.11.2025 р.
2. Термін виконання проекту: з 29.09.2025 р. по 21.12.2025 р.
3. Вихідні дані до роботи: провести опис методики та застосунку багатоплатформенної системи онлайн-магазину за допомогою середовища об'єктно-орієнтованого програмування WebStorm.
4. Зміст пояснювальної записки:
 1. Теоретичні засади створення системи.
 2. Проектування програмної системи.
 3. Опис реалізації програмної системи.
5. Перелік обов'язкового графічного (ілюстративного) матеріалу:
 1. Структурна схема бази даних багатоплатформенної системи електронної комерції.
 2. Функціональні можливості багатоплатформенної системи електронної комерції.
 3. Інтерфейс веб-додатку та мобільного застосунку.
 4. Діаграма компонентів системи.
 5. Діаграма класів системи.
 6. Демонстрація роботи програми.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Розробка та затвердження графіка роботи	29.09-01.10.2025	виконано
2.	Ознайомлення з постановкою задачі, вивчення інформаційних джерел та складання плану роботи.	02.10-05.10.2025	виконано
2.	Підготовка 1 розділу та подання його керівнику	06.10-19.10.2025	виконано
3.	Підготовка 2 розділу та подання його керівнику	20.10-02.11.2025	виконано
4.	Підготовка 3 розділу та подання його керівнику	03.11-30.11.2025	виконано
6.	Загальне редагування пояснювальної записки, графічного матеріалу. Представлення роботи для перевірки на академічну доброчесність. Проходження нормоконтролю.	01.12-14.12.2025	виконано
7.	Отримання відгуку керівника. Підготовка презентації та тексту доповіді.	14.12-15.12.2025	виконано
8.	Попередній захист (представлення електронної версії пояснювальної записки, презентації, позитивного відгуку керівника).	16.12-17.12.2025	виконано
9.	Рецензування кваліфікаційної роботи	18.12-22.12.2025	виконано
10.	Здача секретарю ЕК пояснювальної записки: електронної версії кваліфікаційної роботи; презентації доповіді; відгуку керівника, рецензії; результату проходження перевірки на плагіат; довідки про успішність, декларації про академічну доброчесність.	23.12-24.12.2025	виконано
11.	Захист кваліфікаційної роботи перед екзаменаційною комісією	26.12.2025	виконано

Дата видачі завдання 29.09.2025 р.

Керівник кваліфікаційної роботи:
к. т. н., доцент, доцент

Наталія ШИБИЦЬКА

Завдання прийняв до виконання:

Юрій ХОМЕНКО

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Методика та застосунок багатоплатформенної системи онлайн-магазину»: 106 сторінок, 50 рисунків, 32 таблиці, 49 використаних джерел.

Об'єкт дослідження – процеси організації та реалізації багатоплатформенних систем електронної комерції.

Мета кваліфікаційної роботи – розробка методики та багатоплатформенного програмного засобу для онлайн-магазину, що забезпечує ефективну взаємодію між продавцем і покупцем, зручний доступ до товарів з різних пристроїв та підвищення конверсії продажів шляхом застосування сучасних технологій веб-розробки та мобільних додатків.

Методи дослідження – аналіз документації, порівняльне дослідження існуючих рішень у сфері електронної комерції, моделювання процесів взаємодії користувача з системою, метод інтерв'ювання потенційних користувачів, системний аналіз бізнес-процесів, прототипування інтерфейсів, А/В-тестування.

Результати роботи можуть бути використані при розробці програмних систем електронної комерції різних масштабів: від малих інтернет-магазинів до великих маркетплейсів; в усіх галузях роздрібною торгівлі, де є потреба в багатоплатформенному доступі до каталогу товарів, автоматизації процесів оформлення замовлень, інтеграції з платіжними системами та службами доставки, управлінні товарними залишками та аналізі поведінки користувачів.

Розробка та дослідження проводилися під управлінням ОС Windows 11. Серверна частина реалізована на фреймворку NestJS з використанням TypeScript. Веб-застосунок розроблено на Next.js з Tailwind CSS. Мобільний застосунок створено за допомогою React Native для iOS та Android платформ.

ОНЛАЙН-МАГАЗИН, ЕЛЕКТРОННА КОМЕРЦІЯ, БАГАТОПЛАТФОРМЕННА СИСТЕМА, WEB-ЗАСТОСУНОК, МОБІЛЬНИЙ ДОДАТОК, NESTJS, NEXT.JS, REACT NATIVE, UI/UX ДИЗАЙН, AGILE/SCRUM, DEVOPS

ABSTRACT

Explanatory note to the qualification work "Methodology and application of a multi-platform online store system": 106 pages, 50 figures, 32 tables, 49 sources used.

The object of the research is the processes of organization and implementation of multi-platform e-commerce systems.

Meta - qualification work is the development of a methodology and a multi-platform software task for an online store that ensures effective interaction between the seller and the buyer, convenient access to goods from different devices and increased sales conversion by using modern web development technologies and mobile applications.

Research methods are documentation analysis, comparative study of existing solutions in the field of e-commerce, modeling of user interaction processes with the system, a method of interviewing your users, system analysis of business processes, interface prototyping, A/B testing.

The results of the work can be used in the development of e-commerce software systems of various scales: from small online stores to large marketplaces; in all retail industries where there is a need for multiple access to the product catalog, automation of order processing processes, integration with payment systems and delivery services, inventory management and analysis of user behavior.

Development and research were carried out under the control of the Windows 11 OS. The server part is implemented on the NestJS framework using TypeScript. The web application is developed on Next.js with Tailwind CSS. The mobile application is created using React Native for iOS and Android platforms.

ONLINE STORE, E-COMMERCE, MULTI-PLATFORM SYSTEM, WEB APPLICATION, MOBILE APPLICATION, NESTJS, NEXT.JS, REACT NATIVE, UI/UX DESIGN, AGILE/SCRUM, DEVOPS

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ СИСТЕМИ.....	9
1.1 Дослідження аудиторії.....	9
1.2 Методика створення ефективного UI/UX.....	11
1.3 Процес вибору платформ для створення систем онлайн-магазинів.....	12
1.4 Безпека користувачів та персональних даних.....	14
1.5 Вибір методології розробки.....	15
1.6 Аналіз ринку електронної комерції.....	18
1.7 Огляд інструментів та засобів для розробки.....	19
Висновок.....	22
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ.....	24
2.1 Моделювання вимог до системи.....	24
2.2 Менеджмент процесу проектування системи.....	33
2.3 Проектування інтерфейсу користувача.....	39
2.4 Проектування структури бази даних.....	44
Висновок.....	63
РОЗДІЛ 3. ОПИС РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ.....	65
3.1 UML-моделювання.....	65
3.2 Діаграма прецедентів.....	67
3.2 Діаграма класів.....	69
3.4 Діаграма пакетів.....	72
3.5 Огляд застосунку.....	73
3.6 Вимоги до апаратного і програмного забезпечення.....	95
3.7 Проектування можливих вдосколанень системи та подальших ітерацій... ..	96
Висновок.....	100
ВИСНОВОК.....	101
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	103

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

API - Application Programming Interface (програмний інтерфейс застосунку)

БД - база даних

CSS - Cascading Style Sheets (каскадні таблиці стилів)

HTML - HyperText Markup Language (мова гіпертекстової розмітки)

HTTP - HyperText Transfer Protocol (протокол передачі гіпертексту)

JSON - JavaScript Object Notation (нотація об'єктів JavaScript)

JWT - JSON Web Token (веб-токен JSON)

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

ПЗ - програмне забезпечення

REST - Representational State Transfer (передача репрезентативного стану)

SDK - Software Development Kit (набір засобів розробки)

SEO - Search Engine Optimization (пошукова оптимізація)

SQL - Structured Query Language (мова структурованих запитів)

SSR - Server-Side Rendering (серверний рендеринг)

СУБД - система управління базами даних

UI - User Interface (користувацький інтерфейс)

UX - User Experience (користувацький досвід)

ВСТУП

Актуальність теми. У сучасному світі великі компанії дедалі більше зосереджують увагу на методиках розробки програмного забезпечення [1][2], адже саме вони є основою створення конкурентних та ефективних цифрових продуктів. Одним із напрямів, що стрімко розвивається, є електронна комерція, яка вимагає створення багатоплатформених систем, доступних як з комп'ютерів, так і з мобільних пристроїв. Онлайн-магазини стали важливим інструментом для розвитку бізнесу, оскільки забезпечують постійний доступ покупців до товарів і послуг. Дослідження розкриває, чому методи розробки програмних продуктів є вирішальними для створення сучасних онлайн-магазинів та як багатоплатформений підхід може забезпечити зручність користування, масштабованість та відповідність вимогам замовника. Актуальність роботи також полягає у важливості організації якісної взаємодії між покупцем і продавцем, забезпеченні ефективної комунікації та доступу до публічної інформації про товари й акції. Дослідження охоплює аналіз проблем доступності електронної комерції для різних категорій користувачів та пошук рішень для підвищення рівня інформованості споживачів.

Об'єкт дослідження. Забезпечення можливості організації багатоплатформенної системи електронної комерції.

Предмет дослідження. Структура та реалізація багатоплатформенного застосунку для онлайн-магазину.

Мета проєкту. Розробка багатоплатформенного програмного засобу для онлайн-магазину. Дослідження методик і варіантів розробки програмного забезпечення, вибір оптимальної архітектури та технологій для створення системи електронної комерції, яка відповідає сучасним вимогам. Додатковим завданням є підвищення ефективності бізнес-процесів за рахунок спрощення взаємодії між продавцем і покупцем та забезпечення зручного доступу до інформації про товари й послуги.

Завдання дослідження: проаналізувати термінологічний апарат

дослідження; визначити специфіку, вимоги та характерні особливості багатоплатформенних систем; провести дослідження існуючих рішень у сфері онлайн-торгівлі; визначити методи та засоби для розроблення онлайн-магазину; спроектувати структуру багатоплатформенного застосунку; розробити концепцію та архітектуру; здійснити програмну реалізацію системи з урахуванням вимог користувачів.

Методи дослідження. Використано аналіз документації, порівняльне дослідження існуючих рішень, а також метод інтерв'ювання потенційних користувачів для уточнення вимог.

Технічні та програмні засоби. У дипломному проєкті описано процес створення системи із застосуванням сучасних фреймворків і середовищ розробки, що забезпечують кросплатформеність та інтеграцію з базами даних.

Елементи наукової новизни. Розробка конкретного багатоплатформенної системи з дотриманням загальновідомих процесів розробки ПЗ та інтеграцією сучасних методів електронної комерції.

Практична значимість проєкту. Застосунок являє собою сервіс, що надає користувачу можливість у зручному середовищі здійснювати покупки незалежно від типу пристрою. Це спрощує взаємодію між продавцем і покупцем та підвищує ефективність бізнесу.

Особистий внесок. Розробка концепції, архітектури та структури застосунку, класифікації функціональних можливостей, дизайн користувацького інтерфейсу, програмна реалізація, проєктування бази даних.

Апробація результатів. Система представлена у вигляді багатоплатформенного застосунку зі зручним інтерфейсом та широкою функціональністю, що надає можливість покупцю швидко знаходити та замовляти потрібні товари.

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ СИСТЕМИ

1.1 Дослідження аудиторії

Дослідження аудиторії є ключовим етапом під час створення багатоплатформених систем електронної комерції, оскільки воно дає змогу визначити потреби, поведінкові шаблони і очікування покупців. Одним із фундаментальних методів являється опитування та анкетування, що дають змогу отримати кількісні дані щодо демографічних характеристик, купівельних звичок та рівня задоволеності користувачів. Використання стандартизованих анкет дозволяє систематизувати інформацію та виявити закономірності, які впливають на користувацький досвід.

Формування портрету ідеального покупця - це один із ключових методів дослідження аудиторії, який дозволяє більш глибоко зрозуміти цільових користувачів та адаптувати функціонал і дизайн магазину під їхні потреби. Даний процес ґрунтується на методах сегментації ринку, що включає аналіз демографічних (вік, стать, освіта), географічних (місце проживання) та поведінкових характеристик (частота покупок, середній чек, лояльність).

Інший важливий метод - інтерв'ю та фокус-групи, які надають якісні дані, наприклад, ставлення до інтерфейсу, очікування щодо функціональності та труднощі у взаємодії з системою. Такі підходи надають змогу виявити неочевидні проблеми використання.

Також активно застосовується аналіз поведінкових даних (behavioral analytics), який включає вивчення натискань, часу перебування на сторінці, конверсійних шляхів і карт теплових зон (heatmaps). Даний метод базується на зборі даних у реальному часі й дозволяє оцінити ефективність окремих елементів інтерфейсу та оптимізувати їх для підвищення зручності

Окрім цього, достатньо обґрунтованим є метод А/В-тестування, який передбачає порівняння двох чи більше версій інтерфейсу або функціоналу з метою визначення, яка з них отримує кращі результати (конверсія, час

завершення покупки, кількість повторних відвідувань). На рис. 1.1 зображено наведені основні методи дослідження аудиторії:

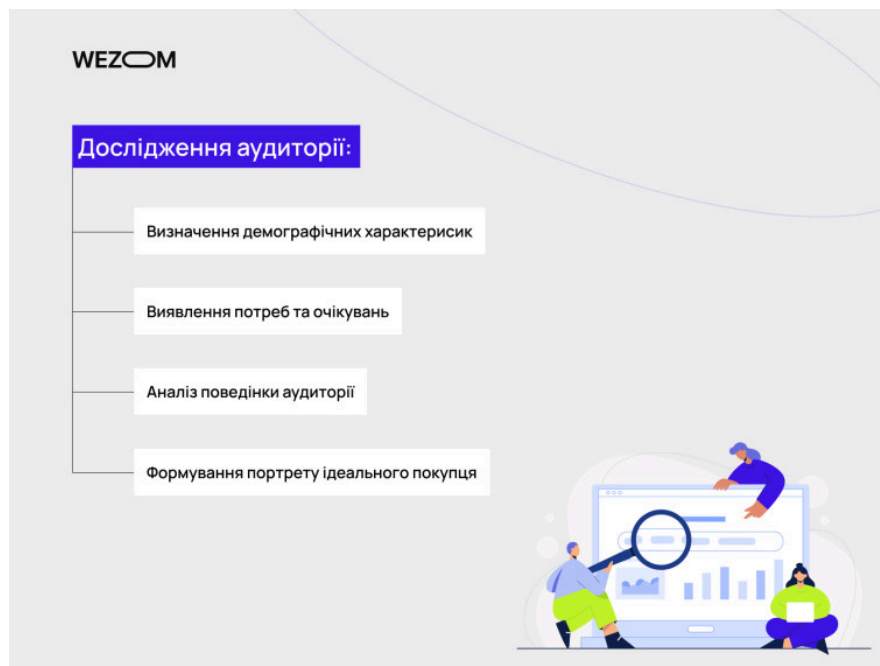


Рис. 1.1. Методи дослідження аудиторії

1.2 Методика створення ефективного UI/UX

Створення користувацького інтерфейсу (UI) для багатоплатформенних систем онлайн-магазинів ґрунтується на забезпеченні зручності використання (UX), ефективності та безпеки. Важливим аспектом є інтуїтивно зрозумілий та адаптивний дизайн [5], що відповідає вимогам різних платформ. Стандартизована навігація, чіткі картки товарів та оптимізований процес оформлення замовлення підвищують конверсію та загальну задоволеність користувачів.

Зазвичай, для створення ефективних та зручних інтерфейсів дотримуються наступних основних законів (рис. 1.2):

1. Закон Якоба (Jakob's Law) стверджує, що користувачі проводять більшість часу на інших вебсайтах і програмних платформах, тому очікують, що ваш інтерфейс працюватиме подібно до знайомих їм систем. Це означає, що дизайн має базуватися на загальноприйнятих патернах та стандартах, щоб

зменшити когнітивне навантаження і полегшити навчання користувачів.

2. Закон Фітта (Fitts's Law) описує залежність часу, необхідного для переміщення курсора до цільового елемента, від відстані до нього та його розміру. Чим ближче і більший елемент, тим швидше його можна активувати. Закон широко використовується при проектуванні кнопок, меню та інтерактивних елементів інтерфейсу для підвищення швидкості та точності взаємодії.

3. Закон Хікса (Hick's Law) стверджує, що час прийняття рішення користувачем збільшується логарифмічно з кількістю варіантів вибору. Тому інтерфейси з великою кількістю опцій потребують групування та ієрархічного подання інформації, щоб зменшити когнітивне навантаження та прискорити вибір.

4. Закон Міллера (Miller's Law) говорить про те, що середньо-статистична людина може одночасно утримувати в пам'яті приблизно 7 (+2) елементів. Це важливо враховувати при проектуванні меню, форм і списків: надлишкова інформація повинна бути подрібнена на блоки або етапи для полегшення сприйняття.



Рис. 1.2. Основні закони створення інтерфейсів

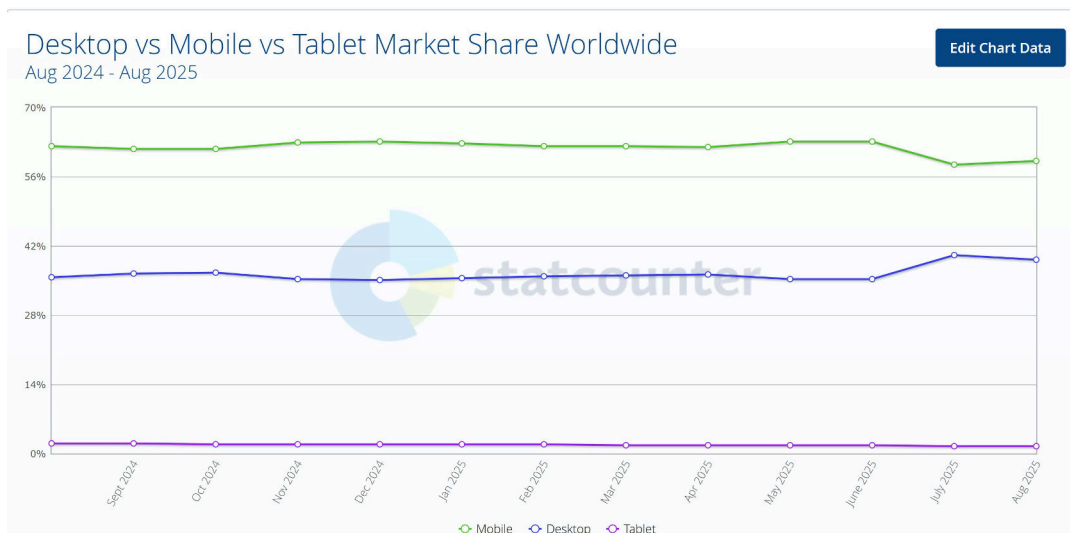
1.3 Процес вибору платформ для створення систем онлайн-магазинів

Вибір технологій для багатоплатформених систем онлайн-магазинів є критичним. Дослідження показують, що комбінація нативної та багатоплатформенної розробки може бути ефективною для проєктів середньої та великої складності.

Онлайн-магазини доцільно створювати у багатоплатформенному форматі, щоб мати можливість охопити якомога ширшу аудиторію та забезпечити зручний доступ із різних пристроїв. Для цього варто реалізувати адаптивний вебсайт, який є універсальною точкою доступу через браузер, мобільні застосунки для Android та iOS, що забезпечать кращу продуктивність і нативний користувацький досвід, а також десктопні додатки для Windows і macOS, орієнтовані на користувачів, які віддають перевагу роботі з ПК (рис 1.3).



Рис. 1.3. Платформи, що мають підтримувати систему



Global Ecommerce Sales by Device (2024)

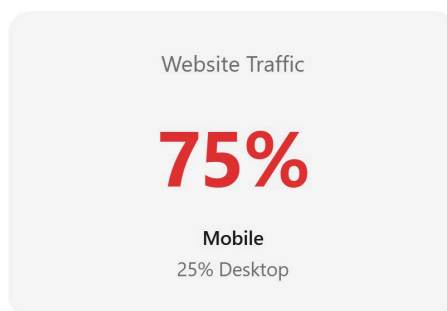
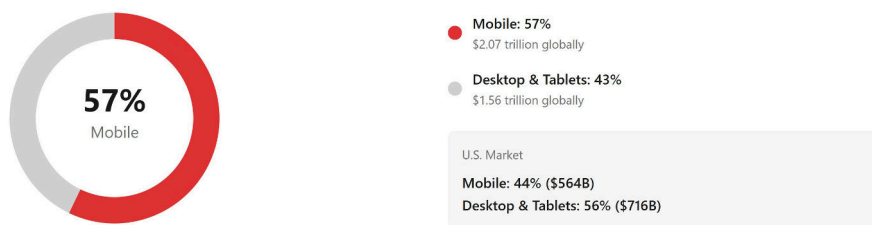


Рис. 1.4. - 1.6. Статистика використання платформ у електронній комерції

Як видно із рис. 1.4 - 1.6, статистика використання платформ у сфері електронної комерції вказує на те, що:

- За даними StatCounter, $\approx 63.3\%$ глобального веб-трафіку наразі йде з мобільних пристроїв, тоді як на десктопи припадає близько 36.7% .
- У сфері електронної комерції приблизно 57% світових продажів у 2024 році здійснюються через мобільні пристрої.
- Близько 75% відвідувань сайтів ритейлерів надходить із смартфонів.

1.4 Безпека користувачів та персональних даних

Безпека користувачів та аутентифікація також відіграють ключову роль. Впровадження системи єдиного входу (SSO) із використанням протоколу OpenID Connect (OIDC) дозволяє спростити процес аутентифікації, знизити ризики та підвищити зручність використання сервісів.

Для підвищення ефективності пошуку та навігації користувачів у системі застосовують фасетний пошук та системи рекомендацій. Такі інтерфейси допомагають швидко знаходити необхідні товари, що позитивно впливає на задоволеність користувачів та ефективність взаємодії з онлайн-магазином.

Завершальним етапом є інтеграція зручного UI з надійними заходами безпеки, такими як аутентифікація через сторонні сервіси (наприклад, Google) та захищені платіжні шлюзи. Це забезпечує безпечний та ефективний онлайн-шопінг, захищає дані користувачів та підвищує рівень довіри до системи.

1.5 Вибір методології розробки

Для ефективної розробки системи доцільно застосувати Agile/Scrum у поєднанні з DevOps. Agile забезпечує гнучкість реакції на зміну бізнес-вимог та швидкий випуск мінімально життєздатного продукту (MVP). DevOps дозволяє автоматизувати процеси інтеграції та доставки (CI/CD), що особливо важливо для сфери електронної комерції, де швидкість виходу нових функцій прямо впливає на прибутки та задоволеність клієнтів. Паралельне застосування Agile та DevOps забезпечує безперервну взаємодію команд розробки та експлуатації, зменшує час на виправлення помилок і підвищує стабільність продукту.

Agile - це гнучка методологія розробки програмного забезпечення, яка акцентує увагу на коротких ітераціях, тісній взаємодії з замовником і швидкому реагуванні на зміни у вимогах.

Scrum пропонує структуру роботи через спринти (2-4 тижні), що дозволяє ефективно планувати, відстежувати і коригувати задачі (рис. 1.7).

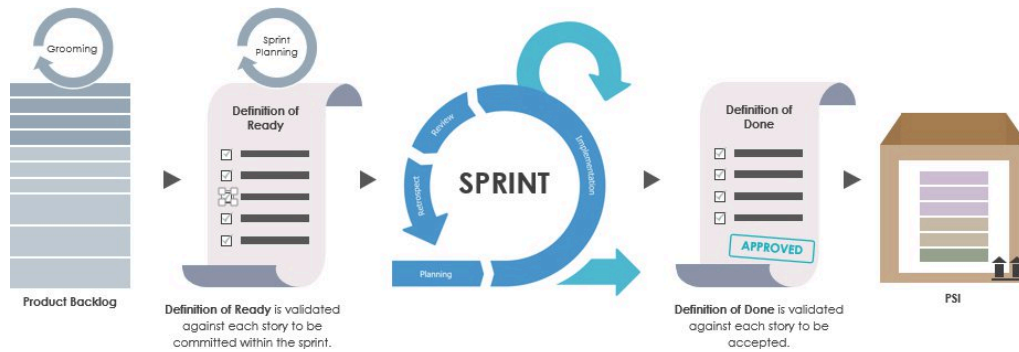


Рис. 1.7. Спринт в методології Scrum

У рамках Scrum основним інструментом планування і управління завданнями є беклог продукту (Product Backlog) [35], який можна побачити на рис. 1.8. Це впорядкований список всіх вимог, функцій, покращень та виправлень, які повинні бути реалізовані в системі.

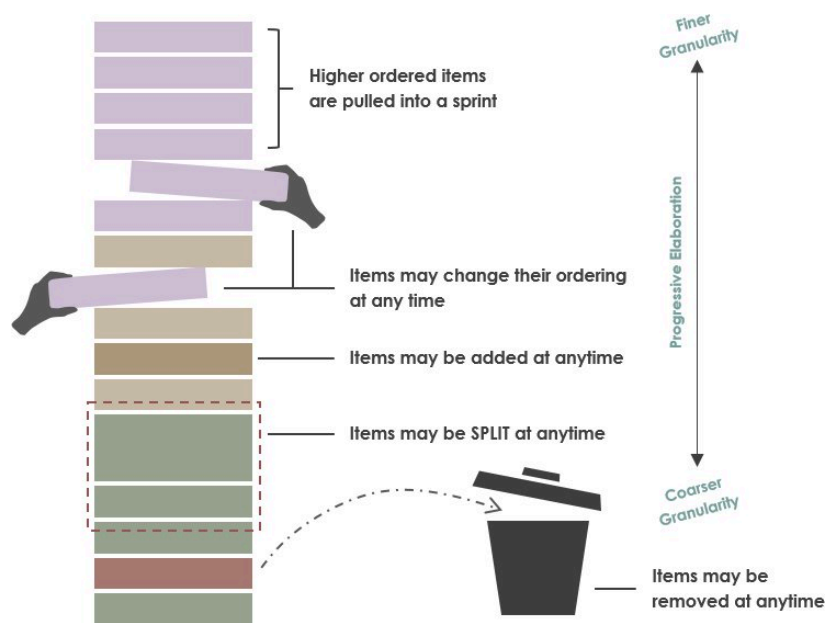


Рис. 1.8. Беклог задач

Scrum Board - це візуальний інструмент, який допомагає команді відстежувати виконання задач із беклогу під час спринту (рис 1.9). Зазвичай дошка поділена на колонки, що відображають стан завдань: To Do, In Progress, Testing/Review, Done. Під час розробки багатоплатформенного онлайн-магазину дана дошка може використовуватися для координації роботи над різними модулями (catalog, cart, order, payment, delivery) та фронтенд-клієнтами (web,

mobile). Це дозволяє команді швидко реагувати на зміни пріоритетів і ефективно управляти спринтами.

Structure of a Scrum board

Product Backlog	User Story	Work to Do	Work in Progress	On Hold	Work Done	Work in Review

Рис. 1.9. Структура дошки Scrum

DevOps (від англ. Development + Operations) - це підхід до розробки та експлуатації програмного забезпечення, що інтегрує процеси розробки (Development) і підтримки (Operations) з метою підвищення швидкості, надійності та ефективності створення і доставки ПЗ. Це практика, яка поєднує методи безперервної інтеграції та доставки (CI/CD) (рис 1.10), автоматизації інфраструктури, управління конфігураціями та моніторингу для оптимізації життєвого циклу програмних продуктів.

DevOps активно використовує технології контейнеризації, такі як Docker [19], для стандартизації середовища виконання програм і забезпечення переносимості додатків між різними системами (рис. 1.11). Docker дозволяє ізолювати програму разом із всіма її залежностями в контейнер [20], що спрощує розгортання та тестування [3], роблячи процеси більш передбачуваними та відтворюваними. На основі таких контейнерів у DevOps налаштовуються пайплайни безперервної інтеграції та доставки (CI/CD), які автоматизують збірку, тестування і розгортання програмного забезпечення, забезпечуючи швидку і надійну доставку нових версій продукту.

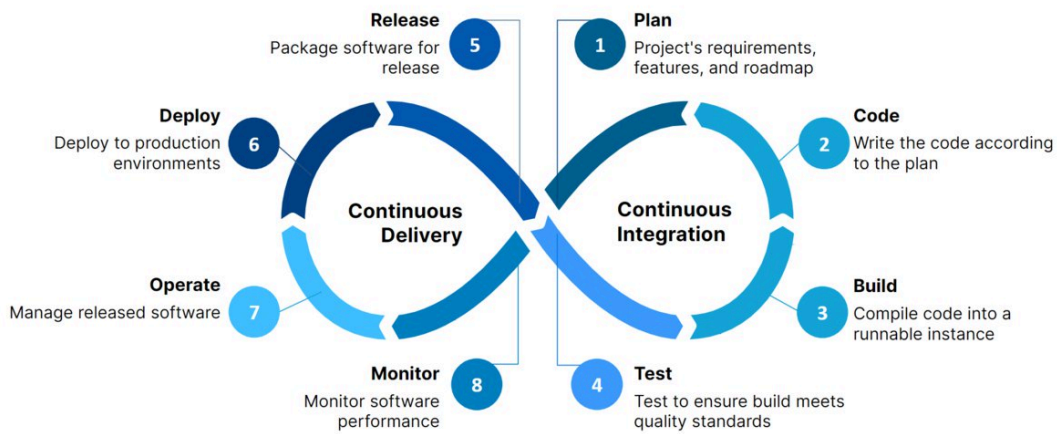


Рис. 1.10. CI/CD

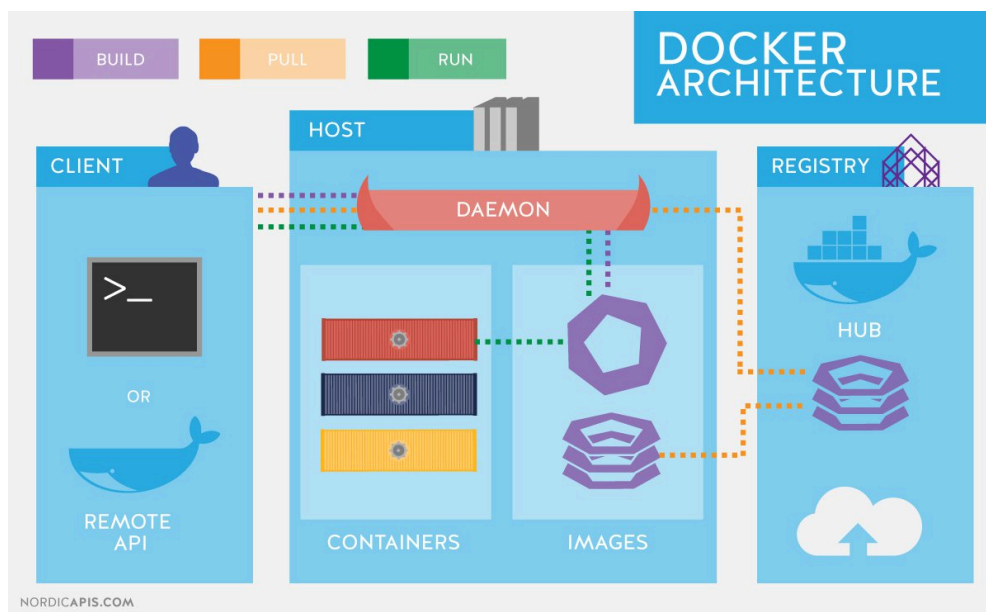


Рис. 1.11. Архітектура Docker

1.6 Аналіз ринку електронної комерції

Amazon історично розвивав розподілені сервіси з фокусом на low-latency, high-availability, automation CI/CD і масовий fulfilment (WMS/OMS).

Zalando - класичний приклад event-driven платформи: продукт як “event stream”, materialized views для сервінгу, принципи microservices & API-first. Це дає змогу масштабувати командно-орієнтовані домени й знижувати coupling.

ASOS багато публікували про міграцію від on-prem моноліту до cloud і мікросервісів [24], одночасно інвестуючи в mobile apps і PWA для кращої конверсії на мобільних пристроях.

IKEA вирізняється складністю товарних конфігурацій (меблів), hybrid commerce (фізичний магазин + онлайн-магазин), click-and-collect і AR-додатком для візуалізації товару в домі - це потребує сильної синхронізації інвентару й WMS/OMS інтеграцій.

Altudo Sephora активно застосовує персоналізацію, AI/AR для beauty-визначення, loyalty інтеграцію і сильний mobile/UI досвід, є прикладом «retail + tech» для підвищення конверсії.

Etsy веде інтенсивну роботу над рекомендаційними системами і поєднує практики підтримки моноліту там, де це ефективніше, і мікросервісів там [24], де потрібна ізоляція.

Walmart/Target - приклад омніканалу (ship-from-store, pickup, grocery), із сильною інтеграцією POS і fulfilment; на практиці застосовують distributed microservices, event-driven інтеграції та продвинуті DR/BC практики.

Rozetka - приклад українського великого ритейлу з мільйонними відвідинами й великою SKU-базою, специфічними логістичними партнерствами та локальними платіжними інтеграціями.

1.7 Огляд інструментів та засобів для розробки

Реалізація системи електронної комерції FullMag базується на сучасному стеку технологій, що забезпечує високу продуктивність, масштабованість та зручність розробки.

Архітектура проекту.

Проект організовано як монорепозиторій - єдине сховище коду, що містить усі компоненти системи: веб-застосунок, мобільний застосунок, серверну частину та спільні бібліотеки. Для управління залежностями використовується rnpm - сучасний менеджер пакетів для Node.js з вбудованою підтримкою монорепозиторіїв. Основною мовою програмування є TypeScript [12], що забезпечує статичну типізацію та раннє виявлення помилок на етапі компіляції.

Серверна частина.

Backend побудовано на фреймворку NestJS - прогресивному Node.js-

фреймворку [11] з модульною архітектурою, ін'єкцією залежностей та декларативним підходом. Як основну СУБД використано PostgreSQL з ORM-бібліотекою TypeORM для роботи з сутностями бази даних. Для кешування та черг завдань застосовується Redis разом із бібліотекою Bull. Повнотекстовий пошук товарів реалізовано через Elasticsearch. Комунікація в реальному часі забезпечується Socket.io, а система автентифікації побудована на Passport.js з JWT-токенами.

Веб-застосунок.

Frontend реалізовано на Next.js - React-фреймворку з підтримкою серверного рендерингу (SSR) для SEO-оптимізації [5]. Для стилізації використовується Tailwind CSS, управління станом - Zustand, робота з формами - React Hook Form у поєднанні з Zod для валідації.

Мобільний застосунок.

Кросплатформний мобільний застосунок розроблено на React Native з єдиною кодовою базою для iOS та Android. Навігація реалізована через React Navigation [5], управління станом - Redux Toolkit з Redux-Persist для збереження даних між сесіями.

Інтеграції та інфраструктура.

Для обробки платежів інтегровано Stripe [18], email-розсилки реалізовано через Nodemailer з шаблонізатором Handlebars. Сповіднення адміністраторів надходять через Telegram Bot API, Viber Bot API та Whatsapp Business Api. Моніторинг помилок здійснюється через Sentry. Контейнеризація сервісів реалізована за допомогою Docker та Docker Compose [20].

Таблиця 1.1

Основні технології системи FullMag

Категорія	Технологія	Версія	Призначення
Мова	TypeScript	5.3.3	Основна мова розробки
Runtime	Node.js	>=20.0.0	Середовище виконання
Package Manager	pnpm	10.23.0	Управління залежностями
Backend Framework	NestJS	10.3.0	Серверний фреймворк
Web Framework	Next.js	14.1.0	Веб-застосунок

Mobile Framework	React Native	0.82.1	Мобільний застосунок
UI Library	React	18.2.0 / 19.1.1	Користувацький інтерфейс
Database	PostgreSQL	16	Реляційна БД
ORM	TypeORM	0.3.19	Object-Relational Mapping
Cache	Redis	7	Кешування
Search	Elasticsearch	8.11.0	Повнотекстовий пошук
Queue	Bull	4.12.0	Черги завдань
Real-time	Socket.io	4.6.1	WebSocket комунікація
Auth	Passport + JWT	0.7.0	Автентифікація
Payments	Stripe	20.0.0	Обробка платежів
Email	Nodemailer	6.9.8	Email-розсилки
Monitoring	Sentry	10.26.0	Моніторинг помилок
CSS	Tailwind CSS	3.4.1	Стилізація (web)
State (Web)	Zustand	4.5.0	Управління станом
State (Mobile)	Redux Toolkit	2.10.1	Управління станом
Testing	Jest	29.x	Тестування
Linting	ESLint	8.56.0	Аналіз коду
Formatting	Prettier	3.2.4	Форматування коду
Containerization	Docker	-	Контейнеризація
Orchestration	Docker Compose	3.8	Оркестрація контейнерів
Reverse Proxy	Nginx	alpine	Проксі-сервер

На рис. 1.12 можна побачити візуальну схему технологій створюваної системи:



Рисунок 1.12 - Схема технологій системи

Висновок

У першому розділі було проведено комплексний теоретичний аналіз засад створення багатоплатформених систем електронної комерції. Досліджено основні методи вивчення аудиторії, включаючи опитування, інтерв'ю, поведінкову аналітику та A/B-тестування, що дозволяє створювати користувацькі інтерфейси, максимально адаптовані до потреб цільових користувачів.

Розглянуто ключові закони та принципи проєктування ефективних UI/UX інтерфейсів (закони Якоба, Фітта, Хікса та Міллера), дотримання яких

забезпечує інтуїтивно зрозумілий та зручний досвід користування системою. Обґрунтовано необхідність підтримки множинних платформ (веб, мобільні iOS та Android) для охоплення максимальної аудиторії, що підтверджується статистикою переважання мобільного трафіку в електронній комерції. Проаналізовано питання безпеки користувачів та персональних даних, включаючи протоколи аутентифікації та захищені платіжні механізми. Обґрунтовано вибір методології розробки Agile/Scrum у поєднанні з DevOps-практиками та контейнеризацією Docker для забезпечення гнучкості та автоматизації процесів [19] [20].

На основі аналізу провідних систем електронної комерції (Amazon, Zalando, ASOS, IKEA, Rozetka та інших) визначено кращі практики побудови архітектури та функціональності. Обрано сучасний технологічний стек: NestJS для серверної частини, Next.js для веб-застосунку, React Native для мобільних платформ [8] [9], PostgreSQL та Redis для зберігання даних.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

2.1 Моделювання вимог до системи

Система має вдовольняти наступні функціональні вимоги:

- Система повинна дозволяти користувачам реєструватися за допомогою email та пароля
- Система повинна підтримувати автентифікацію через JWT токени
- Система повинна надавати можливість відновлення паролю
- Токени оновлення (refresh tokens) повинні автоматично оновлювати access tokens
- Система повинна підтримувати вихід з усіх пристроїв
- Система повинна підтримувати три ролі: admin, manager, customer
- Адміністратори мають повний доступ до всіх функцій
- Менеджери мають доступ до управління замовленнями, продуктами, клієнтами
- Клієнти мають доступ до покупок та перегляду своїх замовлень
- Користувачі можуть переглядати та редагувати свій профіль
- Користувачі можуть зберігати декілька адрес доставки
- Користувачі можуть переглядати історію своїх замовлень
- Користувачі можуть управляти списком бажань (wishlist)
- Система повинна підтримувати ієрархічну структуру категорій (категорії та підкатегорії)
- Категорії можуть мати необмежену глибину вкладеності
- Адміністратор може створювати, редагувати та видаляти категорії
- Неможливо видалити категорію, яка містить товари або підкатегорії
- Система повинна запобігати циклічним посиланням при налаштуванні батьківської категорії
- Система повинна підтримувати різні типи атрибутів: текст, число, boolean, select, multi-select, діапазон

- Атрибути можуть бути глобальними або прив'язаними до конкретної категорії
- Атрибути можуть бути позначені як фільтруємі, обов'язкові, видимі
- Атрибути можуть мати одиниці вимірювання (GB, inches, MP тощо)
- Адміністратор може визначати порядок відображення атрибутів
- Система повинна підтримувати створення, редагування та видалення продуктів
- Кожен продукт має: назву, опис, ціну, валюту, SKU, кількість на складі
- Продукти можуть мати декілька зображень
- Продукти прив'язуються до категорій
- Продукти мають динамічні атрибути згідно з їх категорією
- Система відстежує кількість товару на складі
- Система підтримує знижки на товари
- Система повинна підтримувати імпорт товарів з CSV файлів
- Адміністратор може створювати профілі імпорту з налаштуваннями: відображення колонок CSV на поля продукту, відображення атрибутів, правила валідації, правила трансформації даних
- Система повинна показувати попередній перегляд імпорту з статистикою
- Система повинна зберігати історію імпортів з детальними звітами
- Користувачі можуть шукати товари за назвою, описом, SKU
- Пошук повинен працювати в режимі реального часу
- Пошук повинен підтримувати часткове співпадіння
- Користувачі можуть фільтрувати товари за категоріями
- Користувачі можуть фільтрувати за ціновим діапазоном
- Користувачі можуть фільтрувати за наявністю на складі
- Користувачі можуть фільтрувати за динамічними атрибутами товарів
- Фільтри повинні зберігатися в URL для можливості поділитися посиланням
- Система повинна показувати кількість товарів для кожного фільтру

- Користувачі можуть сортувати товари за: ціною, назвою, датою додавання, популярністю
- Сортування повинно зберігатися в URL
- Користувачі можуть перемикає між сіткою та списком
- Система підтримує пагінацію результатів
- Користувачі можуть вибирати кількість товарів на сторінці
- Авторизовані користувачі мають персональний кошик, що зберігається в БД
- Неавторизовані користувачі мають кошик в localStorage
- Користувачі можуть додавати товари в кошик
- Користувачі можуть змінювати кількість товарів в кошику
- Користувачі можуть видаляти товари з кошика
- Система показує загальну вартість кошика
- Система перевіряє наявність товару перед додаванням до кошика
- Користувачі можуть оформити замовлення з товарів в кошику
- Система інтегрована з API Нової Пошти для вибору міста та відділення
- Користувачі вводять дані отримувача (ім'я, телефон)
- Користувачі вибирають спосіб оплати: онлайн оплата (Stripe) [18], готівка при отриманні
- Після успішного створення замовлення кошик очищається
- Система інтегрована з Stripe для онлайн оплати
- Користувачі можуть переглядати список своїх замовлень
- Користувачі можуть переглядати деталі кожного замовлення
- Адміністратори можуть переглядати всі замовлення
- Адміністратори можуть змінювати статус замовлення: PENDING → PROCESSING → PAID → SHIPPED → DELIVERED, CANCELLED
- Система генерує інвойси для замовлень
- Користувачі можуть завантажити інвойс в PDF форматі
- Користувачі можуть створити запит на повернення для доставленого замовлення

- Користувачі вказують причину повернення та опис
- Користувачі можуть додати фото товару
- Адміністратори бачать список запитів на повернення
- Адміністратори можуть схвалити або відхилити повернення
- При схваленні повернення товар повертається на склад
- Система відстежує статуси повернень: PENDING, APPROVED, REJECTED, REFUNDED
- Користувачі можуть залишати відгуки на куплені товари
- Відгук містить: рейтинг (1-5 зірок), текст, можливість додати фото
- Користувачі можуть редагувати свої відгуки
- Відгуки можна позначати як корисні (лайки)
- Адміністратори можуть відповідати на відгуки
- Користувачі можуть скаржитися на неприйнятні відгуки
- Адміністратори переглядають скарги на відгуки
- Адміністратори можуть видалити або приховати відгук
- Користувач отримує повідомлення про статус скарги
- Користувачі можуть додавати товари до списку бажань
- Користувачі можуть видаляти товари зі списку бажань
- Користувачі можуть переглядати свій список бажань
- З списку бажань можна швидко додати товар в кошик
- Користувачі можуть додавати товари до порівняння
- Порівняння доступне лише для товарів однієї категорії
- Система показує атрибути товарів в таблиці порівняння
- Користувачі можуть видаляти товари з порівняння
- Адміністратор може встановлювати знижки на товари
- Знижки мають відсоток та період дії
- Система автоматично застосовує активні знижки
- Адміністратор може створювати купони з кодом
- Купони мають тип: відсоток, фіксована сума, безкоштовна доставка

- Купони мають обмеження: мінімальна сума, дата початку/закінчення, кількість використань
- Користувачі вводять код купона при оформленні замовлення
- Система відстежує використання купонів
- Система автоматично виявляє кошики, не оформлені більше 24 годин
- Адміністратори бачать список покинутих кошиків
- Система показує загальну вартість покинутих кошиків
- Система автоматично надсилає email нагадування через 24 години
- Email містить список товарів та посилання на кошик
- Адміністратор може налаштувати шаблон email
- Користувачі можуть відкрити чат з підтримкою
- Чат працює в реальному часі через WebSocket
- Менеджери бачать список активних чатів
- Менеджери можуть відповідати на повідомлення
- Система зберігає історію повідомлень
- Користувачі можуть зв'язати Telegram акаунт
- Telegram бот надсилає повідомлення про статус замовлення
- Користувачі можуть отримувати відповіді через бота
- Користувачі можуть зв'язати WhatsApp акаунт
- WhatsApp бот надсилає повідомлення про статус замовлення
- Користувачі можуть ставити запитання через WhatsApp
- Система надсилає email при реєстрації
- Система надсилає email при створенні замовлення
- Система надсилає email при зміні статусу замовлення
- Система надсилає email при успішній оплаті
- Email шаблони редагуються через Handlebars
- Адміністратор може створювати банери для головної сторінки
- Банери мають: зображення, заголовок, опис, посилання, порядок відображення
- Банери можуть бути активними/неактивними

- Система відстежує покази банерів
- Система показує загальну статистику продажів
- Графіки продажів за період (день, тиждень, місяць)
- Топ товарів за продажами
- Розподіл замовлень за статусами
- Статистика доходів за категоріями
- Статистика по кожному товару (перегляди, покупки, конверсія)
- Товари з низьким залишком
- Розподіл товарів за ціновими діапазонами
- Загальна вартість інвентаря
- Кількість нових користувачів за період
- Активні користувачі
- Середній чек замовлення
- Lifetime Value (LTV) клієнтів
- Нефункціональні вимоги:
- Час завантаження головної сторінки не повинен перевищувати 2 секунди
- Час відгуку API для операцій CRUD не повинен перевищувати 500ms
- Пошук товарів повинен працювати за менше 1 секунди
- Фільтрація повинна відбуватися за менше 800ms
- WebSocket повідомлення в чаті повинні доставлятися за менше 100ms
- Система повинна підтримувати до 10,000 товарів
- Система повинна підтримувати до 1,000 одночасних користувачів
- Система повинна обробляти до 100 замовлень на годину
- База даних повинна підтримувати до 100,000 записів замовлень
- Статичні ресурси повинні кешуватися в браузері
- API відповіді повинні стискатися через gzip
- Зображення повинні оптимізуватися та підтримувати lazy loading
- Redis [16] використовується для кешування частих запитів
- Система повинна бути доступна 99.5% часу
- Плановані технічні роботи проводяться у нічний час

- Система повинна автоматично відновлюватися після збоїв
- Всі критичні помилки логуються
- Користувачі бачать зрозумілі повідомлення про помилки
- Система не показує технічні деталі помилок кінцевим користувачам
- Помилки інтеграцій (Stripe [18], Нова Пошта) обробляються gracefully
- Автоматичне резервне копіювання БД щоденно
- Зберігання резервних копій за останні 30 днів
- Можливість відновлення з backup за менше 1 години
- Паролі зберігаються у вигляді bcrypt хешів
- JWT токени підписані секретним ключем
- Access tokens дійсні 1 день, refresh tokens - 7 днів
- Після 5 невдалих спроб входу акаунт блокується на 15 хвилин
- Всі API ендпоінти захищені CORS політикою
- HTTPS обов'язковий для production середовища
- Чутлива інформація (паролі, токени) не логується
- SQL injection захищений через ORM [30] (TypeORM)
- XSS захист через sanitization вводу користувача
- Ролевий доступ (RBAC) для всіх захищених ресурсів
- Користувачі можуть отримати доступ лише до своїх даних
- API ендпоінти адміністратора захищені JWT guards
- Дані платіжних карток не зберігаються в системі
- Оплата обробляється через PCI-DSS сертифікований Stripe [18]
- Webhook підписи від Stripe валідуються [18]
- Підтримка Chrome 90+
- Підтримка Firefox 88+
- Підтримка Safari 14+
- Підтримка Edge 90+
- Адаптивний дизайн для desktop (1920px [5], 1366px)
- Адаптивний дизайн для tablet (768px [5], 1024px)
- Адаптивний дизайн для mobile (320px [5], 375px, 414px)

- Touch-friendly інтерфейс для мобільних пристроїв
- API сумісне з актуальними версіями Stripe API [18]
- API сумісне з Нова Пошта [26] API v2
- Підтримка Telegram Bot [28] API останньої версії
- Підтримка WhatsApp Business API [27]
- Детальне логування всіх API запитів
- Логування змін критичних даних (замовлення, платежі)
- Rotation логів кожні 30 днів
- Інтеграція з Sentry [32] для моніторингу помилок
- Моніторинг доступності системи
- Моніторинг використання ресурсів (CPU, RAM, Disk)
- Алерти при критичних помилках
- Моніторинг часу відгуку API
- Наявність README з інструкціями по встановленню
- Документація API endpoints
- Документація архітектури системи
- Інструкції по deployment
- Інтуїтивна навігація без необхідності навчання
- Послідовність елементів UI на всіх сторінках
- Використання стандартних UI паттернів
- Доступність через клавіатуру (keyboard navigation)
- Зрозумілі повідомлення про помилки українською мовою
- Success повідомлення після успішних операцій
- Loading індикатори для довгих операцій
- Підтвердження для критичних дій (видалення)
- Основна мова інтерфейсу - українська
- Підтримка UAH як основної валюти
- Формат дат згідно з українськими стандартами
- Формат телефонів: +380XXXXXXXXXX
- Застосунок працює на Node.js 20+

- Застосунок працює на PostgreSQL [15] 15+
- Застосунок працює на Redis 7+
- Можливість розгортання в Docker [19] контейнерах
- Всі налаштування через змінні оточення (.env)
- Різні конфігурації для dev, staging, production
- Можливість роботи без зовнішніх сервісів в режимі розробки
- Бізнес-вимоги:
- Забезпечити онлайн платформу для продажу електроніки та гаджетів
- Автоматизувати процес від замовлення до доставки
- Збільшити конверсію покинутих кошиків через email нагадування
- Підвищити задоволеність клієнтів через багатоканальну підтримку
- Середній час обробки замовлення - менше 5 хвилин
- Конверсія відвідувач → покупець - мінімум 2%
- Середній чек - мінімум 1000 UAH
- Повернення покупців - мінімум 30%
- Бюджет на розробку обмежений
- Використання безкоштовних або дешевих сервісів де можливо
- Можливість масштабування при зростанні бізнесу
- Відповідність законам України про електронну комерцію
- Збереження персональних даних згідно з GDPR принципами
- Надання чеку/інвойсу для кожного замовлення
- Право на повернення товару протягом 14 днів
- Вимоги до інтерфейсу користувача:
- Мінімалістичний та сучасний дизайн
- Використання корпоративних кольорів (primary-600: #2563eb)
- Tailwind CSS для стилізації
- Виділення активних елементів та hover ефектів
- Фіксований header з навігацією
- Хлібні крихти (breadcrumbs) для категорій і продуктів
- Швидкий доступ до кошика та профілю

- Dropdown меню для адміністративних розділів
- Inline валідація полів форм
- Автоматичний фокус на першому полі
- Placeholder текст для пояснення полів
- Кнопки з loading станом для запобігання подвійних відправок
- Toast нотифікації для швидких повідомлень
- Modal діалоги для підтвердження дій
- Progress bar для довгих операцій
- Skeleton loaders для контенту що завантажується
- ARIA labels для інтерактивних елементів
- Достатній контраст тексту (мінімум 4.5:1)
- Можливість навігації через Tab
- Альтернативний текст для зображень

2.2 Менеджмент процесу проєктування системи

Управління завданнями за допомогою ClickUp [35].

Для ефективного управління проєктом та організації робочого процесу було обрано платформу ClickUp - сучасний інструмент для управління завданнями та проєктами. Згідно з офіційною документацією, ClickUp позиціонується як "everything app for work", що об'єднує завдання, документи, цілі та комунікацію в єдиному середовищі. Платформа надає широкий спектр функціональних можливостей, включаючи понад 50 дій, тригерів та умов для автоматизації рутинних процесів, що дозволяє зменшити ручну роботу та підвищити продуктивність команди. Система підтримує кастомні статуси, підзавдання, залежності між завданнями [35], story points та пріоритети для гнучкого планування робочого процесу.

Платформа включає потужні інструменти співпраці, зокрема онлайн-дошки для брейнстормінгу, командний месенджер та функцію редагування документів у реальному часі. Вбудований редактор ClickUp Docs забезпечує можливість спільного редагування [35], створення бази знань та wiki для

проектної документації. Для відстеження прогресу доступні понад 15 видів відображення даних з візуальними віджетами, включаючи Workload View для візуалізації щоденного навантаження команди та виявлення перевантажених або недозавантажених учасників. Важливою складовою платформи є ClickUp AI - штучний інтелект для автоматичного створення завдань, генерації планів проєктів та оновлення статусів, що значно прискорює рутинні операції з управління проєктом.

Проєкт інтернет-магазину музичних інструментів організовано з використанням ієрархічної структури ClickUp [35]. Головний простір проєкту має назву FullMag, в якому створено папку Development для завдань розробки. Ця папка містить окремі списки для різних напрямків роботи: Backend API для завдань серверної частини, Web Frontend для веб-інтерфейсу, Mobile App для мобільного застосунку, Infrastructure для налаштування інфраструктури та DevOps, а також Bugs для списку виявлених помилок. Кожне завдання містить детальний опис, критерії приймання, прикріплені файли дизайну з Figma, посилання на відповідні гілки в Git та коментарі з обговоренням реалізації.

Типовий життєвий цикл завдання в проєкті включає шість послідовних статусів. Спочатку завдання потрапляє до Backlog, де воно визначене, але ще не заплановане до виконання. Після планування на поточний спринт завдання переходить у статус To Do [36]. Коли розробник починає активно працювати над завданням, воно отримує статус In Progress. Після написання коду завдання переходить на Code Review для перевірки іншими членами команди. Далі функціонал проходить Testing, і після успішного завершення всіх етапів завдання позначається як Done.

На рис. 2.1 можна побачити створений проєкт з дошкою для поточної системи:

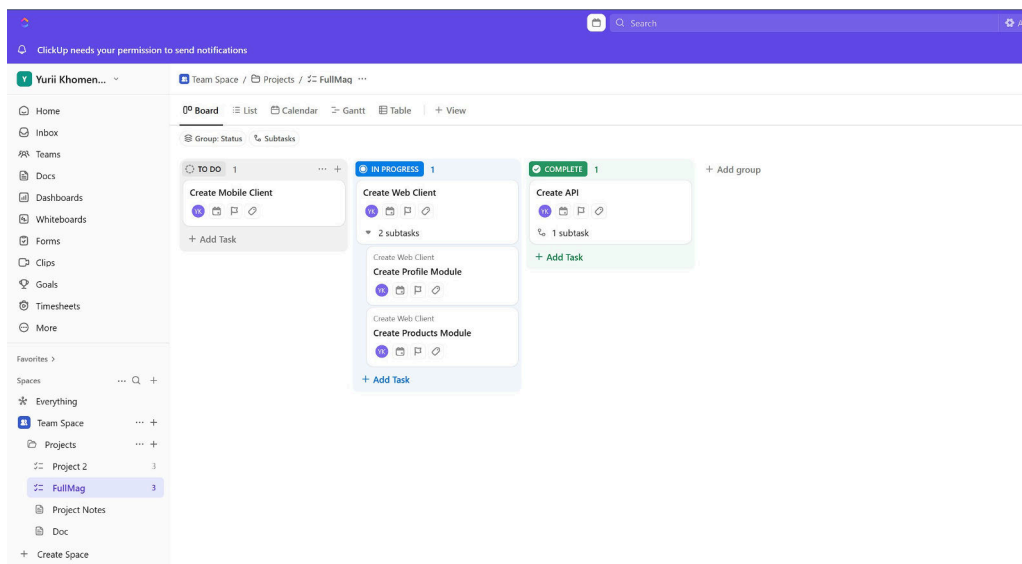


Рис. 2.1. Дошка с Clickup

Контроль версій за допомогою Git.

Для управління вихідним кодом проекту використовується розподілена система контролю версій Git з хостингом репозиторію на платформі GitHub. Згідно з офіційною документацією GitHub, GitHub Flow є легким робочим процесом на основі гілок [37], який підтримує команди та проекти з регулярними розгортаннями. У проекті застосовується саме ця стратегія, яка базується на принципі постійної готовності головної гілки до розгортання. Як зазначає GitKraken, GitHub Flow є найпростішою з основних стратегій розгалуження Git [38] і добре підходить для невеликих команд та веб-застосунків завдяки підтримці безперервної інтеграції та доставки.

Структура гілок проекту складається з головної гілки main, що містить стабільний production-ready код. Згідно з рекомендаціями Microsoft Azure DevOps, код у головній гілці повинен проходити тести, успішно збиратися та завжди бути актуальним. Для розробки нових можливостей створюються функціональні гілки з префіксом feature/, наприклад feature/user-authentication або feature/shopping-cart. Виправлення помилок виконуються в гілках з префіксом bugfix/, а термінові виправлення критичних помилок у production - в гілках hotfix/.

Для забезпечення читабельності історії комітів використовується

стандарт Conventional Commits - легка конвенція для повідомлень комітів [40], яка надає простий набір правил для створення явної історії змін. Ця конвенція узгоджується з семантичним версіонуванням SemVer, описуючи функції, виправлення та критичні зміни у повідомленнях комітів. Структура повідомлення коміту має формат, що включає тип, опціональну область та опис. Тип feat використовується для додавання нової функціональності, що відповідає MINOR у SemVer. Тип fix застосовується для виправлення помилок, що відповідає PATCH у SemVer. Інші типи включають docs для змін у документації, style для форматування коду без зміни логіки, refactor для рефакторингу коду, test для додавання або зміни тестів, та chore для технічних змін. Переваги використання Conventional Commits включають автоматичну генерацію CHANGELOG, автоматичне визначення версії на основі типів комітів та покращення комунікації про характер змін.

Перед злиттям коду в основні гілки обов'язково проводиться перевірка коду через Pull Request. Згідно з рекомендаціями GitLab, перевірка коду є критичною для покращення якості коду, а оптимальна кількість рецензентів - два. Microsoft Azure DevOps рекомендує захищати критичні гілки, налаштовуючи правила, що вимагають перевірки pull request та проходження статусних перевірок перед злиттям. Процес починається зі створення розробником Pull Request з детальним описом змін. Далі PR автоматично перевіряється CI/CD pipeline, який виконує літінг, запускає тести та перевіряє збірку проєкту. Рецензент перевіряє код на відповідність стандартам, аналізує логіку та виявляє потенційні проблеми. Після схвалення рецензентом PR зливається в цільову гілку.

Проєкт організований як монорепозиторій з використанням рпрт workspaces, що дозволяє зберігати всі компоненти системи в єдиному репозиторії. Директорія /apps/web містить веб-застосунок на Next.js, /apps/mobile - мобільний застосунок на React Native [25]. Серверний API на NestJS розміщено в /services/api [31]. Спільні типи та утиліти знаходяться в /packages/common, а документація проєкту - в директорії /docs. На рис. 2.2 зображено репозиторій

СИСТЕМИ:

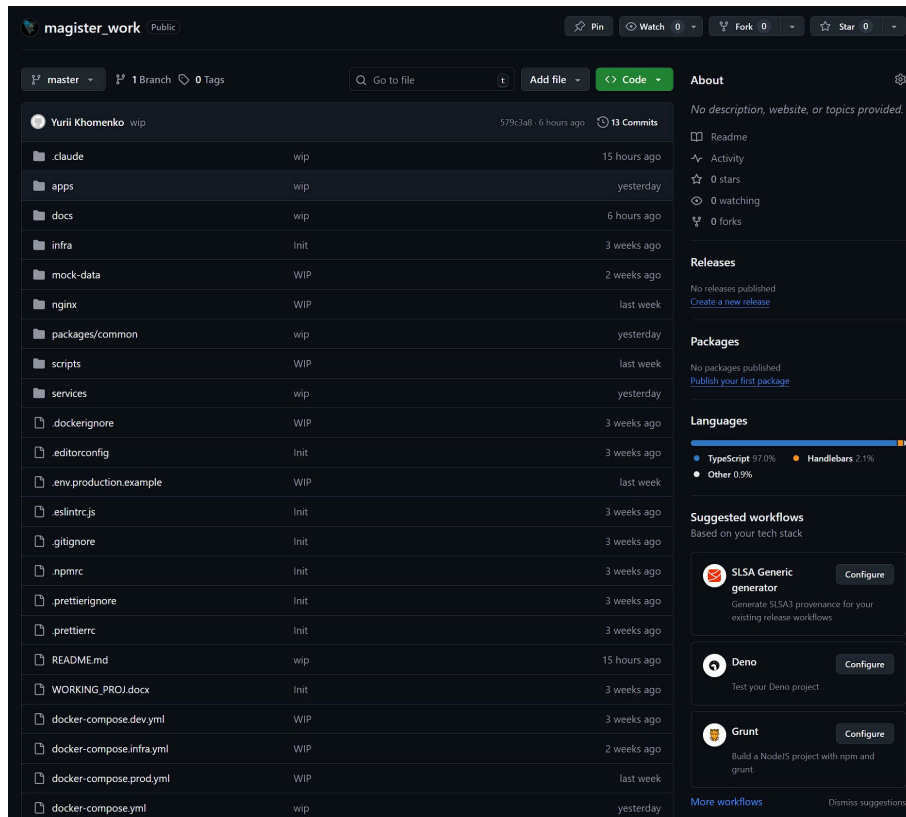


Рисунок 2.2. Репозиторій створюваної системи

Дизайн інтерфейсу за допомогою Figma.

Figma - це хмарний інструмент для дизайну інтерфейсів, який забезпечує спільну роботу над макетами в реальному часі. На конференції Config 2025 Figma представила найбільший набір нових продуктів, включаючи Figma Make - AI-інструмент для генерації коду з дизайнів, Figma Sites для публікації динамічних веб-сайтів та Figma Draw з розширеними інструментами векторного редагування. На конференції Schema 2025 було анонсовано значні покращення для роботи з дизайн-системами.

Для проєкту було створено комплексну дизайн-систему, що включає колірну палітру з визначенням основних та акцентних кольорів, нейтральних відтінків та семантичних кольорів для станів success, warning та error. Типографіка охоплює шрифтові стилі для заголовків від H1 до H6, основного тексту, підписів та кнопок. Бібліотека компонентів містить усі необхідні UI-елементи: кнопки, поля вводу, картки, модальні вікна та інші. Також розроблено

набір іконок для навігації, дій користувача та категорій товарів. Використовуються змінні Figma для кольорів та spacing, при цьому Professional план підтримує до 10 режимів змінних на колекцію. Нова функція Slots дозволяє додавати текст, іконки або вкладені компоненти без відокремлення від головної системи, що спрощує підтримку дизайн-системи.

Дизайн-проект організований у вигляді окремих сторінок для різних аспектів системи. Сторінка Design System містить базові елементи дизайн-системи. Макети веб-версії розділені на Web Desktop для десктопних екранів та Web Mobile для адаптивних мобільних версій. Окрема сторінка Mobile App призначена для дизайну мобільного застосунку [7], а Prototypes - для інтерактивних прототипів, що використовуються для тестування користувацького досвіду.

У Figma було спроектовано основні екрани інтернет-магазину (рис. 2.3). Головна сторінка включає банери, категорії товарів та рекомендовані продукти. Каталог товарів представляє сітку продуктів з фільтрами та можливостями сортування. Сторінка товару відображає детальну інформацію, галерею зображень, характеристики та відгуки покупців. Також розроблено дизайн кошика зі списком товарів та можливістю зміни кількості, сторінку оформлення замовлення з формою даних доставки та оплати, особистий кабінет з профілем користувача та історією замовлень, форми авторизації для входу та реєстрації, а також інтерфейс чату підтримки для спілкування з оператором.

Згідно з офіційною документацією Figma, Dev Mode надає все необхідне для навігації по дизайн-файлах та трансформації дизайнів у код. Dev Mode дозволяє дизайнерам та розробникам залишатися на одній хвилі, забезпечуючи збереження важливих деталей у процесі передачі. Ready for Dev View відображає всі дизайни, позначені як готові до розробки, в єдиному місці, що дозволяє розробникам швидко знаходити потрібне. Focus View забезпечує ізоляцію конкретного дизайну для детального інспектування шарів та перегляду історії версій. Дизайнери можуть використовувати Annotations для додавання анотацій з додатковим контекстом, специфікаціями та вимірами. Code Connect дозволяє

зв'язувати Figma з кодовою базою дизайн-системи для миттєвої генерації готових до використання фрагментів коду. Інтеграція з VS Code через Figma for VS Code дозволяє інспектувати дизайн-файли без перемикання контексту між програмами.

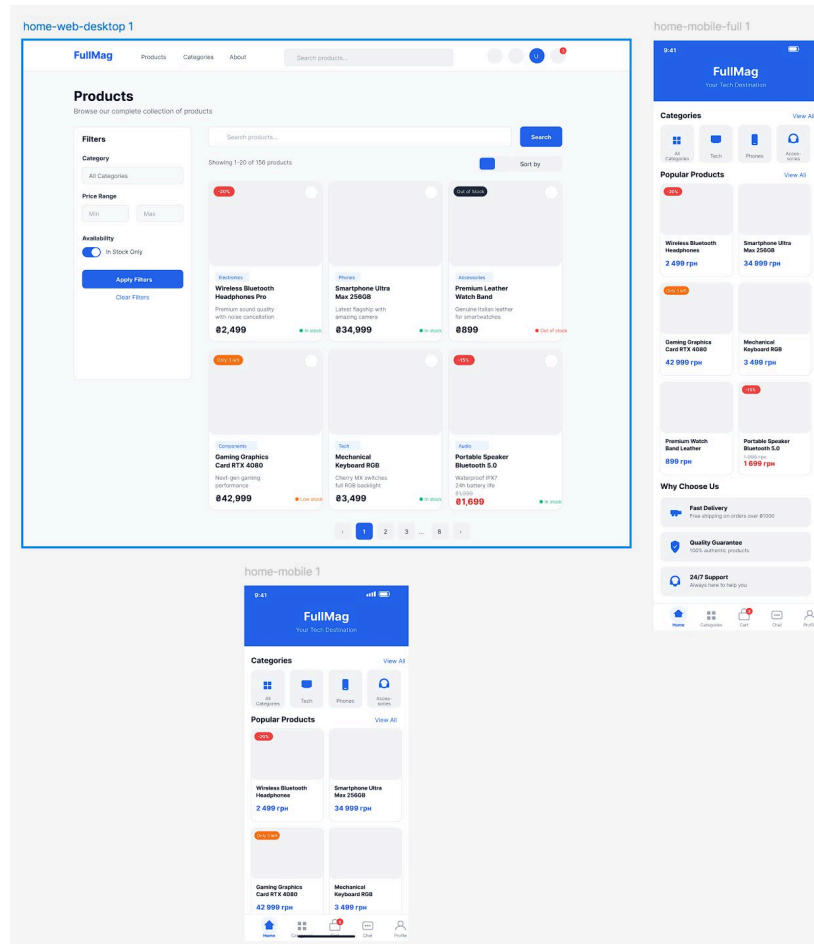


Рисунок 2.3. Макети додатку в Figma

2.3 Проектування інтерфейсу користувача

Інтерфейс користувача системи FullMag побудований з використанням сучасних веб-технологій та принципів Mobile-First дизайну. Архітектура фронтенду базується на компонентному підході React з використанням Next.js App Router для серверного рендерингу та оптимізації SEO [9].

Основна мета проектування - створення інтуїтивно зрозумілого, швидкого та адаптивного інтерфейсу, який забезпечує комфортний досвід покупок як на мобільних пристроях, так і на десктопних комп'ютерах.

Технологічний стек фронтенду.

Для реалізації інтерфейсу використовуються наступні технології:

- Next.js 14 - фреймворк для React [7] з підтримкою серверних компонентів та App Router
- React 18 [7] - бібліотека для побудови користувацьких інтерфейсів
- TypeScript [12] - типізована мова програмування для підвищення надійності коду
- Tailwind CSS - utility-first CSS фреймворк для стилізації
- Zustand - легковісна бібліотека для управління станом
- React Hook Form [7] - бібліотека для роботи з формами
- Recharts - бібліотека для візуалізації даних у вигляді графіків
- Socket.IO Client - клієнт для real-time комунікації
- Axios - HTTP-клієнт для взаємодії з API

Структура фронтенд-додатку.

Фронтенд організований за функціональним принципом, де компоненти групуються за доменною областю:

- app/ - сторінки Next.js App Router [14] (products, cart, checkout, profile, admin)
- components/ - перевикористовувані React-компоненти
- lib/ - утиліти, API-клієнт та Zustand stores
- styles/ - глобальні стилі та конфігурація Tailwind

Компонентна архітектура.

Система компонентів побудована ієрархічно з чітким розподілом відповідальності.

Компоненти макету:

- Header - глобальна навігація з логотипом, меню категорій, кошиком та профілем користувача
- Footer - інформаційний блок з посиланнями та віджетом чату
- BannerProvider - система промо-банерів

Компоненти каталогу:

- ProductList - контейнер списку товарів з підтримкою grid/list режимів

- ProductCard - картка товару з зображенням, ціною та кнопками дій
- FilterSidebar - бокова панель фільтрації з категоріями, ціновим діапазоном та наявністю
- Pagination - компонент пагінації
- SortDropdown - сортування товарів
- ViewToggle - перемикач відображення grid/list

Компоненти товару:

- ProductImageGallery - галерея зображень з можливістю збільшення
- ProductInfo - інформація про товар з ціною та кнопками дій
- ProductTabs - вкладки з описом, характеристиками та відгуками
- ProductReviews - блок відгуків покупців

Компоненти кошика та замовлення:

- CartItem - елемент кошика з управлінням кількістю
- DeliveryForm - форма доставки з пошуком міст та відділень
- StripePaymentForm - форма оплати через Stripe

Управління станом.

Для управління станом використовується Zustand з наступними stores:

- AuthStore - зберігає інформацію про авторизованого користувача, токен доступу та методи login/logout. Дані персистуються у localStorage для збереження сесії.
- CartStore - управляє вмістом кошика з синхронізацією з сервером. Надає методи додавання, видалення та оновлення кількості товарів.
- ProductStore - зберігає список товарів, фільтри та стан завантаження для сторінки каталогу.
- ChatStore - управляє повідомленнями чату з підтримкою WebSocket-з'єднання.

Дизайн-система.

Кольорова палітра:

- Primary (Sky Blue) - основний колір для кнопок, посилань та акцентів
- Gray Scale - нейтральні кольори для тексту, фонів та меж

- Semantic Colors - зелений для успіху, червоний для помилок, жовтий для попереджень

Типографіка:

- Шрифт Inter з підтримкою кирилиці
- Ієрархія заголовків від H1 до H6
- Розміри тексту від xs до 2xl

Компоненти UI:

- Кнопки - primary, secondary, outline, ghost варіанти
- Картки - з тінню та заокругленими кутами
- Форми - інпути, селекти, чекбокси зі стилізацією
- Модальні вікна - для підтверджень та діалогів

Адаптивний дизайн.

Інтерфейс побудований за принципом Mobile-First з використанням брейкпоінтів Tailwind:

- Mobile (< 640px) - одноколонковий layout
- Tablet (640px - 1024px) - двоколонковий layout
- Desktop (> 1024px) - повний layout з бічною панеллю

Ключові адаптивні елементи:

- Грід товарів змінюється від 1 до 4 колонок
- Навігація трансформується в мобільне меню
- Фільтри згортаються в колапс-панель на мобільних
- Таблиці перетворюються на картки
- Основні сторінки інтерфейсу.
- Каталог товарів - сторінка з сіткою товарів, бічною панеллю фільтрів, сортуванням та пагінацією. Підтримує режими відображення grid та list.
- Сторінка товару - детальна інформація з галереєю зображень, описом, характеристиками, відгуками та рекомендованими товарами.

- Кошик - список доданих товарів з можливістю зміни кількості, видалення та переходу до оформлення.
- Оформлення замовлення - багатокроковий процес з введенням адреси доставки, вибором способу оплати та підтвердженням.
- Профіль користувача - особистий кабінет з історією замовлень, збереженими адресами та налаштуваннями.
- Список бажань - збережені товари для подальшої покупки.
- Порівняння товарів - табличне порівняння характеристик до 4 товарів.

Адміністративна панель.

Адмін-панель надає інструменти для управління магазином:

- Dashboard - аналітика з графіками продажів, замовлень та відвідувачів
- Товари - CRUD операції, масовий імпорт з CSV/Excel
- Категорії - ієрархічне управління категоріями
- Замовлення - перегляд та обробка замовлень
- Клієнти - управління користувачами
- Купони - створення та управління знижками
- Аналітика - детальні звіти та метрики

UX-патерни.

В інтерфейсі реалізовані наступні UX-патерни:

- Loading States - скелетон-лоадери та спінери під час завантаження
- Empty States - інформативні повідомлення з call-to-action для порожніх сторінок
- Error Handling - toast-повідомлення та inline-помилки у формах
- Confirmation Dialogs - модальні вікна для підтвердження критичних дій
- Optimistic UI - миттєве відображення змін у кошику до відповіді сервера
- Infinite Scroll - підтримка безкінечної прокрутки для каталогу

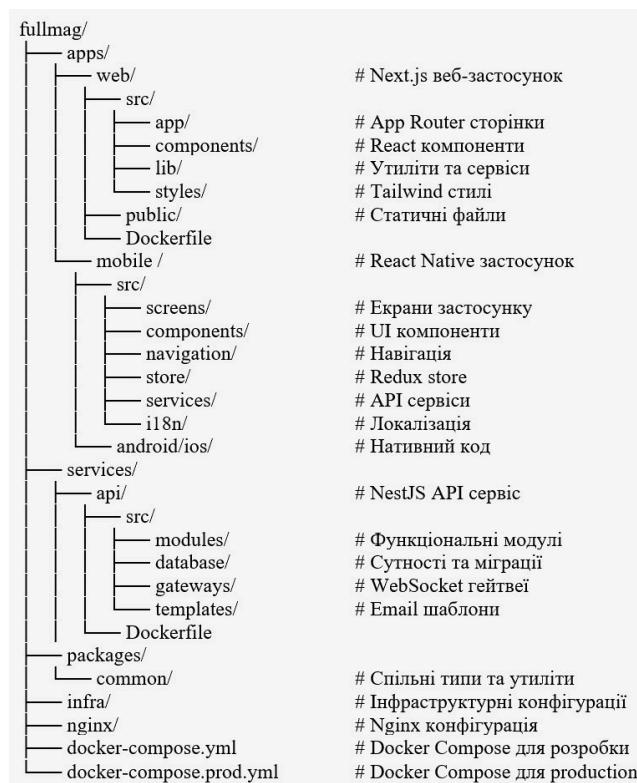


Рисунок 2.4. Структура каталогів проекту

2.4 Проектування структури бази даних

Таблиця users (табл. 2.1) зберігає облікові записи користувачів з їхніми даними (email, пароль, ім'я, роль).

Таблиця 2.1

Структура таблиці users

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор користувача (UUID)
email	VARCHAR(255) UNIQUE	Електронна пошта користувача
password	VARCHAR(255)	Хешований пароль
firstName	VARCHAR(100)	Ім'я користувача
lastName	VARCHAR(100)	Прізвище користувача
phone	VARCHAR(20)	Номер телефону
role	ENUM('customer', 'employee', 'admin')	Роль користувача в системі
avatar	VARCHAR(500)	URL аватара користувача
isActive	BOOLEAN	Статус активності облікового запису

createdAt	TIMESTAMP	Дата та час створення запису
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця categories (табл. 2.2) представляє ієрархічний каталог категорій товарів з підтримкою вкладеності через parentId.

Таблиця 2.2

Структура таблиці categories

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор категорії (UUID)
name	VARCHAR(100)	Назва категорії
slug	VARCHAR(100) UNIQUE	URL-friendly ідентифікатор категорії
description	TEXT	Опис категорії
image	VARCHAR(500)	URL зображення категорії
parentId	VARCHAR(36) FK	Посилання на батьківську категорію
sortOrder	INT	Порядок сортування категорії
isActive	BOOLEAN	Статус активності категорії
createdAt	TIMESTAMP	Дата та час створення запису
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця characteristics (табл. 2.3) визначає типи характеристик (розмір, колір тощо), які можна призначати товарам певної категорії.

Таблиця 2.3

Структура таблиці characteristics

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор характеристики (UUID)
name	VARCHAR(100)	Назва характеристики
type	ENUM('string', 'number', 'boolean', 'select')	Тип значення характеристики
unit	VARCHAR(50)	Одиниця виміру (якщо застосовно)
options	JSON	Можливі значення для типу select

categoryId	VARCHAR(36) FK	Посилання на категорію
isRequired	BOOLEAN	Чи є характеристика обов'язковою
isFilterable	BOOLEAN	Чи використовується для фільтрації
sortOrder	INT	Порядок відображення
createdAt	TIMESTAMP	Дата та час створення запису
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Основна таблиця товарів (табл. 2.4) з інформацією про назву, ціну, кількість на складі та зображення.

Таблиця 2.4

Структура таблиці products

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор товару (UUID)
name	VARCHAR(255)	Назва товару
slug	VARCHAR(255) UNIQUE	URL-friendly ідентифікатор товару
sku	VARCHAR(50) UNIQUE	Артикул товару
description	TEXT	Повний опис товару
shortDescription	VARCHAR(500)	Короткий опис товару
price	DECIMAL(10,2)	Ціна товару
oldPrice	DECIMAL(10,2)	Стара ціна (для акцій)
quantity	INT	Кількість на складі
categoryId	VARCHAR(36) FK	Посилання на категорію
brand	VARCHAR(100)	Бренд виробника
images	JSON	Масив URL зображень товару
isActive	BOOLEAN	Статус активності товару
isFeatured	BOOLEAN	Чи є товар рекомендованим
viewsCount	INT	Кількість переглядів
createdAt	TIMESTAMP	Дата та час створення запису
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Зв'язкова таблиця між товарами та характеристиками (табл. 2.5), що зберігає конкретні значення.

Таблиця 2.5

Структура таблиці product_characteristics

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор зв'язку (UUID)
productId	VARCHAR(36) FK	Посилання на товар
characteristicId	VARCHAR(36) FK	Посилання на характеристику
value	VARCHAR(500)	Значення характеристики для товару
createdAt	TIMESTAMP	Дата та час створення запису
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця reviews (табл. 2.6) представляє відгуки користувачів на товари з оцінками, перевагами та недоліками.

Таблиця 2.6

Структура таблиці reviews

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор відгуку (UUID)
productId	VARCHAR(36) FK	Посилання на товар
userId	VARCHAR(36) FK	Посилання на користувача
rating	INT	Оцінка товару (1-5)
title	VARCHAR(200)	Заголовок відгуку
comment	TEXT	Текст відгуку
pros	TEXT	Переваги товару
cons	TEXT	Недоліки товару
isVerified	BOOLEAN	Чи підтверджена покупка
isApproved	BOOLEAN	Чи схвалений відгук модератором
createdAt	TIMESTAMP	Дата та час створення запису
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця orders (табл. 2.7) представляє замовлення клієнтів з інформацією про доставку, оплату та статус.

Таблиця 2.7

Структура таблиці orders

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор замовлення (UUID)
orderNumber	VARCHAR(20) UNIQUE	Номер замовлення
userId	VARCHAR(36) FK	Посилання на користувача
status	ENUM('pending', 'confirmed', 'processing', 'shipped', 'delivered', 'cancelled')	Статус замовлення
totalAmount	DECIMAL(10,2)	Загальна сума замовлення
shippingAddress	JSON	Адреса доставки
shippingMethod	VARCHAR(50)	Спосіб доставки
shippingCost	DECIMAL(10,2)	Вартість доставки
paymentMethod	VARCHAR(50)	Спосіб оплати
paymentStatus	ENUM('pending', 'paid', 'failed', 'refunded')	Статус оплати
notes	TEXT	Примітки до замовлення
createdAt	TIMESTAMP	Дата та час створення замовлення
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця order_items (табл. 2.8) представляє позиції замовлень, що фіксують товари, їх кількість та ціну на момент покупки.

Структура таблиці order_items

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор позиції (UUID)
orderId	VARCHAR(36) FK	Посилання на замовлення
productId	VARCHAR(36) FK	Посилання на товар
productName	VARCHAR(255)	Назва товару на момент замовлення
productSku	VARCHAR(50)	Артикул товару на момент замовлення
price	DECIMAL(10,2)	Ціна за одиницю на момент замовлення
quantity	INT	Кількість одиниць товару
totalPrice	DECIMAL(10,2)	Загальна вартість позиції
createdAt	TIMESTAMP	Дата та час створення запису
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця chats (табл. 2.9) представляє чати підтримки між клієнтами та працівниками з прив'язкою до замовлень.

Структура таблиці chats

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор чату (UUID)
customerId	VARCHAR(36) FK	Посилання на клієнта
employeeId	VARCHAR(36) FK	Посилання на працівника
subject	VARCHAR(200)	Тема чату
status	ENUM('open', 'in_progress', 'resolved', 'closed')	Статус чату
priority	ENUM('low', 'medium', 'high')	Пріоритет звернення
orderId	VARCHAR(36) FK	Посилання на замовлення (якщо пов'язано)
lastMessageAt	TIMESTAMP	Дата та час останнього повідомлення
createdAt	TIMESTAMP	Дата та час створення чату

updatedAt	TIMESTAMP	Дата та час останнього оновлення
-----------	-----------	----------------------------------

Таблиця messages (табл. 2.10) представляє повідомлення в чатах з підтримкою різних типів контенту (текст, зображення, файли).

Таблиця 2.10

Структура таблиці messages

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор повідомлення (UUID)
chatId	VARCHAR(36) FK	Посилання на чат
senderId	VARCHAR(36) FK	Посилання на відправника
content	TEXT	Текст повідомлення
type	ENUM('text', 'image', 'file', 'system')	Тип повідомлення
attachments	JSON	Прикріплені файли
isRead	BOOLEAN	Чи прочитане повідомлення
readAt	TIMESTAMP	Дата та час прочитання
createdAt	TIMESTAMP	Дата та час відправлення
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця carts (табл. 2.11) представляє кошики покупців для збереження обраних товарів перед оформленням замовлення.

Таблиця 2.11

Структура таблиці carts

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор кошика (UUID)
userId	VARCHAR(36) FK	Посилання на користувача
createdAt	TIMESTAMP	Дата та час створення кошика
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця cart_items (табл. 2.12) зберігає позиції кошика з інформацією про обрані товари та їх кількість.

Структура таблиці cart_items

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор позиції (UUID)
cartId	VARCHAR(36) FK	Посилання на кошик
productId	VARCHAR(36) FK	Посилання на товар
quantity	INT	Кількість одиниць товару
price	DECIMAL(10,2)	Ціна товару на момент додавання
createdAt	TIMESTAMP	Дата та час додавання до кошика
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця wishlists (табл. 2.13) представляє списки бажаних товарів користувачів для збереження цікавих позицій.

Структура таблиці wishlists

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор запису (UUID)
userId	VARCHAR(36) FK	Посилання на користувача
productId	VARCHAR(36) FK	Посилання на товар
createdAt	TIMESTAMP	Дата та час додавання до списку

Таблиця payments (табл. 2.14) зберігає інформацію про платежі замовлень через різні платіжні системи (LiqPay, Monobank, ПриватБанк).

Структура таблиці payments

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор платежу (UUID)
orderId	VARCHAR(36) FK	Посилання на замовлення
provider	ENUM('liqpay', 'monobank', 'privat24', 'cash')	Платіжний провайдер

providerPaymentId	VARCHAR(255)	Ідентифікатор платежу у провайдера
status	ENUM('pending', 'processing', 'success', 'failed', 'refunded')	Статус платежу
amount	DECIMAL(10,2)	Сума платежу
currency	VARCHAR(3)	Валюта платежу (UAH, USD, EUR)
providerResponse	JSONB	Відповідь платіжної системи
createdAt	TIMESTAMP	Дата та час створення платежу
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця coupons (табл. 2.15) зберігає промокоди та купони на знижки з гнучкими налаштуваннями лімітів використання.

Таблиця 2.15

Структура таблиці coupons

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор купона (UUID)
code	VARCHAR(50) UNIQUE	Код купона
description	TEXT	Опис купона
type	ENUM('percentage', 'fixed')	Тип знижки (відсоток або фіксована сума)
value	DECIMAL(10,2)	Значення знижки
minOrderAmount	DECIMAL(10,2)	Мінімальна сума замовлення
maxDiscount	DECIMAL(10,2)	Максимальна знижка
usageLimit	INT	Загальний ліміт використань
usageCount	INT	Кількість використань
perUserLimit	INT	Ліміт використань на користувача
startDate	TIMESTAMP	Дата початку дії
endDate	TIMESTAMP	Дата закінчення дії
isActive	BOOLEAN	Статус активності купона
createdAt	TIMESTAMP	Дата та час створення
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця coupon_usages (табл. 2.16) фіксує історію використання купонів

користувачами із зазначенням суми отриманої знижки.

Таблиця 2.16

Структура таблиці coupon_usages

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор запису (UUID)
couponId	VARCHAR(36) FK	Посилання на купон
userId	VARCHAR(36) FK	Посилання на користувача
orderId	VARCHAR(36) FK	Посилання на замовлення
discountAmount	DECIMAL(10,2)	Сума отриманої знижки
createdAt	TIMESTAMP	Дата та час використання

Таблиця notifications (табл. 2.17) зберігає сповіщення користувачам через різні канали доставки: email, push-повідомлення, SMS та Telegram.

Таблиця 2.17

Структура таблиці notifications

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор сповіщення (UUID)
userId	VARCHAR(36) FK	Посилання на користувача
type	VARCHAR(50)	Тип сповіщення
channel	ENUM('email', 'push', 'sms', 'telegram')	Канал доставки
title	VARCHAR(255)	Заголовок сповіщення
payload	JSONB	Дані сповіщення
status	ENUM('pending', 'sent', 'delivered', 'failed', 'read')	Статус доставки
sentAt	TIMESTAMP	Дата та час відправки
readAt	TIMESTAMP	Дата та час прочитання
createdAt	TIMESTAMP	Дата та час створення

Таблиця banners (табл. 2.18) представляє рекламні банери для головної сторінки та сторінок категорій із налаштуванням періоду показу.

Структура таблиці banners

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор банера (UUID)
title	VARCHAR(255)	Заголовок банера
description	TEXT	Опис банера
imageUrl	VARCHAR(500)	URL зображення банера
linkUrl	VARCHAR(500)	URL посилання при кліку
type	ENUM('hero', 'promo', 'category', 'sidebar')	Тип банера
status	ENUM('draft', 'active', 'scheduled', 'expired')	Статус банера
startDate	TIMESTAMP	Дата початку показу
endDate	TIMESTAMP	Дата закінчення показу
priority	INT	Пріоритет відображення
createdAt	TIMESTAMP	Дата та час створення
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця return_requests (табл. 2.19) зберігає заявки на повернення товарів з причинами, статусами обробки та інформацією про повернення коштів.

Структура таблиці return_requests

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор заявки (UUID)
orderId	VARCHAR(36) FK	Посилання на замовлення
userId	VARCHAR(36) FK	Посилання на користувача
orderId	VARCHAR(36) FK	Посилання на позицію замовлення
reason	ENUM('defective', 'wrong_item', 'not_as_described', 'changed_mind', 'other')	Причина повернення
description	TEXT	Детальний опис проблеми

status	ENUM('pending', 'approved', 'rejected', 'processing', 'completed')	Статус заявки
images	JSONB	Фотографії товару
refundAmount	DECIMAL(10,2)	Сума повернення
adminNotes	TEXT	Примітки адміністратора
processedById	VARCHAR(36) FK	Посилання на працівника, що обробив заявку
processedAt	TIMESTAMP	Дата та час обробки
createdAt	TIMESTAMP	Дата та час створення заявки
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця user_addresses (табл. 2.20) зберігає адреси доставки користувачів з підтримкою різних типів доставки та інтеграцією з Новою Поштою.

Таблиця 2.20

Структура таблиці user_addresses

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор адреси (UUID)
userId	VARCHAR(36) FK	Посилання на користувача
title	VARCHAR(100)	Назва адреси (Дім, Робота тощо)
recipientName	VARCHAR(200)	Ім'я отримувача
recipientPhone	VARCHAR(20)	Телефон отримувача
city	VARCHAR(100)	Місто доставки
address	VARCHAR(255)	Адреса доставки
deliveryType	ENUM('nova_poshta_warehouse', 'nova_poshta_courier', 'ukrposhta', 'self_pickup')	Тип доставки
warehouseRef	VARCHAR(50)	Референс відділення Нової Пошти
warehouseNumber	VARCHAR(20)	Номер відділення
isDefault	BOOLEAN	Чи є адреса за замовчуванням
createdAt	TIMESTAMP	Дата та час створення

updatedAt	TIMESTAMP	Дата та час останнього оновлення
-----------	-----------	----------------------------------

Таблиця telegram_users (табл. 2.21) представляє зв'язок облікових записів користувачів системи з їхніми акаунтами Telegram для надсилання сповіщень.

Таблиця 2.21

Структура таблиці telegram_users

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор запису (UUID)
userId	VARCHAR(36) FK	Посилання на користувача
chatId	BIGINT UNIQUE	Ідентифікатор чату Telegram
username	VARCHAR(100)	Username в Telegram
firstName	VARCHAR(100)	Ім'я в Telegram
lastName	VARCHAR(100)	Прізвище в Telegram
isActive	BOOLEAN	Чи активний зв'язок
createdAt	TIMESTAMP	Дата та час прив'язки
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця whatsapp_users (табл. 2.22) представляє зв'язок облікових записів користувачів системи з їхніми акаунтами WhatsApp для надсилання сповіщень.

Таблиця 2.22

Структура таблиці whatsapp_users

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор запису (UUID)
userId	VARCHAR(36) FK	Посилання на користувача
phoneNumber	VARCHAR(20) UNIQUE	Номер телефону WhatsApp
whatsappId	VARCHAR(50)	Ідентифікатор WhatsApp
pushName	VARCHAR(100)	Ім'я в WhatsApp
isActive	BOOLEAN	Чи активний зв'язок
createdAt	TIMESTAMP	Дата та час прив'язки

updatedAt	TIMESTAMP	Дата та час останнього оновлення
-----------	-----------	----------------------------------

Таблиця review_likes (табл. 2.23) зберігає відмітки «корисно» для відгуків, залишені іншими користувачами.

Таблиця 2.23

Структура таблиці review_likes

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор відмітки (UUID)
reviewId	VARCHAR(36) FK	Посилання на відгук
userId	VARCHAR(36) FK	Посилання на користувача
createdAt	TIMESTAMP	Дата та час відмітки

Таблиця review_replies (табл. 2.24) зберігає відповіді на відгуки від адміністрації магазину або інших користувачів з підтримкою вкладеності.

Таблиця 2.24

Структура таблиці review_replies

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор відповіді (UUID)
reviewId	VARCHAR(36) FK	Посилання на відгук
userId	VARCHAR(36) FK	Посилання на автора відповіді
parentReplyId	VARCHAR(36) FK	Посилання на батьківську відповідь (для вкладених відповідей)
content	TEXT	Текст відповіді
isOfficial	BOOLEAN	Чи є відповідь офіційною від магазину
createdAt	TIMESTAMP	Дата та час створення
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця review_complaints (табл. 2.25) зберігає скарги користувачів на відгуки для подальшої модерації адміністрацією.

Структура таблиці review_complaints

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор скарги (UUID)
reviewId	VARCHAR(36) FK	Посилання на відгук
userId	VARCHAR(36) FK	Посилання на автора скарги
reason	ENUM('spam', 'inappropriate', 'fake', 'offensive', 'other')	Причина скарги
description	TEXT	Детальний опис скарги
status	ENUM('pending', 'reviewed', 'resolved', 'rejected')	Статус скарги
resolvedById	VARCHAR(36) FK	Посилання на модератора
resolvedAt	TIMESTAMP	Дата та час розгляду
createdAt	TIMESTAMP	Дата та час створення скарги

Таблиця compare_items (табл. 2.26) зберігає товари, додані користувачами до списку порівняння для зіставлення характеристик.

Структура таблиці compare_items

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор запису (UUID)
userId	VARCHAR(36) FK	Посилання на користувача
productId	VARCHAR(36) FK	Посилання на товар
createdAt	TIMESTAMP	Дата та час додавання до порівняння

Таблиця cart_reminders (табл. 2.27) зберігає записи про нагадування щодо покинутих кошиків для маркетингових email-розсилок.

Структура таблиці cart_reminders

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор нагадування (UUID)
userId	VARCHAR(36) FK	Посилання на користувача
cartId	VARCHAR(36) FK	Посилання на кошик
reminderNumber	INT	Порядковий номер нагадування (1, 2, 3...)
emailSent	BOOLEAN	Чи було відправлено email
sentAt	TIMESTAMP	Дата та час відправки нагадування

Таблиця delivery_addresses (табл. 2.28) зберігає адреси доставки з інтеграцією API Нової Пошти для вибору відділень та кур'єрської доставки.

Таблиця 2.28

Структура таблиці delivery_addresses

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор адреси (UUID)
userId	VARCHAR(36) FK	Посилання на користувача
type	ENUM('nova_poshta_warehouse', 'nova_poshta_courier', 'self_pickup')	Тип доставки
cityRef	VARCHAR(50)	Референс міста в API Нової Пошти
cityName	VARCHAR(255)	Назва міста
warehouseRef	VARCHAR(50)	Референс відділення Нової Пошти
warehouseDescription	VARCHAR(500)	Опис відділення (адреса)
streetRef	VARCHAR(50)	Референс вулиці для кур'єрської доставки
streetName	VARCHAR(255)	Назва вулиці
houseNumber	VARCHAR(20)	Номер будинку
apartmentNumber	VARCHAR(20)	Номер квартири
recipientName	VARCHAR(200)	Ім'я отримувача
recipientPhone	VARCHAR(20)	Телефон отримувача

isDefault	BOOLEAN	Чи є адреса за замовчуванням
createdAt	TIMESTAMP	Дата та час створення
updatedAt	TIMESTAMP	Дата та час останнього оновлення

Таблиця `analytics_snapshots` (табл. 2.29) зберігає агреговані знімки аналітичних даних для побудови звітності та дашбордів.

Таблиця 2.29

Структура таблиці `analytics_snapshots`

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор знімка (UUID)
snapshotDate	DATE	Дата знімка даних
type	ENUM('daily', 'weekly', 'monthly')	Тип періоду агрегації
totalRevenue	DECIMAL(12,2)	Загальний дохід за період
totalOrders	INT	Кількість замовлень
completedOrders	INT	Кількість завершених замовлень
cancelledOrders	INT	Кількість скасованих замовлень
averageOrderValue	DECIMAL(12,2)	Середній чек замовлення
newCustomers	INT	Кількість нових клієнтів
returningCustomers	INT	Кількість повторних клієнтів
productsSold	INT	Кількість проданих товарів
lowStockProducts	INT	Кількість товарів з низьким залишком
topProducts	JSONB	Топ продуктів за продажами
topCategories	JSONB	Топ категорій за доходом
revenueByPaymentMethod	JSONB	Дохід за методами оплати
createdAt	TIMESTAMP	Дата та час створення знімка

Таблиця `import_profiles` (табл. 2.30) зберігає профілі налаштувань для імпорту товарів з CSV/Excel файлів з правилами мапінгу та валідації.

Структура таблиці import_profiles

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор профілю (UUID)
name	VARCHAR(255)	Назва профілю імпорту
description	TEXT	Опис профілю
delimiter	VARCHAR(10)	Роздільник колонок (кома, крапка з комою тощо)
encoding	VARCHAR(50)	Кодування файлу (utf-8, windows-1251 тощо)
hasHeader	BOOLEAN	Чи має файл заголовок
columnMapping	JSONB	Відповідність колонок файлу полям товару
transformations	JSONB	Правила трансформації даних (конвертація валют, ціни)
validationRules	JSONB	Правила валідації даних
attributeMapping	JSONB	Відповідність атрибутів товарів
isDefault	BOOLEAN	Чи є профіль за замовчуванням
isActive	BOOLEAN	Чи активний профіль
createdById	VARCHAR(36) FK	Посилання на користувача, що створив профіль
createdAt	TIMESTAMP	Дата та час створення
updatedAt	TIMESTAMP	Дата та час останнього оновлення

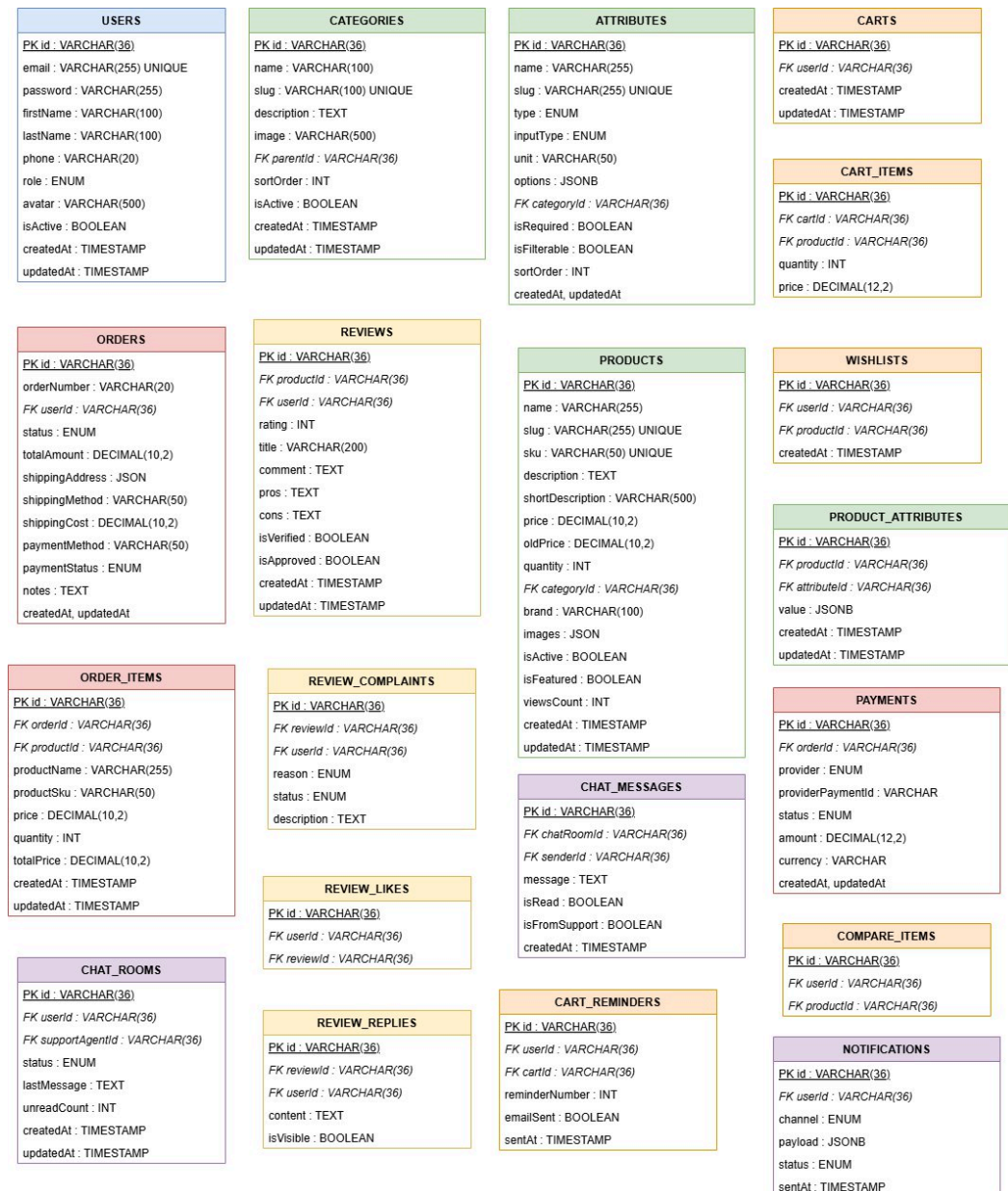
Таблиця import_history (табл. 2.31) фіксує історію операцій імпорту товарів з детальною статистикою успішних та невдалих записів.

Структура таблиці import_history

Поле	Тип даних	Опис
id	VARCHAR(36) PK	Унікальний ідентифікатор запису (UUID)
fileName	VARCHAR(255)	Назва імпортованого файлу
profileId	VARCHAR(36) FK	Посилання на профіль імпорту
importedBy	VARCHAR(36) FK	Посилання на користувача, що виконав імпорт
status	ENUM('pending', 'processing', 'completed', 'failed')	Статус операції імпорту

stats	JSONB	Статистика імпорту (успішних, невдалих, пропущених)
errorMessage	TEXT	Повідомлення про помилку (якщо є)
createdAt	TIMESTAMP	Дата та час початку імпорту
completedAt	TIMESTAMP	Дата та час завершення імпорту

На рис. 2.5-2.6 можна побачити схематичне зображення бази даних:



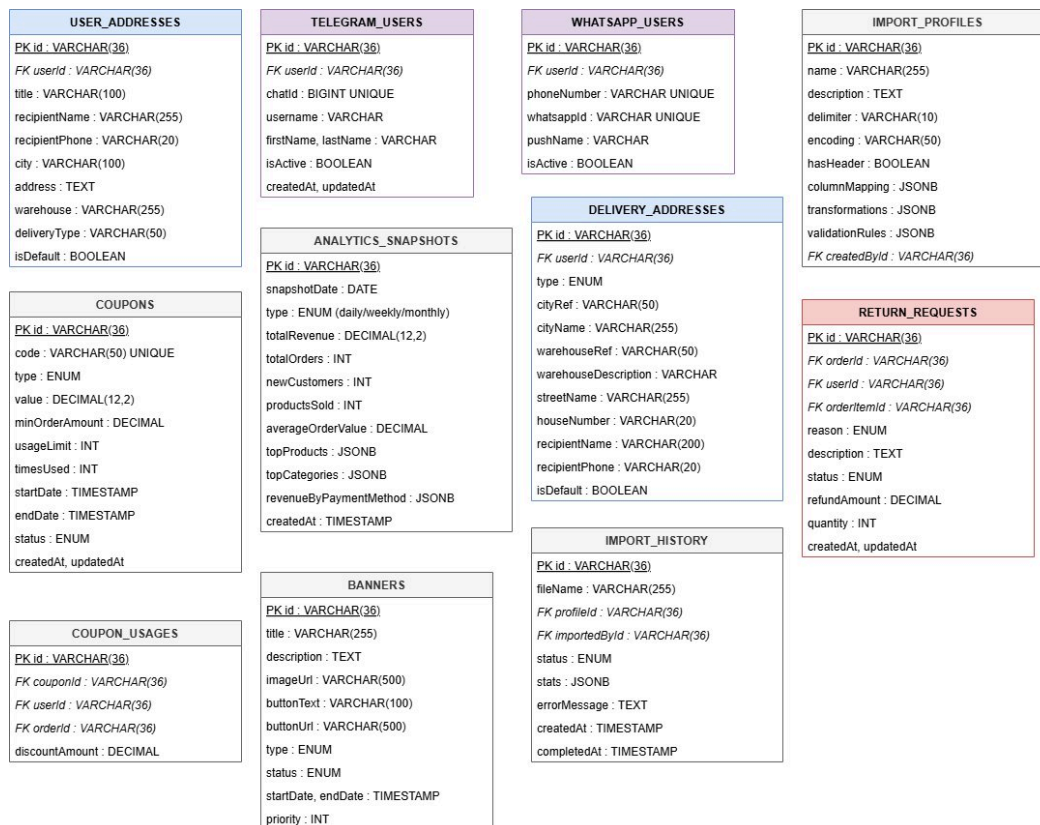


Рис. 2.5 – 2.6. Концептуальна схема бази даних

Висновок

У другому розділі було здійснено детальне проєктування програмної системи електронної комерції FullMag. Сформульовано та систематизовано понад 150 функціональних вимог до системи, що охоплюють усі аспекти роботи інтернет-магазину: автентифікацію та авторизацію користувачів, управління каталогом товарів з ієрархічними категоріями та динамічними атрибутами, пошук і фільтрацію, кошик та оформлення замовлень, систему відгуків, програму лояльності, підтримку клієнтів через чат та месенджери.

Визначено нефункціональні вимоги щодо продуктивності, масштабованості, надійності, безпеки та сумісності з сучасними браузерами та мобільними пристроями.

Описано організацію процесу розробки з використанням інструментів ClickUp для управління завданнями [35], GitHub для контролю версій з дотриманням GitHub Flow, Figma для проєктування інтерфейсів. Спроектовано користувацький інтерфейс веб-застосунку та мобільного додатку з урахуванням

принципів юзабіліті.

Розроблено детальну структуру бази даних з описом основних сутностей (користувачі, товари, категорії, замовлення, відгуки, чати) та їх взаємозв'язків. Визначено архітектуру серверної частини з поділом на модулі та їх функціональні обов'язки.

РОЗДІЛ 3. ОПИС РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

3.1 UML-моделювання

Уніфікована мова моделювання (Unified Modeling Language, UML) є стандартизованою мовою візуального моделювання, яка широко використовується в програмній інженерії для специфікації, візуалізації, конструювання та документування артефактів програмних систем. UML була розроблена в середині 1990-х років Граді Бучем, Джеймсом Рамбо та Іваром Якобсоном, які об'єднали свої методології об'єктно-орієнтованого аналізу та проектування. У 1997 році UML була прийнята консорціумом Object Management Group (OMG) як стандарт, і з того часу стала де-факто стандартом для моделювання програмного забезпечення.

Актуальна версія UML 2.5.1, опублікована OMG у грудні 2017 року, визначає 14 типів діаграм, які поділяються на дві основні категорії: структурні діаграми та діаграми поведінки. Структурні діаграми описують статичну архітектуру системи та включають діаграми класів, компонентів, розгортання, об'єктів, пакетів, профілів та композитної структури. Діаграми поведінки моделюють динамічні аспекти системи та охоплюють діаграми діяльності, станів, варіантів використання, а також діаграми взаємодії (послідовності, комунікації, огляду взаємодії та часові діаграми).

Для моделювання веб-застосунку електронної комерції особливо корисними є декілька типів UML-діаграм. Діаграма варіантів використання (Use Case Diagram) дозволяє визначити функціональні вимоги системи з точки зору взаємодії користувачів з системою. Як зазначає Мартін Фаулер у своїй роботі "UML Distilled", діаграми варіантів використання є ефективним інструментом для комунікації між розробниками та стейкхолдерами на ранніх

Діаграма класів (Class Diagram) є центральною для об'єктно-орієнтованого проектування та відображає структуру системи через класи, їх атрибути, методи та зв'язки між ними. За словами Крейга Лармана, автора книги

"Applying UML and Patterns", діаграма класів є найбільш широко використовуваною UML-діаграмою і служить основою для генерації коду та проектування бази даних.

Діаграми послідовності (Sequence Diagrams) моделюють взаємодію між об'єктами в часовому вимірі та є особливо корисними для документування складних бізнес-процесів, таких як оформлення замовлення або обробка платежу [18]. Діаграми станів (State Machine Diagrams) ефективно описують життєвий цикл об'єктів, що мають складну поведінку залежно від стану, наприклад, статуси замовлення або процес повернення товару.

Діаграма компонентів (Component Diagram) та діаграма розгортання (Deployment Diagram) використовуються для моделювання фізичної архітектури системи. Вони показують, як програмні компоненти розподіляються по серверах та як взаємодіють між собою, що є критично важливим для сучасних розподілених веб-застосунків.

Використання UML у процесі розробки програмного забезпечення забезпечує ряд переваг: покращення комунікації між членами команди, раннє виявлення архітектурних проблем, створення документації для подальшої підтримки системи та полегшення процесу впровадження нових розробників у проект. Дослідження, проведене Briand та інші, показало, що використання UML-моделювання на етапі проектування зменшує кількість дефектів у кінцевому продукті та підвищує якість програмного забезпечення.

3.2 Діаграма прецедентів

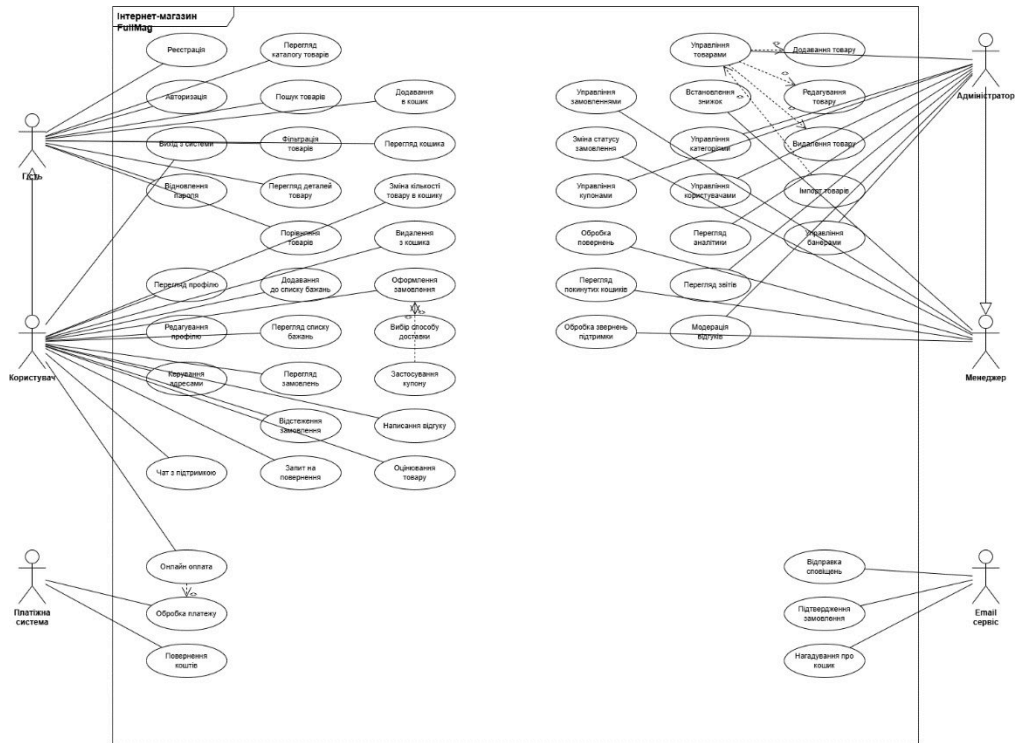


Рисунок 3.1. Діаграма перцедентів

Діаграма прецедентів (Use Case Diagram) (рис. 3.1) системи FullMag демонструє взаємодію шести основних акторів з функціоналом інтернет-магазину. Гість (неавторизований користувач) має доступ до базових функцій: перегляд каталогу та категорій товарів, використання фільтрів та сортування, пошук товарів, перегляд детальної інформації про товар та відгуків, додавання товарів до кошика та списку порівняння, а також реєстрацію та авторизацію в системі.

Авторизований користувач успадковує всі можливості гостя та отримує додаткові функції: оформлення замовлення з вибором способу доставки та оплати, управління особистим кабінетом (редагування профілю, адрес доставки), перегляд історії замовлень та їх статусів, написання відгуків на товари, управління списком бажань (wishlist), застосування промокодів та отримання email-сповіщень про статус замовлення.

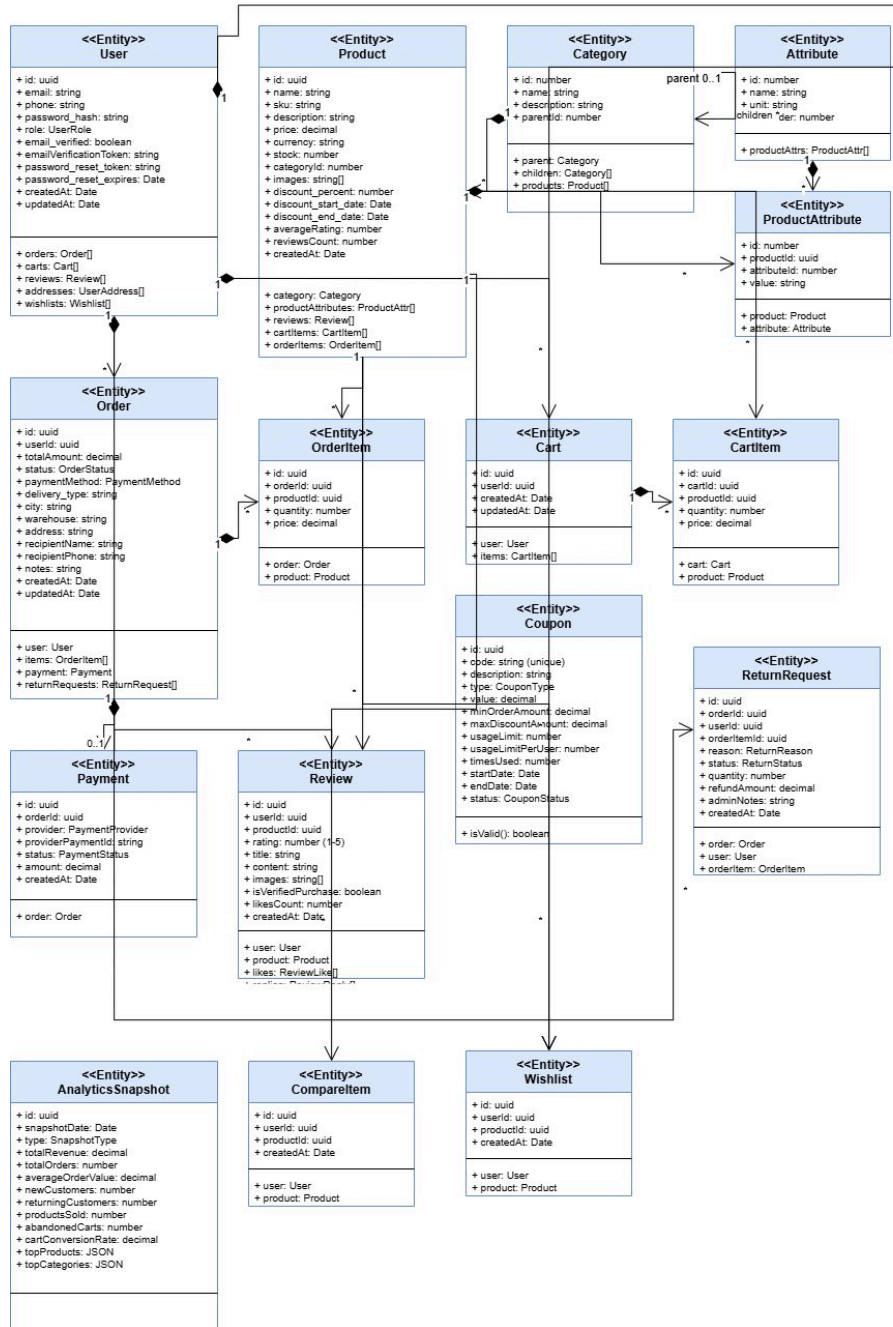
Менеджер відповідає за операційне управління: обробку та підтвердження замовлень, зміну статусів замовлень, перегляд аналітики продажів, управління

запитами на повернення товарів та комунікацію з клієнтами.

Адміністратор має повний доступ до системи управління: CRUD-операції з товарами та категоріями, управління характеристиками (атрибутами) товарів, модерація відгуків користувачів, управління обліковими записами та ролями, налаштування промокодів та знижок, імпорт товарів з CSV-файлів та перегляд повної аналітики.

Платіжна система як зовнішній актор забезпечує обробку онлайн-платежів через інтеграцію з платіжними шлюзами. Email-сервіс відповідає за автоматичне надсилання транзакційних листів: підтвердження реєстрації, сповіщення про замовлення, нагадування про покинутий кошик та інформування про зміну статусу замовлення.

3.2 Діаграма класів



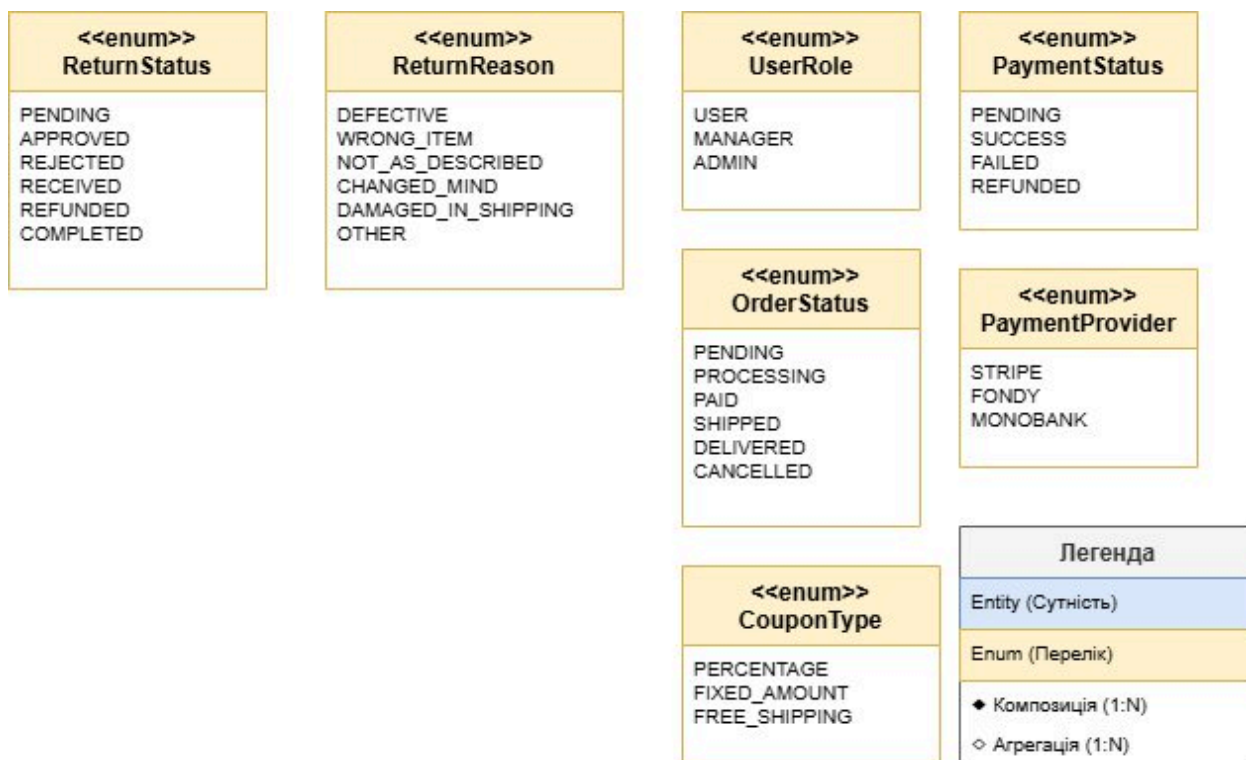


Рис. 3.2 - 3.3. Діаграма класів

Діаграма класів (Class Diagram) (рис. 3.2-3.3) відображає структуру даних серверної частини системи FullMag та взаємозв'язки між сутностями. Центральною сутністю є User (Користувач), яка містить поля: id, email, password (хешований), firstName, lastName, phone, role (enum: USER, MANAGER, ADMIN), isActive, createdAt та має зв'язки з усіма основними сутностями системи.

Product (Товар) є ключовою бізнес-сутністю з полями: id, name, slug, description, price, compareAtPrice (для відображення знижок), sku, quantity, isActive, images (масив зображень), createdAt, updatedAt. Товар пов'язаний з Category через many-to-one зв'язок, з Review, CartItem, OrderItem, WishlistItem та CompareItem через one-to-many зв'язки. Динамічні характеристики товару реалізовані через ProductAttribute, що пов'язує Product з Attribute.

Category (Категорія) підтримує ієрархічну структуру через self-referencing зв'язок parent-children та містить: id, name, slug, description, image, isActive, sortOrder.

Order (Замовлення) агрегує інформацію про покупку: id, orderNumber,

status (enum: PENDING, CONFIRMED, PROCESSING, SHIPPED, DELIVERED, CANCELLED, RETURNED), totalAmount, shippingAddress, paymentMethod, paymentStatus, notes, createdAt. Пов'язаний з User та містить колекцію OrderItem.

Cart (Кошик) реалізує тимчасове зберігання товарів перед оформленням замовлення через CartItem з полями quantity та прив'язкою до Product.

Payment (Платіж) фіксує транзакції: id, amount, status (enum: PENDING, COMPLETED, FAILED, REFUNDED), paymentMethod, transactionId, createdAt та пов'язаний з Order.

Review (Відгук) містить: id, rating (1-5), comment, isApproved, createdAt та пов'язує User з Product.

Coupon (Промокод) реалізує систему знижок: id, code, discountType (PERCENTAGE, FIXED), discountValue, minOrderAmount, maxUses, usedCount, startDate, endDate, isActive.

ReturnRequest (Запит на повернення) обробляє повернення товарів: id, reason, status (PENDING, APPROVED, REJECTED, COMPLETED), createdAt.

Допоміжні сутності включають: Wishlist та WishlistItem для списку бажань, CompareItem для порівняння товарів, AnalyticsSnapshot для збереження аналітичних даних, Attribute та ProductAttribute для динамічних характеристик товарів.

Query для роботи з API [7]. Mobile (@fullmag/mobile) — мобільний застосунок на React Native з Expo, що забезпечує кросплатформну підтримку iOS та Android, використовує React Navigation для навігації та спільну логіку з веб-застосунком.

Шар services містить основний бекенд-сервіс API (@fullmag/api), побудований на NestJS з модульною архітектурою [11]. Сервіс включає понад 20 функціональних модулів: Auth (JWT-аутентифікація, bcrypt-хешування), Users (управління користувачами та ролями), Products (каталог товарів з фільтрацією), Categories (ієрархічна структура категорій), Cart (кошик покупок), Orders (обробка замовлень), Payments (інтеграція з платіжними системами), Reviews (система відгуків), Wishlist (список бажань), Compare (порівняння товарів), Coupons (промокоди та знижки), Returns (обробка повернень), Analytics (збір та аналіз статистики), Import (імпорт товарів з CSV), Email (транзакційні листи через nodemailer), SMS (SMS-сповіщення), Telegram (бот-сповіщення), Upload (завантаження зображень), Attributes (динамічні характеристики товарів), Addresses (адреси доставки).

Шар packages містить спільну бібліотеку Common (@fullmag/common), яка включає: TypeScript типи та інтерфейси [12], що використовуються як на клієнті, так і на сервері; утиліти форматування (ціни, дати); константи (статуси замовлень, ролі користувачів); валідаційні схеми.

Шар infrastructure забезпечує розгортання та роботу системи: PostgreSQL (основна реляційна база даних) [15], Redis (кешування та сесії), MinIO/S3 (зберігання зображень), Docker та docker-compose (контейнеризація), Nginx (reverse проху та статика).

Залежності між шарами: apps залежить від packages та services; services залежить від packages та infrastructure; packages є незалежним шаром спільного коду.

3.5 Огляд застосунку

На рис. 3.5 зображено екрани створення облікового запису та логіну, які демонструють інтерфейс реєстрації нового користувача та входу до існуючого

облікового запису. Форма реєстрації містить поля для введення імені, електронної пошти та пароля з підтвердженням. Форма входу включає поля email та пароль.

Create your account
Already have an account? [Sign in](#)

Email

Phone (optional)

Password

Confirm Password

Sign Up

Sign in to your account
Or [create a new account](#)

Email

Пароль

Увійти

Рис. 3.5. Екрани створення облікового запису та логіну

Головний екран магазину (рис 3.6) демонструє основну сторінку веб-застосунку з каталогом товарів. Верхня частина містить навігаційну панель з логотипом, пошуком, посиланнями на категорії та іконками кошика і профілю. Основна область відображає сітку карток товарів з зображеннями, назвами, цінами та кнопками додавання в кошик. Бічна панель містить фільтри за категоріями, ціною та характеристиками.

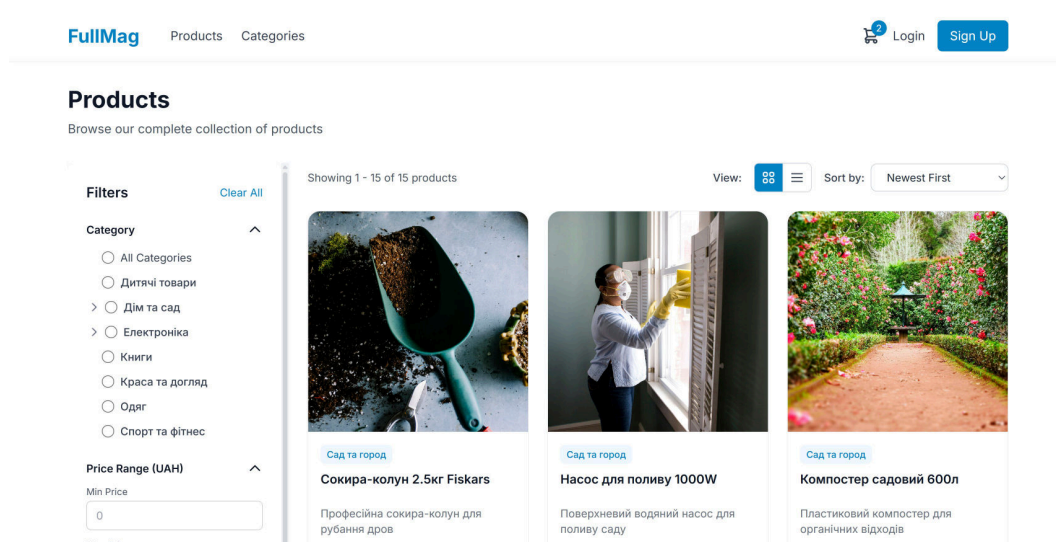


Рис. 3.6. Головний екран магазину

На рис. 3.7 показано функціональність фільтрації товарів у каталозі. Ліва панель відображає активні фільтри: обрані категорії, ціновий діапазон через повзунок, наявність на складі, характеристики товарів. При застосуванні фільтрів каталог миттєво оновлюється, показуючи лише відповідні товари.

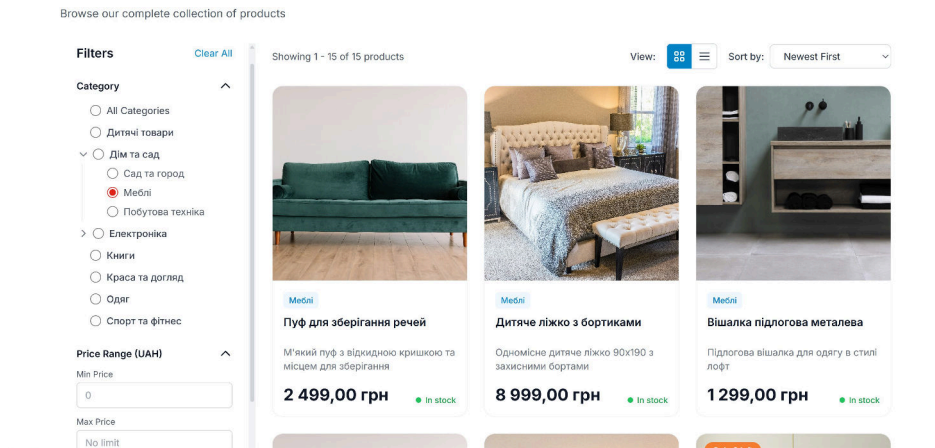


Рис. 3.7. Застосування фільтрів

На екрані показано альтернативний спосіб відображення товарів у вигляді карткового лейауту (рис. 3.8). Кожен товар представлено окремою карткою, що містить зображення продукту, його назву, ціну та кнопку додавання до кошика. Картковий вигляд забезпечує більш наочне представлення товарів та зручну навігацію. Користувач може перемикатися між сітковим та картковим виглядом за допомогою відповідних іконок у верхній частині каталогу.

Showing 1 - 15 of 15 products

View:



Sort by:

Newest First



МЕБЛІ

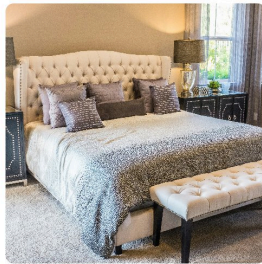
Пуф для зберігання речей

М'який пуф з відкидною кришкою та місцем для зберігання

In Stock (25 available)

2 499,00 грн

Add to Cart



МЕБЛІ

Дитяче ліжко з бортиками

Одномісне дитяче ліжко 90x190 з захисними бортами

In Stock (12 available)

8 999,00 грн

Add to Cart



МЕБЛІ

Рис. 3.8. Застосування карткового лейауту

Картка товару (рис. 3.9) відображає детальну інформацію про окремий продукт у компактному форматі. На картці розміщено головне зображення товару, його назву, ціну (включаючи стару ціну при наявності знижки), індикатор наявності на складі. Присутня кнопка для швидкого додавання товару до кошика. Дизайн картки оптимізовано для зручного сприйняття інформації та швидкого прийняття рішення про покупку.

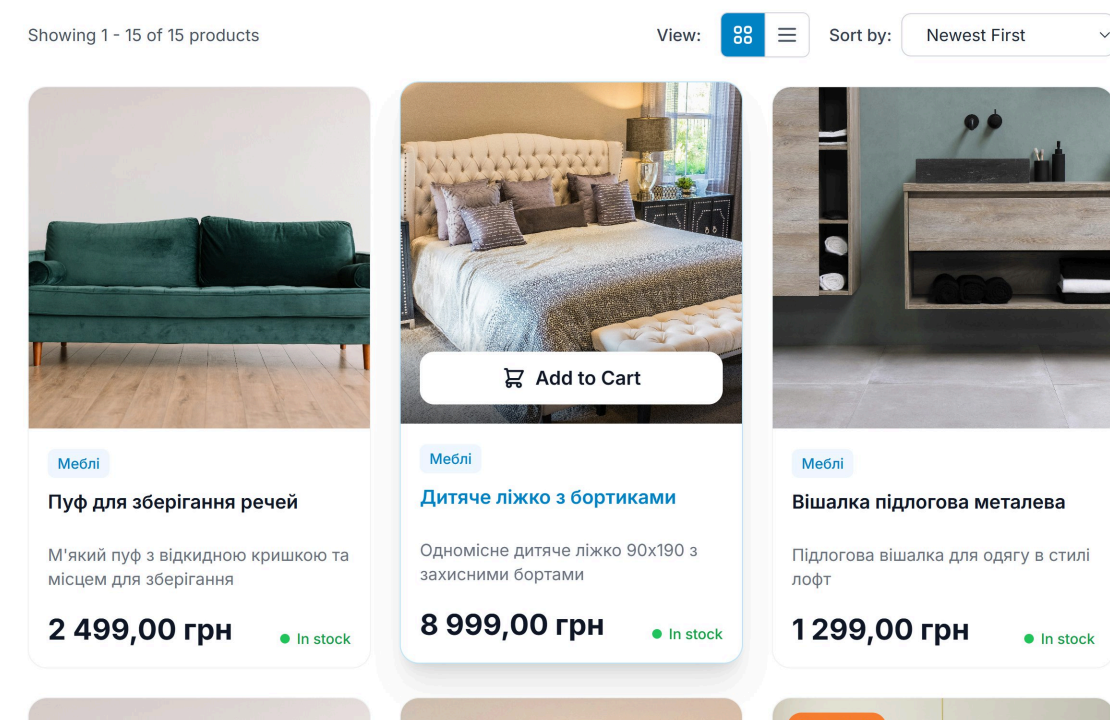


Рис. 3.9. Картка товару

Екран демонструє функціонал сортування товарів у каталозі (рис. 3.10). Випадаюче меню дозволяє обрати критерій сортування: за ціною (від низької до високої або навпаки), за новизною, за назвою тощо. Обраний параметр сортування застосовується до всього списку товарів у поточній категорії. Це дозволяє користувачам швидко знаходити товари відповідно до їхніх пріоритетів.

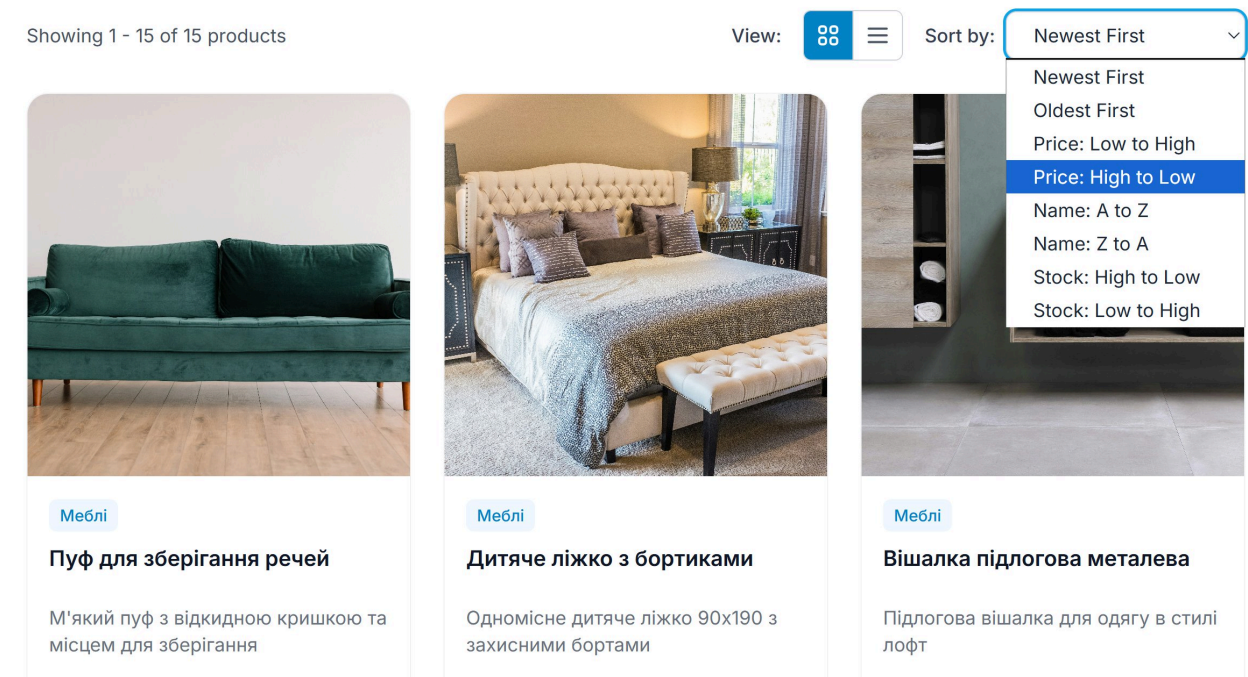


Рис. 3.10. Сортування товарів

Сторінка товару містить повну інформацію про обраний продукт (рис. 3.11). У верхній частині розміщено галерею зображень з можливістю збільшення та перегляду з різних ракурсів. Праворуч знаходиться блок з назвою товару, ціною, вибором кількості та кнопкою додавання до кошика. Нижче представлено детальний опис товару, технічні характеристики у вигляді таблиці (рис. 3.12), відгуки покупців з оцінками та блок рекомендованих супутніх товарів.

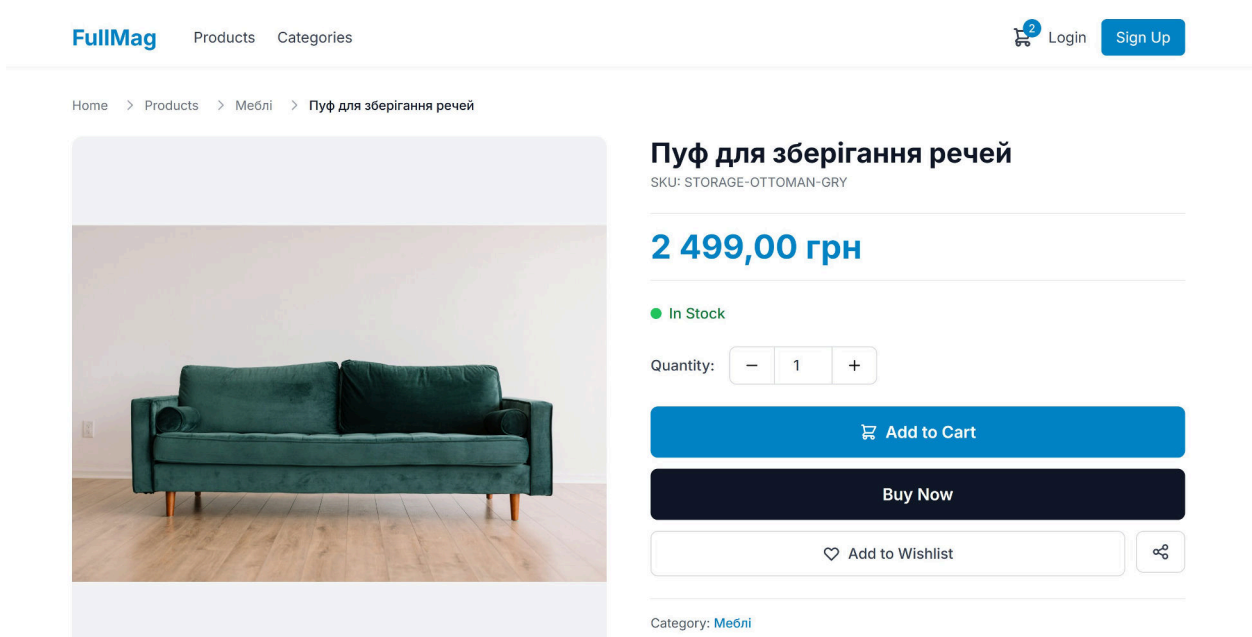


Рис. 3.11. Сторінка товару

Description		Specifications	12	Reviews
Brand	Samsung			
Color	Black,White,Violet,Lime			
Warranty Period	24 months			
Condition	New			
Screen Size	6.4			
RAM	8GB			
Storage	128GB			
Battery Capacity	5000			
Camera Resolution	50MP			
5G Support	Yes			
Display Type	AMOLED			
Operating System	Android 13			





Рис. 3.12. Таблиця характеристик товару

Сторінка кошику товарів (рис. 3.13) відображає список обраних користувачем товарів для покупки. Для кожного товару показано зображення, назву, ціну та можливість змінити кількість або видалити товар. У нижній частині сторінки розміщено підсумкову вартість замовлення та кнопку переходу до оформлення замовлення.

FullMag Products Categories Dashboard Store Sales Customers Analytics admin@fullmag.com Logout

Кошик

2 товари (2 шт.) [Очистити кошик](#)

 <p>Пазл Ravensburger 1000 елементів 749,00 грн за шт.</p>	<p>– 1 + Макс: 35</p>	<p>749,00 грн</p>	
 <p>Дитячий планшет LeapPad 3 999,00 грн за шт.</p>	<p>– 1 + Макс: 18</p>	<p>3 999,00 грн</p>	

Підсумок замовлення


Товарів (2 шт.) 4 748,00 грн

Доставка **Безкоштовно**

Всього 4 748,00 грн

[Оформити замовлення](#)

[Продовжити покупки](#)

 Безпечна оплата

<p>FullMag</p> <p>Multi-platform e-commerce system</p>	<p>Shop</p> <p>All Products</p> <p>Categories</p>	<p>Account</p> <p>My Profile</p> <p>Order History</p>	<p>Contact</p> <p>Email: support@fullmag.com</p> <p>Phone: +380 50 123 4567</p>
---	--	--	--

© 2024 FullMag. All rights reserved.

Рис. 3.13. Сторінка кошику товарів

Сторінка облікового запису користувача (рис. 3.14) містить персональну інформацію: ім'я, електронну пошту, номер телефону, адреси доставки. Користувач може редагувати свої дані, змінювати пароль, управляти налаштуваннями сповіщень. Також відображаються посилання на історію замовлень, список бажань.

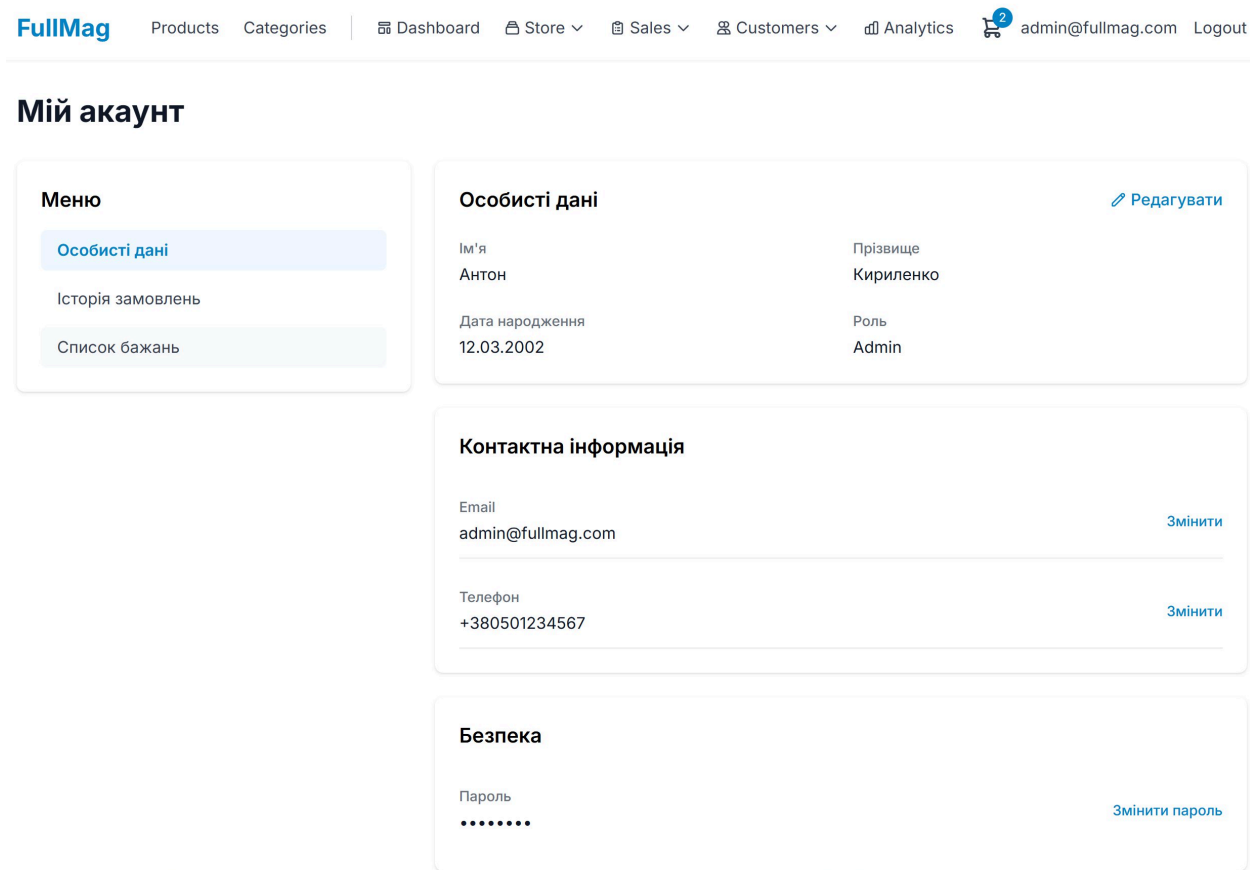


Рис. 3.14. Сторінка облікового запису користувача

Сторінка історії замовлень (рис. 3.15) відображає перелік усіх замовлень користувача в хронологічному порядку. Кожен запис містить номер замовлення, дату, статус, загальну суму та кнопку для перегляду детальної інформації. Передбачено фільтрацію замовлень за статусом та періодом.

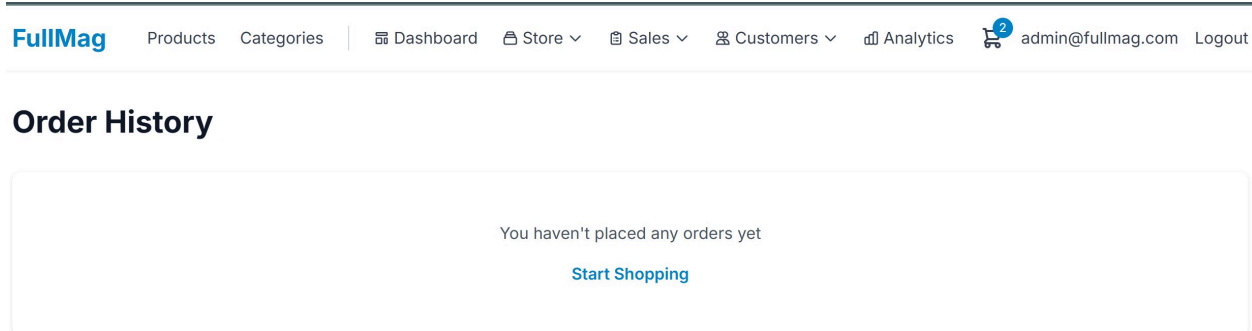


Рис. 3.15. Сторінка історії замовлень

Сторінка аналітики для адміністраторів (рис. 3.16) надає комплексну інформацію про роботу магазину. Відображаються графіки продажів за період, діаграми розподілу замовлень за статусами, топ продаваних товарів, дані про конверсію та середній чек. Інформація оновлюється в режимі реального часу.

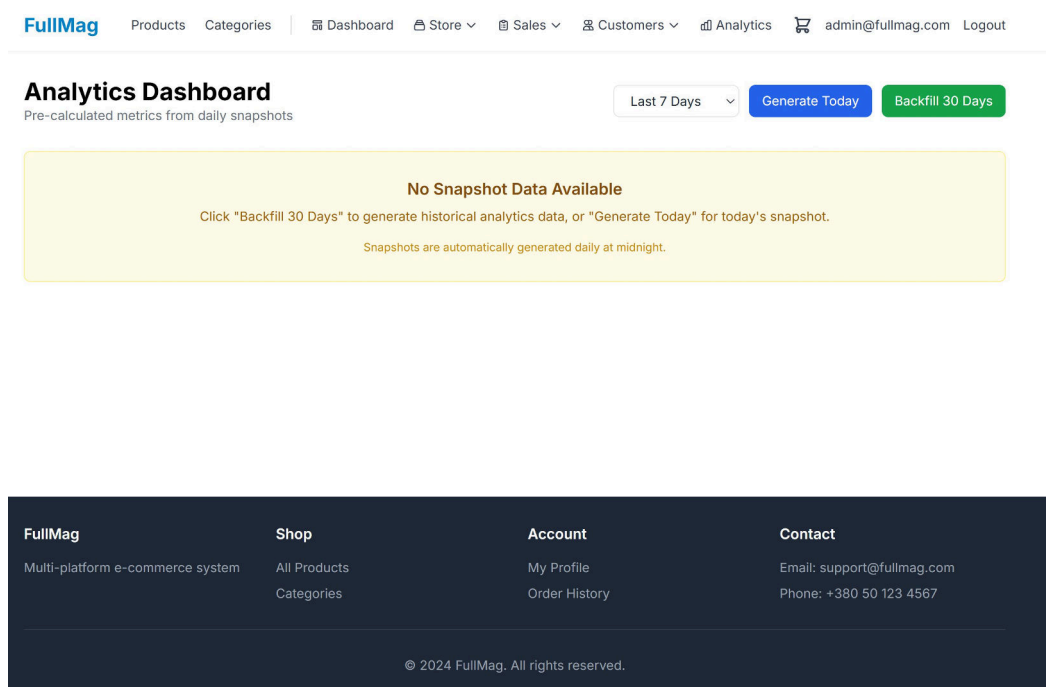


Рис. 3.16. Сторінка аналітики для адміністраторів

Сторінка керування користувачами (рис. 3.17) дозволяє адміністраторам

переглядати та управляти обліковими записами клієнтів. Таблиця містить інформацію про кожного користувача: ім'я, email, дату реєстрації, кількість замовлень, статус акаунта. Доступні функції пошуку, фільтрації, блокування та редагування профілів.

User Management Refresh

Total Users: 3 | Admins: 1 | Managers: 1 | Regular Users: 1 | Verified: 0 | Unverified: 3

Search: Role Filter: Search

USER	ROLE	STATUS	ORDERS	JOINED	ACTIONS
admin@fullmag.com +380501234567	Admin	Unverified	0	12/2/2025	
manager@fullmag.com +380501234568	Manager	Unverified	0	12/2/2025	Delete
user@fullmag.com +380501234569	User	Unverified	0	12/2/2025	Delete

Showing 3 users

FullMag | Shop | Account | Contact
Multi-platform e-commerce system | All Products | My Profile | Email: support@fullmag.com

Рис. 3.17. Сторінка керування користувачами для адміністраторів

Сторінка керування товарами (рис. 3.18) надає інструменти для адміністрування каталогу продукції. Таблиця товарів містить зображення, назву, ціну, кількість на складі, категорію та статус. Передбачено можливості масового редагування, експорту та імпорту товарів, налаштування знижок та акцій.

Product Management

Manage your product catalog and inventory

[Hide Charts](#)

Total Products

0

Inventory Value

0,00 грн

Avg Price

0,00 грн

In Stock

0

Low Stock

0

Out of Stock

0

[Import CSV](#)[+ Add Product](#)

PRODUCT	SKU	CATEGORY	PRICE	STOCK	ACTIONS
Дитячий планшет LeapPad	LEAPFROG-LEAP-PAD3	Дитячі товари	3 999,00 грн	18	Edit Delete
Музичний килимок для танців	DANCE-MAT-MUSIC-USB	Дитячі товари	1 299,00 грн	25	Edit Delete
Ляльковий театр дерев'яний	PUPPET-THEATER-WOOD	Дитячі товари	1 899,00 грн	20	Edit Delete
Пазл Ravensburger 1000 елементів	RAVENSBURGER-1000-LAND	Дитячі товари	749,00 грн	35	Edit Delete
Настільна гра Монополія класична	MONOPOLY-CLASSIC-UA	Дитячі товари	899,00 грн	50	Edit Delete
Конструктор Magformers 62 елементи	MAGFORMERS-62-STAND	Дитячі товари	2 999,00 грн	30	Edit Delete
Електромобіль дитячий Mercedes-Benz	KID-CAR-MERC-G63-BLK	Дитячі товари	12 999,00 грн	10	Edit Delete

Рис. 3.18. Сторінка керування товарами для адміністраторів

Сторінка керування категоріями (рис. 3.19) відображає ієрархічну структуру каталогу товарів. Адміністратор може створювати, редагувати та видаляти категорії, налаштовувати їх порядок відображення, призначати зображення та SEO-метадані. Деревоподібна структура дозволяє зручно керувати вкладеними підкатегоріями.

Category Management

Organize your product categories and attributes

Manage Attributes

+ Add Category

Total Categories 17	Root Categories 7	Total Attributes 18
-------------------------------	-----------------------------	-------------------------------

Categories		Expand All	Collapse All
Дитячі товари Категорія Дитячі товари	18 attributes	Edit	Add Sub Delete
> Дім та сад (3 subcategories) Категорія Дім та сад	18 attributes	Edit	Add Sub Delete
> Електроніка (7 subcategories) Категорія Електроніка	18 attributes	Edit	Add Sub Delete
Книги Категорія Книги	18 attributes	Edit	Add Sub Delete
Краса та догляд Категорія Краса та догляд	18 attributes	Edit	Add Sub Delete
Одяг Категорія Одяг	18 attributes	Edit	Add Sub Delete
Спорт та фітнес Категорія Спорт та фітнес	18 attributes	Edit	Add Sub Delete

Рис. 3.19. Сторінка керування категоріями для адміністраторів

Сторінка керування відгуками (рис. 3.20) дозволяє модерувати коментарі та оцінки користувачів. Таблиця містить текст відгуку, рейтинг, дані автора, товар та дату публікації. Адміністратор може схвалювати, редагувати або видаляти відгуки, а також відповідати на них від імені магазину.

FullMag Products Categories | Dashboard Store Sales Customers Analytics admin@fullmag.com Logout

Reviews Management

Refresh

Total Reviews 0	Visible 0	Hidden 0	With Complaints 0	Pending Complaints 0
---------------------------	---------------------	--------------------	-----------------------------	--------------------------------

All Reviews Complaints

Search

Content, user, product... Only show reviews with complaints

REVIEW	PRODUCT	RATING	STATUS	COMPLAINTS	ACTIONS
No reviews found					

Showing 0 reviews

FullMag Multi-platform e-commerce system	Shop All Products Categories	Account My Profile Order History	Contact Email: support@fullmag.com Phone: +380 50 123 4567
--	---	---	---

Рис. 3.20. Сторінка керування відгуками для адміністраторів

Сторінка керування характеристиками товарів (рис. 3.21) дозволяє налаштовувати динамічні атрибути для різних категорій товарів. Адміністратор може створювати нові типи характеристик, визначати їх тип даних, налаштовувати одиниці вимірювання та відображення в фільтрах каталогу.

The screenshot shows the 'Attribute Management' interface. At the top, there are navigation links for 'FullMag', 'Products', 'Categories', 'Dashboard', 'Store', 'Sales', 'Customers', 'Analytics', and a user profile 'admin@fullmag.com' with a 'Logout' option. The main heading is 'Attribute Management' with a subtitle 'Define product attributes for filtering and specifications'. There are two buttons: 'Back to Categories' and '+ Add Attribute'. Below this, there are three summary boxes: 'Total Attributes: 18', 'Global Attributes: 18', and 'Category-Specific: 0'. A dropdown menu is set to 'All Attributes'. The main part of the page is a table with the following data:

ATTRIBUTE	TYPE	CATEGORY	FILTERABLE	REQUIRED	ACTIONS
Brand slug: brand	Single Select Dropdown 12 options	Global	✓	-	Edit Delete
Color slug: color	Multi Select Checkbox 10 options	Global	✓	-	Edit Delete
Warranty Period slug: warranty Unit: months	Number Number Input	Global	✓	-	Edit Delete
Condition slug: condition	Single Select Radio Buttons 3 options	Global	✓	✓	Edit Delete
Screen Size slug: screen-size Unit: inches	Range Range Slider	Global	✓	-	Edit Delete

Рис. 3.21. Сторінка керування характеристиками товарів для адміністраторів

Сторінка імпорту товарів (рис. 3.22) надає інтерфейс для масового завантаження товарів з CSV-файлів. Користувач може завантажити файл, налаштувати відповідність колонок полям системи, переглянути попередній результат імпорту та запустити процес обробки. Відображається прогрес та звіт про успішно імпортовані та помилкові записи.

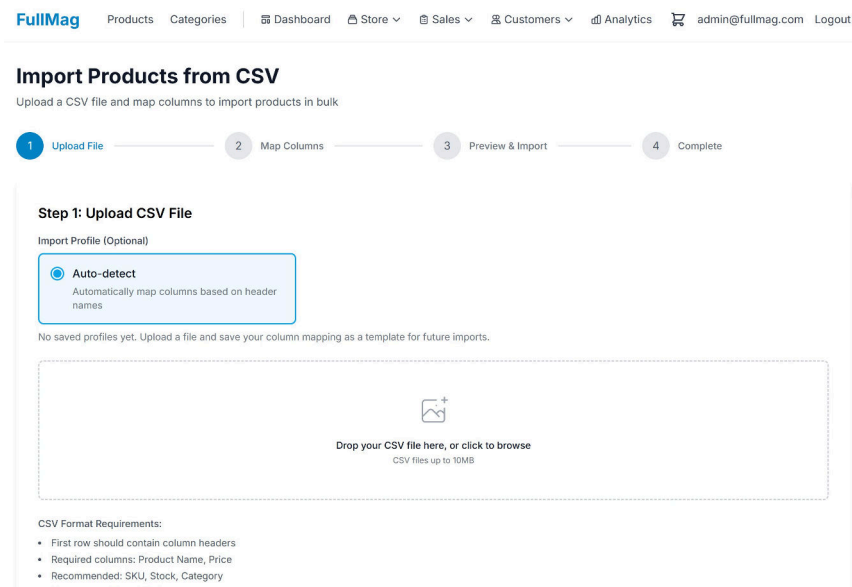


Рис. 3.22. Сторінка імпорту товарів

Вікно чату (рис. 3.23) надає можливість користувачеві встановити зв'язок із менеджерами-асистентами та отримати необхідну допомогу.

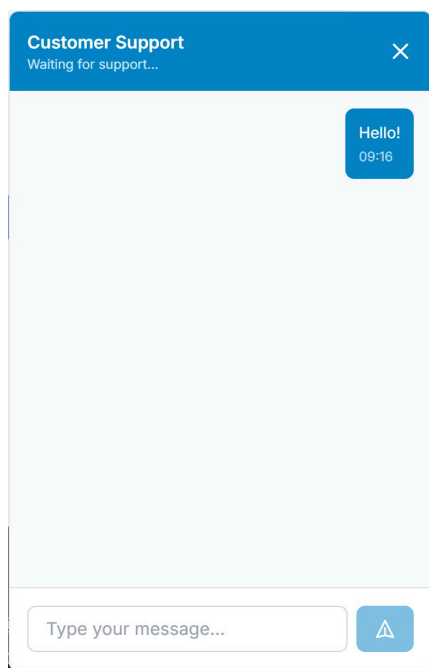


Рис. 3.23. Вікно чату

Головний екран мобільного додатку (рис. 3.24) адаптовано для зручного використання на смартфонах. У верхній частині розміщено логотип магазину та іконку пошуку. Нижче знаходиться горизонтальна карусель з банерами акцій. Основну частину екрану займає вертикальний список категорій товарів з

іконками. У нижній частині розташована панель навігації.

Головний екран мобільного додатку адаптовано для зручного використання на смартфонах. У верхній частині розміщено логотип магазину та іконку пошуку. Нижче знаходиться горизонтальна карусель з банерами акцій. Основну частину екрану займає вертикальний список категорій товарів з іконками. У нижній частині розташована панель навігації з іконками головної сторінки, каталогу, кошика, списку бажань та профілю користувача.

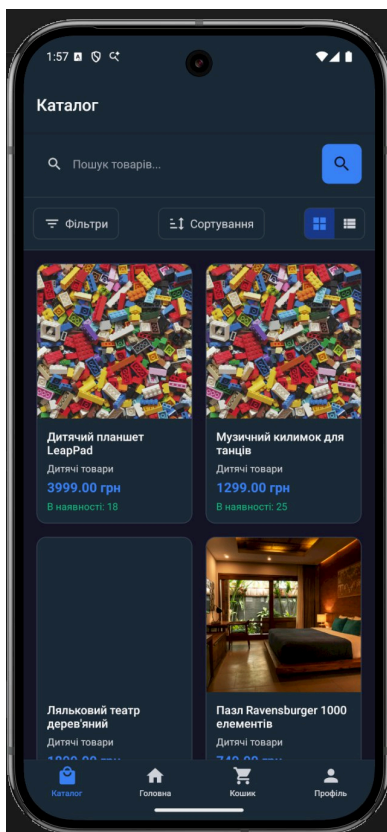


Рис. 3.24. Головний екран мобільного додатку

Екран сортування товарів у мобільному додатку (рис. 3.25) представлено у вигляді модального вікна, що з'являється при натисканні на відповідну кнопку. Список опцій сортування включає: за популярністю, за ціною зростання, за ціною спадання, за новизною та за рейтингом. Обрана опція виділяється кольором або галочкою. Інтерфейс оптимізовано для зручного вибору пальцем на сенсорному екрані.

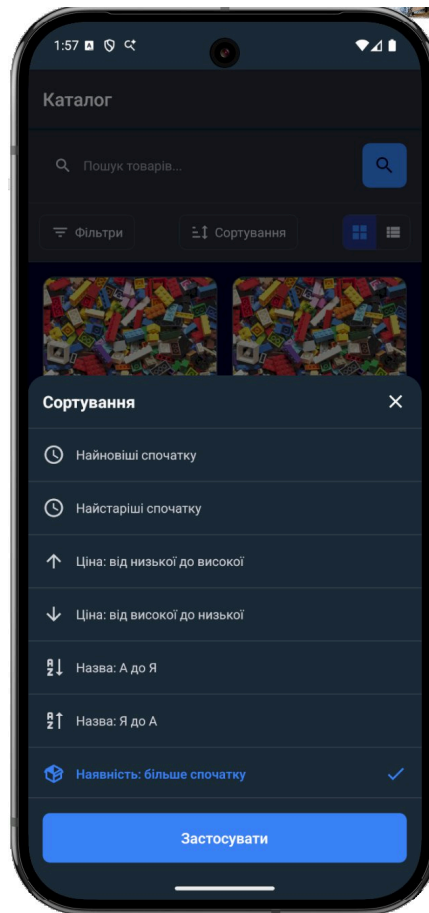


Рис. 3.25. Сортування товарів в мобільному додатку

Екран фільтрації товарів у мобільному додатку (рис. 3.26) відображає повноекранну панель з усіма доступними фільтрами. Фільтри згруповано за категоріями: ціновий діапазон (з повзунками), бренд (з чекбоксами), колір, розмір та інші атрибути. У нижній частині екрану розміщено кнопки "Скинути" для очищення всіх фільтрів та "Застосувати" для відображення відфільтрованих результатів.

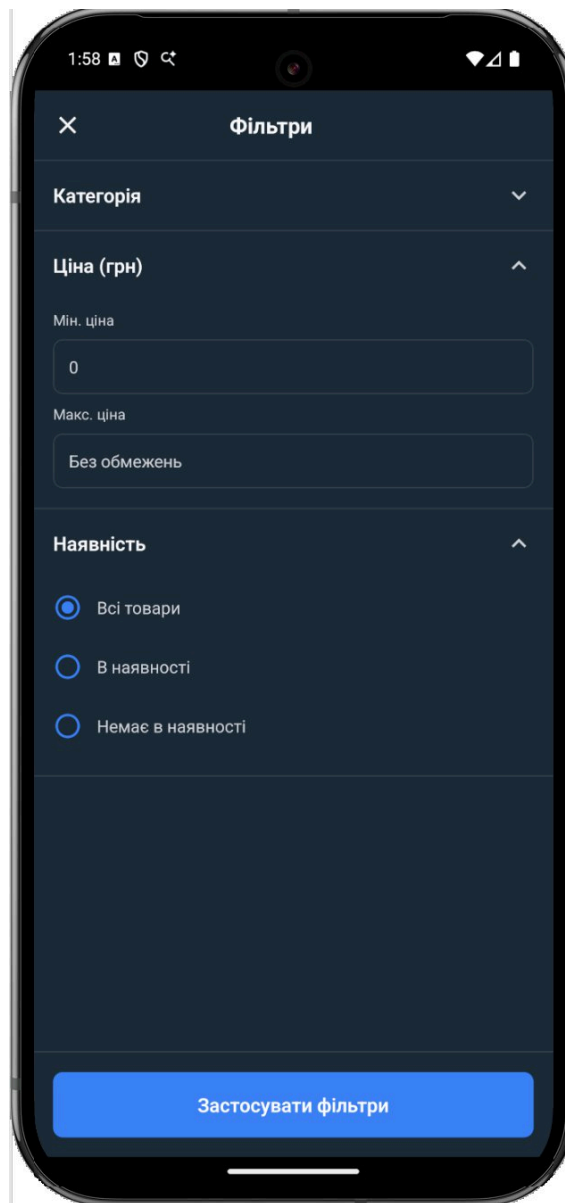


Рис. 3.26. Фільтрація товарів в мобільному додатку

Сторінка контактів у мобільному додатку (рис. 3.27) містить всю необхідну інформацію для зв'язку з магазином. Відображено адресу фізичного магазину (за наявності), номери телефонів з можливістю прямого дзвінка при натисканні, електронну пошту та посилання на соціальні мережі. Також присутня інтерактивна карта з розташуванням магазину та графік роботи служби підтримки.

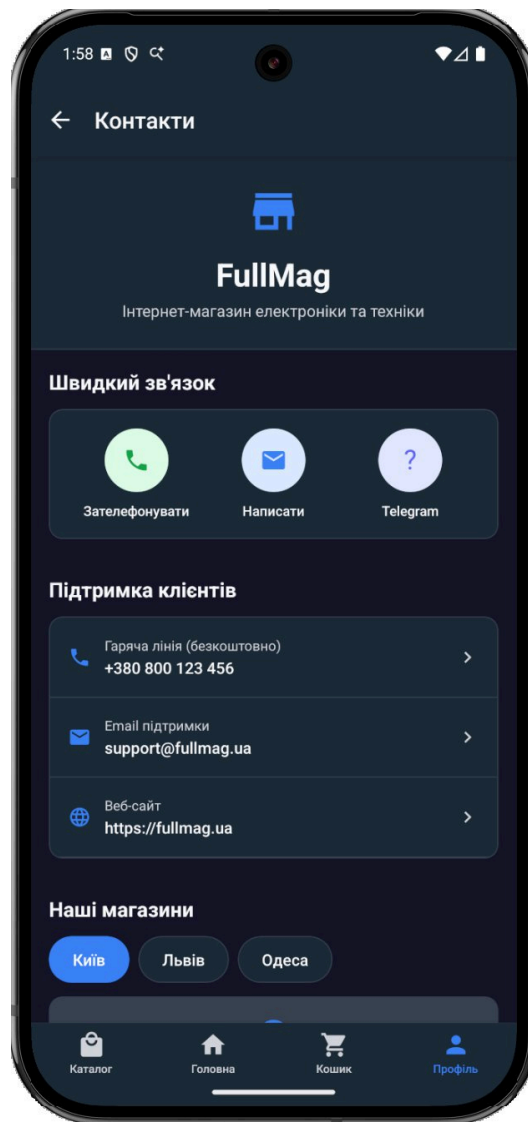


Рис. 3.27. Сторінка контактів в мобільному додатку

Додатковий вигляд сторінки контактів (рис. 3.28) демонструє альтернативне оформлення розділу зв'язку з магазином. Інтерфейс містить форму зворотного зв'язку для надсилання повідомлень безпосередньо з додатку, FAQ-секцію з відповідями на поширені запитання та можливість розпочати чат з оператором підтримки в режимі реального часу.

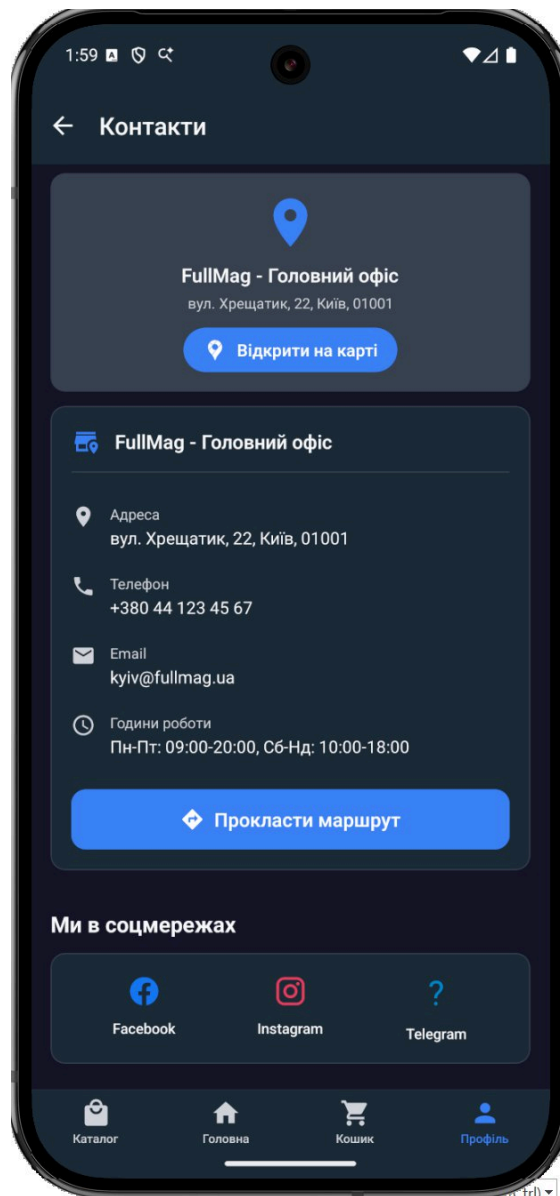


Рис. 3.28. Сторінка контактів в мобільному додатку

Сторінка налаштувань у мобільному додатку (рис. 3.29) дозволяє користувачу персоналізувати роботу застосунку. Доступні опції включають: вибір мови інтерфейсу, налаштування push-сповіщень (про акції, статус замовлення, нові надходження), вибір теми оформлення (світла/темна), налаштування конфіденційності та можливість очищення кешу. Також присутні посилання на політику конфіденційності та умови використання.

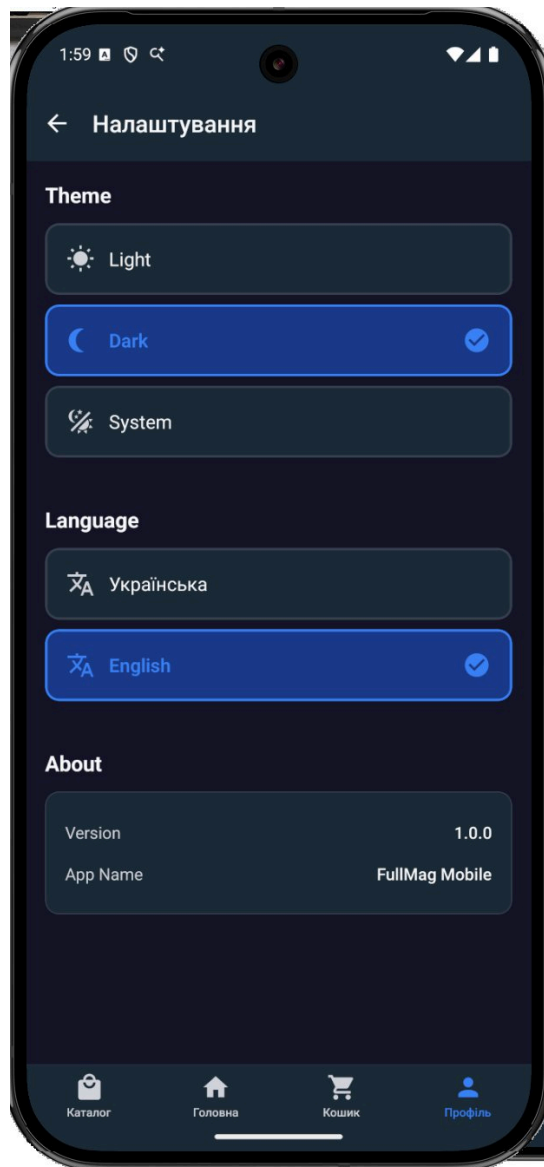


Рис. 3.29. Сторінка налаштувань в мобільному додатку

Сторінка авторизації у мобільному додатку (рис. 3.30) містить форму входу до облікового запису. Представлено текстові поля для введення електронної пошти (або номера телефону) та пароля. Нижче розміщено кнопку "Увійти" та посилання "Забули пароль?" для відновлення доступу. Також доступні альтернативні способи авторизації через соціальні мережі (Google, Facebook) та кнопка переходу до реєстрації нового користувача.



Рис. 3.30. Сторінка авторизації в мобільному додатку

Електронний лист зі сповіщенням про залишені товари у кошику (рис. 3.31) демонструє шаблон email-повідомлення, яке надсилається користувачам для нагадування про незавершене замовлення. Лист містить логотип магазину, персоналізоване привітання, список товарів із зображеннями та цінами, загальну суму кошика, яскраву кнопку для повернення до оформлення замовлення та контактну інформацію магазину.

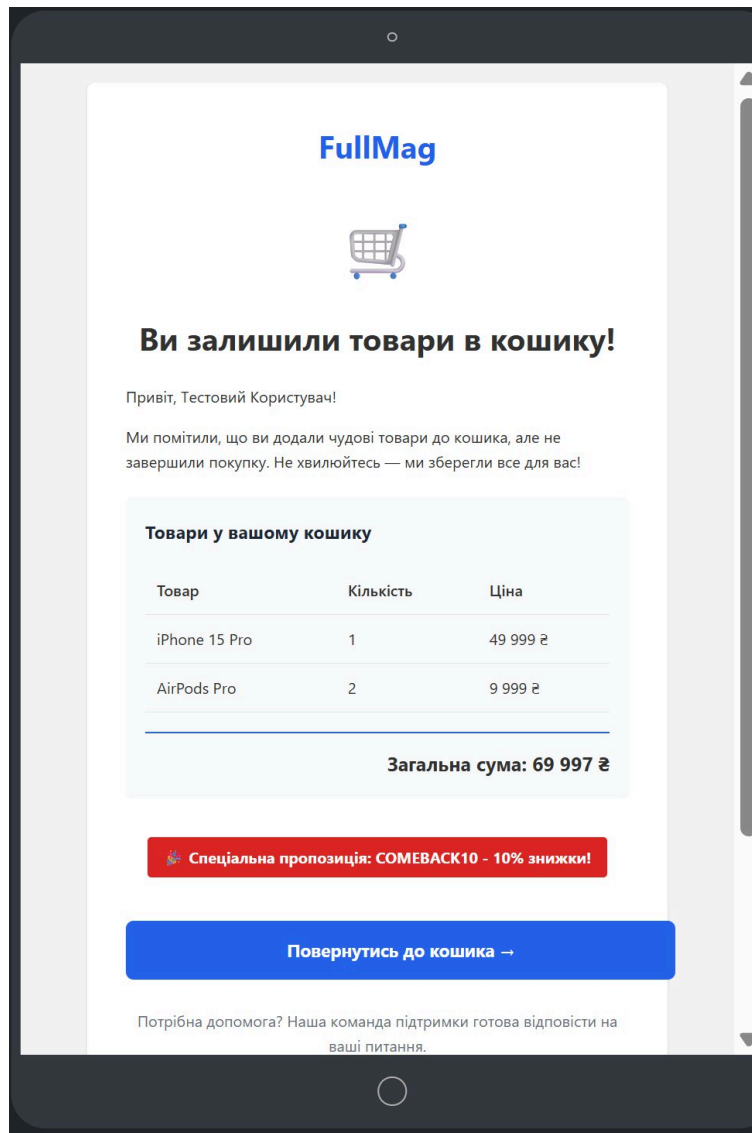


Рис. 3.31. Електронний лист зі сповіщенням про оновлення товарів у кошику

Електронний лист зі сповіщенням про залишені товари у кошику демонструє шаблон email-повідомлення, яке надсилається користувачам для нагадування про незавершене замовлення. Лист містить логотип магазину, персоналізоване звернення до клієнта, список товарів у кошику із зображеннями та цінами, загальну суму та помітну кнопку заклику до дії "Завершити покупку". Також може включати промокод на знижку для стимулювання завершення замовлення.

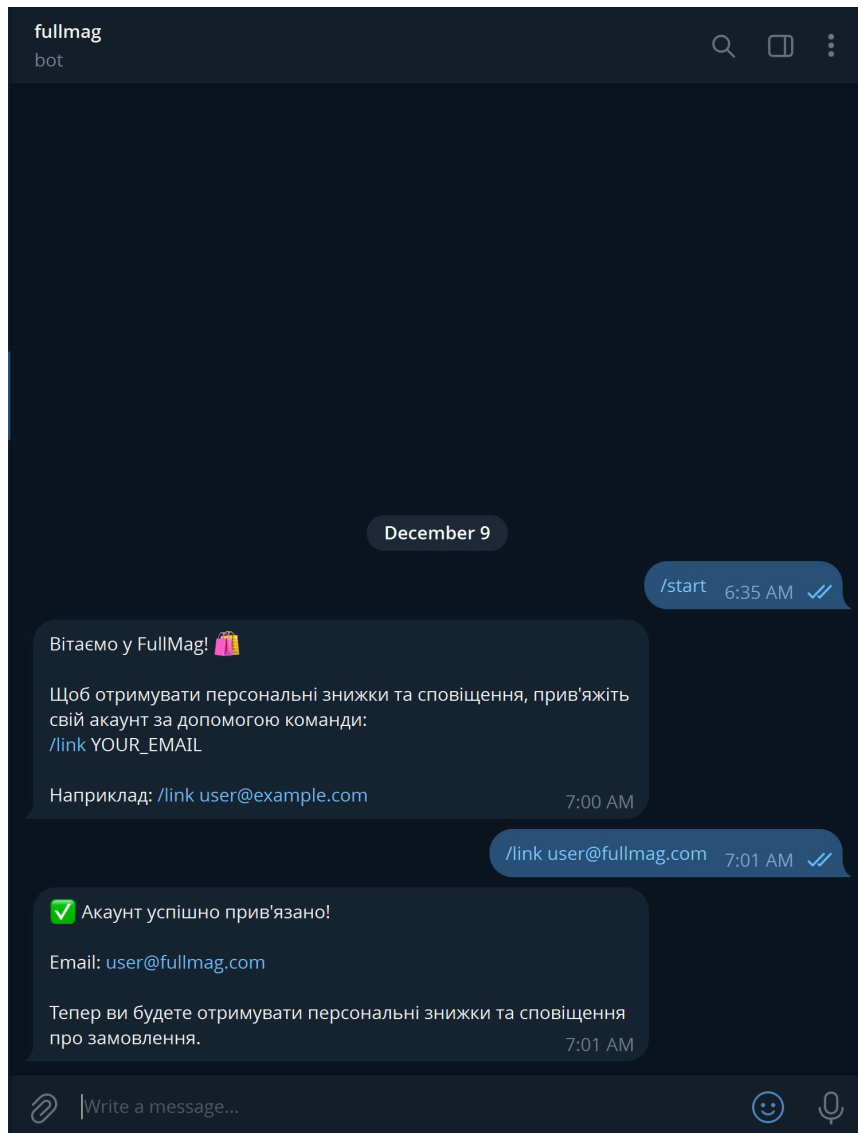


Рис. 3.32. Чат-бот магазину в Telegram

3.6 Вимоги до апаратного і програмного забезпечення

Веб-клієнт.

Мінімальні вимоги до апаратного забезпечення:

- процесор: частота не нижче 2 ГГц;
- оперативна пам'ять: не менше 4 ГБ;
- відеоадаптер і монітор: роздільна здатність 1920 × 1080;
- вільна пам'ять на жорсткому диску: не менше 100 МБ.

Вимоги до програмного забезпечення:

- операційна система: Windows 7+, macOS 10.14+, Linux;

- браузер: Google Chrome 90+, Mozilla Firefox 90+, Safari 14+, Microsoft Edge 90+;
- JavaScript [6]: увімкнено в браузері;
- WebSocket: підтримка протоколу.

Мобільний застосунок.

Мінімальні вимоги до апаратного забезпечення:

- процесор: ARM64 або x86_64;
- оперативна пам'ять: не менше 3 ГБ;
- вільна пам'ять: не менше 150 МБ;
- дисплей: роздільна здатність 720×1280 або вище;
- мережа: Wi-Fi або мобільний інтернет (3G/4G/5G).

Вимоги до програмного забезпечення:

- Android: версія 7.0 (API 24) або вище;
- iOS: версія 13.0 або вище.

3.7 Проктування можливих вдосколанень системи та подальших ітерацій

Вдосконалення пошукового алгоритму з використанням Elasticsearch [17].

Поточна реалізація пошуку товарів у системі базується на SQL-запитах з використанням оператора LIKE, що забезпечує базову функціональність, але має суттєві обмеження при масштабуванні каталогу та зростанні вимог до релевантності результатів. Для вдосконалення пошукового механізму доцільним є впровадження Elasticsearch - розподіленого пошукового та аналітичного рушія [17], побудованого на базі бібліотеки Apache Lucene.

Elasticsearch реалізує концепцію повнотекстового пошуку (full-text search), яка принципово відрізняється від традиційного пошуку в реляційних базах даних. Замість послідовного сканування записів, Elasticsearch використовує інвертований індекс (inverted index) - структуру даних [15], що зберігає відображення від термінів до документів, які їх містять. При індексації тексту відбувається його токенізація, нормалізація (приведення до нижнього регістру, видалення діакритичних знаків), стемінг (приведення слів до основи)

та видалення стоп-слів.

Для інтернет-магазину електроніки Elasticsearch надає ряд критичних переваг [17]. По-перше, це нечіткий пошук (fuzzy search), який дозволяє знаходити товари навіть при помилках у написанні запиту, використовуючи алгоритм відстані Левенштейна. По-друге, механізм автодоповнення (autocomplete) через edge n-grams забезпечує миттєві підказки при введенні пошукового запиту. По-третє, фасетний пошук (faceted search) дозволяє динамічно формувати фільтри на основі агрегацій, показуючи кількість товарів для кожного значення атрибута.

Архітектурно інтеграція Elasticsearch передбачає синхронізацію даних між PostgreSQL та пошуковим індексом [15]. Оптимальним підходом є використання патерну Change Data Capture (CDC) через Debezium, який відстежує зміни в базі даних та автоматично оновлює індекс Elasticsearch, забезпечуючи eventual consistency без навантаження на основну базу даних.

Вдосконалення алгоритму збору та аналізу статистики.

Поточна система збирає базову статистику продажів та користувачів, однак для прийняття обґрунтованих бізнес-рішень необхідна більш глибока аналітика. Існує два основних підходи до вирішення цієї задачі: інтеграція з зовнішніми аналітичними платформами або розробка власного аналітичного модуля.

Для e-commerce систем малого та середнього масштабу оптимальним рішенням є гібридний підхід. Збір даних про поведінку користувачів (перегляди товарів, додавання в кошик, конверсії) доцільно реалізувати через інтеграцію з Google Analytics 4, який надає безкоштовний функціонал для відстеження e-commerce подій та побудови воронки конверсії. Для транзакційної аналітики (фінансові звіти, аналіз замовлень, LTV клієнтів) рекомендується розробка власного модуля з використанням архітектурного патерну CQRS (Command Query Responsibility Segregation).

CQRS передбачає розділення моделей для запису та читання даних. Для аналітичних запитів створюється окрема денормалізована модель даних,

оптимізована для агрегацій та звітності. Це дозволяє виконувати складні аналітичні запити без впливу на продуктивність основної транзакційної бази даних.

Матеріалізовані представлення (materialized views) у PostgreSQL можуть слугувати проміжним рівнем для кешування агрегованих даних [15].

Для обробки великих обсягів даних у реальному часі доцільним є впровадження Apache Kafka як шини подій. Кожна значуща дія в системі (створення замовлення, зміна статусу, реєстрація користувача) публікується як подія в Kafka, звідки вона може бути спожита різними консьюмерами: аналітичним сервісом для агрегації, сервісом нотифікацій, системою виявлення аномалій.

Для візуалізації аналітичних даних рекомендується інтеграція з Metabase - open-source платформою бізнес-аналітики, яка дозволяє створювати інтерактивні дашборди та звіти без необхідності написання SQL-запитів, підключаючись безпосередньо до PostgreSQL [15].

Розгортання на хмарних сервісах та налаштування інфраструктури.

Міграція системи на хмарну інфраструктуру є логічним кроком для забезпечення масштабованості, високої доступності та зменшення операційного навантаження [4]. Серед провідних хмарних провайдерів - Amazon Web Services (AWS), Google Cloud Platform (GCP) та Microsoft Azure - для e-commerce застосунків оптимальним вибором є AWS завдяки найширшому спектру сервісів та зрілій екосистемі [34].

Оркестрація контейнерів є ключовим аспектом хмарного розгортання. Amazon Elastic Kubernetes Service (EKS) надає керований Kubernetes кластер, який автоматизує розгортання, масштабування та управління контейнеризованими застосунками. Альтернативою є Amazon Elastic Container Service (ECS) з AWS.

Fargate - serverless compute engine, який усуває необхідність управління серверами та дозволяє платити лише за фактично використані ресурси.

Для автоматичного масштабування застосунку використовується

Horizontal Pod Autoscaler (HPA) в Kubernetes, який збільшує або зменшує кількість реплік подів на основі метрик CPU, пам'яті або кастомних метрик. Application Load Balancer (ALB) розподіляє вхідний трафік між екземплярами застосунку, забезпечуючи відмовостійкість та рівномірне навантаження.

Захист від DDoS-атак реалізується на кількох рівнях. AWS Shield Standard автоматично захищає від найпоширеніших мережевих атак (SYN/UDP floods, reflection attacks) без додаткової оплати. AWS Shield Advanced надає розширений захист з цілодобовою підтримкою DDoS Response Team та фінансовими гарантіями. AWS Web Application Firewall (WAF) захищає від атак на рівні застосунку (SQL injection, XSS, CSRF), дозволяючи створювати правила фільтрації HTTP-запитів на основі IP-адрес, географічного розташування, розміру запиту та патернів у заголовках.

Amazon CloudFront - глобальна мережа доставки контенту (CDN) - кешує статичні ресурси (зображення товарів, CSS, JavaScript) на edge-локаціях по всьому світу [6], зменшуючи латентність для кінцевих користувачів та знижуючи навантаження на origin-сервери. Для зберігання зображень та файлів використовується Amazon S3 з налаштуванням lifecycle policies для автоматичного переміщення рідко використовуваних файлів у дешевші класи зберігання.

База даних PostgreSQL розгортається через Amazon RDS з увімкненою опцією Multi-AZ [15], яка забезпечує синхронну реплікацію в іншу зону доступності та автоматичне перемикання при збої основного екземпляра. Redis для кешування та управління сесіями розгортається через Amazon ElastiCache з підтримкою cluster mode для горизонтального масштабування.

Моніторинг та observability забезпечуються стеком Amazon CloudWatch для збору метрик та логів, AWS X-Ray для розподіленого трейсінгу запитів через мікросервіси [24], та інтеграцією з Prometheus/Grafana для візуалізації метрик Kubernetes кластера. Налаштування алертів на критичні метрики (час відповіді API > 500ms, error rate > 1%, CPU > 80%) дозволяє проактивно реагувати на проблеми до їх впливу на користувачів.

Висновок

У третьому розділі було здійснено практичну реалізацію спроектованої системи електронної комерції FullMag. Розроблено комплект UML-діаграм для візуалізації архітектури системи: діаграма прецедентів (Use Case) для відображення функціональних можливостей користувачів різних ролей, діаграма класів для представлення основних сутностей та їх взаємозв'язків, діаграма пакетів для демонстрації модульної організації коду.

Продемонстровано функціонування розробленого веб-застосунку та мобільного додатку через серію скріншотів, що ілюструють основні екрани: головну сторінку, каталог товарів з фільтрацією та сортуванням, картку товару, кошик, оформлення замовлення, особистий кабінет користувача, адміністративну панель з аналітикою та управлінням контентом.

Визначено вимоги до апаратного та програмного забезпечення для серверної інфраструктури, веб-застосунку та мобільних пристроїв. Окреслено перспективи подальшого розвитку системи, включаючи впровадження Elasticsearch для повнотекстового пошуку [17], розширення аналітики з використанням CQRS та Apache Kafka, міграцію на хмарну інфраструктуру AWS з використанням Kubernetes для оркестрації контейнерів.

ВИСНОВОК

У результаті виконання магістерської роботи було спроектовано та розроблено багатоплатформну систему електронної комерції FullMag, яка забезпечує повний цикл онлайн-продажів від перегляду каталогу товарів до оформлення замовлення та оплати.

Проведено аналіз предметної області електронної комерції та досліджено існуючі рішення на ринку. Визначено основні функціональні вимоги до сучасних систем онлайн-торгівлі та виявлено їх переваги і недоліки. На основі проведеного аналізу сформульовано вимоги до розроблюваної системи.

Обґрунтовано вибір технологій для реалізації системи. Для серверної частини обрано фреймворк NestJS з використанням TypeScript, що забезпечує типобезпеку та модульну архітектуру. Для веб-клієнта використано Next.js 14 з React 18, що дозволяє реалізувати серверний рендеринг та оптимізувати продуктивність. Мобільний застосунок розроблено на React Native 0.82 для підтримки платформ Android та iOS з єдиної кодової бази.

Спроектовано архітектуру системи на основі монорепозиторію з використанням npm workspaces. Розроблено структуру бази даних PostgreSQL з 25+ сутностями для зберігання інформації про товари, користувачів, замовлення, відгуки та інші бізнес-об'єкти. Реалізовано кешування даних за допомогою Redis для підвищення швидкодії системи.

Реалізовано серверну частину системи (REST API) з наступними модулями: автентифікація та авторизація з JWT-токенами, управління товарами та категоріями, кошик та оформлення замовлень, система відгуків та рейтингів, інтеграція з платіжною системою Stripe, інтеграція з службою доставки Нова Пошта, повнотекстовий пошук через Elasticsearch, система сповіщень через email, Telegram та WhatsApp.

Розроблено веб-застосунок з адаптивним дизайном, який коректно відображається на пристроях з різною роздільною здатністю екрану. Реалізовано адміністративну панель для управління товарами, замовленнями, користувачами, аналітикою продажів та іншими аспектами системи.

Створено мобільний застосунок для платформ Android та iOS з використанням React Native. Застосунок підтримує всі основні функції покупця: перегляд каталогу, пошук та фільтрацію товарів, управління кошиком, оформлення замовлень та перегляд історії покупок.

Розроблена система FullMag відповідає сучасним вимогам до платформ електронної комерції та може бути використана як основа для створення повноцінного інтернет-магазину. Модульна архітектура системи дозволяє легко розширювати функціональність та адаптувати її під конкретні бізнес-потреби.

Перспективами подальшого розвитку системи є: інтеграція з додатковими платіжними системами (LiqPay, Приват24), розширення можливостей аналітики з використанням машинного навчання для рекомендацій товарів, додавання підтримки мультимовності та мультивалютності, інтеграція з маркетплейсами (Rozetka, Prom.ua) та реалізація Progressive Web App (PWA) версії веб-застосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шибицький В.П., Шибицька Н.М., Савчук М.М. та ін. Обчислювальна техніка і програмування / Навч. посібник. - К.: КМУЦА. 1998 - 184 с.
2. Shibitskay N.N. Computer technology: Manual. - Kyiv: NAU, 2001. -76 p.
3. Шибицька Н.М., Кочеткова О.В. Інтеграційне тестування та контроль якості програмного забезпечення// Proceedings 1st international scientific and practical conference «Information Systems and Technology: Results and Prospects» (IST 2024)", March 6, 2024 - К.: FIT TSNUK, 2024.- С.366-369. http://kist.ntu.edu.ua/konferencii/46_konf_2024.pdf
4. Пінчук Д. В., Шибицька Н. М. Хмарний сервіс для динамічного обміну даними автоматизованої системи керування. Збірник тез доповідей Міжнародної науково-практичної конференції «Проблеми та перспективи розвитку науки, освіти і технологій» Ч.2, 27 січня 2022 р., м. Полтава, Україна С.49-51. URL: <http://www.economics.in.ua/2022/02/2.html?m=1>
5. Frain B. Responsive Web Design with HTML5 and CSS: Build future-proof responsive websites using the latest HTML5 and CSS techniques. 4th ed. Birmingham: Packt Publishing, 2022. 408 p.
6. Flanagan D. JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language. 7th ed. Sebastopol: O'Reilly Media, 2020. 706 p.
7. Freeman A. Pro React 18 for JavaScript Development. 5th ed. New York: Apress, 2022. 968 p.
8. Wieruch R. The Road to React: Your journey to master plain yet pragmatic React.js. Self-published, 2023. 350 p.
9. Boduch A., Derks R. React and React Native: Build cross-platform JavaScript and TypeScript apps for the web, desktop, and mobile. 5th ed. Birmingham: Packt Publishing, 2024. 518 p.
10. Katz M., Nance C. React Native Cookbook: Recipes for solving common React Native development problems. 2nd ed. Birmingham: Packt Publishing, 2023. 450 p.

11. Kamil Mysliwiec. NestJS Documentation. URL: <https://docs.nestjs.com/> (дата звернення: 23.11.2025).
12. Cherny B. Programming TypeScript: Making Your JavaScript Applications Scale. Sebastopol: O'Reilly Media, 2019. 324 p.
13. Vanderkam D. Effective TypeScript: 83 Specific Ways to Improve Your TypeScript. 2nd ed. Sebastopol: O'Reilly Media, 2024. 400 p.
14. Next.js Documentation. Vercel Inc. URL: <https://nextjs.org/docs> (дата звернення: 02.11.2025).
15. PostgreSQL 16 Documentation. The PostgreSQL Global Development Group. URL: <https://www.postgresql.org/docs/16/> (дата звернення: 28.10.2025).
16. Carlson J. Redis in Action. Shelter Island: Manning Publications, 2013. 320 p.
17. Gormley C., Tong Z. Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine. Sebastopol: O'Reilly Media, 2015. 724 p.
18. Stripe API Documentation. Stripe Inc. URL: <https://stripe.com/docs/api> (дата звернення: 10.10.2025).
19. Docker Documentation. Docker Inc. URL: <https://docs.docker.com/> (дата звернення: 15.11.2025).
20. Turnbull J. The Docker Book: Containerization is the new virtualization. Self-published, 2023. 342 p.
21. OWASP Top Ten. OWASP Foundation. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 30.11.2025).
22. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston: Prentice Hall, 2017. 432 p.
23. Fowler M. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley Professional, 2002. 560 p.
24. Newman S. Building Microservices: Designing Fine-Grained Systems. 2nd ed. Sebastopol: O'Reilly Media, 2021. 616 p.
25. Banks A., Porcello E. Learning React: Modern Patterns for Developing React Apps. 2nd ed. Sebastopol: O'Reilly Media, 2020. 310 p.

26. Нова Пошта API Documentation. URL: <https://developers.novaposhta.ua/documentation> (дата звернення: 18.10.2025).
27. WhatsApp Business API Documentation. Meta Platforms Inc. URL: <https://developers.facebook.com/docs/whatsapp> (дата звернення: 31.10.2025).
28. Telegram Bot API Documentation. Telegram FZ LLC. URL: <https://core.telegram.org/bots/api> (дата звернення: 01.12.2025).
29. React Native Documentation. Meta Platforms Inc. URL: <https://reactnative.dev/docs/getting-started> (дата звернення: 03.12.2025).
30. TypeORM Documentation. URL: <https://typeorm.io/> (дата звернення: 06.11.2025).
31. Jest Documentation. Meta Platforms Inc. URL: <https://jestjs.io/docs/getting-started> (дата звернення: 29.11.2025).
32. Sentry Documentation. Functional Software Inc. URL: <https://docs.sentry.io/> (дата звернення: 10.11.2025).
33. Richardson L., Ruby S. RESTful Web APIs: Services for a Changing World. Sebastopol: O'Reilly Media, 2013. 406 p.
34. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol: O'Reilly Media, 2017. 616 p.
35. ClickUp™ Product Features. URL: <https://clickup.com/features> (дата звернення: 05.12.2025).
36. ClickUp™ for Project Management Solutions. URL: <https://clickup.com/teams/project-management> (дата звернення: 16.10.2025).
37. GitHub flow - GitHub Docs. URL: <https://docs.github.com/get-started/quickstart/github-flow> (дата звернення: 15.11.2025).
38. What is the best Git branch strategy? | GitKraken. URL: <https://www.gitkraken.com/learn/git/best-practices/git-branch-strategy> (дата звернення: 01.12.2025).

39. Git branching guidance - Azure Repos | Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/devops/repos/git/git-branching-guidance> (дата звернення: 28.11.2025).
40. Conventional Commits 1.0.0. URL: <https://www.conventionalcommits.org/en/v1.0.0/> (дата звернення: 15.11.2025).
41. What are Git version control best practices? | GitLab. URL: <https://about.gitlab.com/topics/version-control/version-control-best-practices/> (дата звернення: 21.11.2025).
42. Config 2025: Pushing Design Further | Figma Blog. URL: <https://www.figma.com/blog/config-2025-recap/> (дата звернення: 16.11.2025).
43. What's new from Schema 2025 - Figma Learn. URL: <https://help.figma.com/hc/en-us/articles/35794667554839-What-s-new-from-Schema-2025> (дата звернення: 11.11.2025).
44. Guide to Dev Mode - Figma Learn. URL: <https://help.figma.com/hc/en-us/articles/15023124644247-Guide-to-Dev-Mode> (дата звернення: 29.10.2025).
45. Guide to developer handoff in Figma. URL: <https://www.figma.com/best-practices/guide-to-developer-handoff/> (дата звернення: 29.10.2025).
46. Object Management Group. About the Unified Modeling Language Specification. URL: <https://www.omg.org/spec/UML> (дата звернення: 21.11.2025).
47. Object Management Group. OMG Unified Modeling Language (OMG UML), Version 2.5.1. December 2017. URL: <https://www.omg.org/spec/UML/2.5.1> (дата звернення: 30.11.2025).
48. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd Edition. Addison-Wesley Professional, 2003. 208 p.

49. Larman C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3rd Edition. Prentice Hall, 2004. 736 p.
50. Rumbaugh J., Jacobson I., Booch G. *The Unified Modeling Language Reference Manual*. 2nd Edition. Addison-Wesley Professional, 2004. 752 p.
51. Briand L.C., Labiche Y., Di Penta M., Yan-Bondoc H. An experimental investigation of formality in UML-based development. *IEEE Transactions on Software Engineering*. 2005. Vol. 31, No. 10. P. 833-849