

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«ІНФОРМАЦІЙНА СИСТЕМА ОНЛАЙН-МАГАЗИНУ
АВТОЗАПЧАСТИН З ІНТЕГРАЦІЄЮ TELEGRAM-СПОВІЩЕНЬ»**

на здобуття освітнього ступеня бакалавр

за спеціальності 126 Інформаційні системи та технології

(код, найменування спеціальності)

освітньо-професійної програми Інформаційні системи та технології

(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Олександр КАРПЕНКО

(ім'я, ПРІЗВИЩЕ здобувача)

Виконав: здобувач вищої освіти гр. ІСД-41

Олександр КАРПЕНКО

(ім'я, ПРІЗВИЩЕ)

Керівник:

к.т.н., доцент

Оксана ТКАЛЕНКО

(ім'я, ПРІЗВИЩЕ)

Рецензент:

(ім'я, ПРІЗВИЩЕ)

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

Каміла СТОРЧАК

“ ____ ” _____ 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Карпенку Олександру Олексійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Інформаційна система онлайн-магазину автозапчастин з інтеграцією Telegram-сповіщень

керівник кваліфікаційної роботи: Оксана ТКАЛЕНКО, к.т.н., доцент

(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “20” лютого 2026 р. №51

2. Строк подання кваліфікаційної роботи “1” червня 2026 р.

3. Вихідні дані кваліфікаційної роботи:

1. Принципи побудови та функціонування онлайн-магазинів автотоварів.
2. Принципи інтеграції месенджерів у вебзастосунки.
3. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз передумов цифровізації торгівлі автозапчастинами та огляд технологій розробки веборієнтованих платформ.
2. Аналітичне обґрунтування вимог до інформаційної системи та моделювання архітектури.
3. Практична реалізація онлайн-магазину з Telegram-сповіщеннями та тестування.

5. Перелік ілюстраційного матеріалу: презентації

6. Дата видачі завдання “20” лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	24.02–03.03.26	
2.	Аналіз передумов цифровізації торгівлі автозапчастинами та огляд технологій	04.03–17.03.26	
3.	Аналітичне обґрунтування вимог та моделювання архітектури системи	18.03–31.03.26	
4.	Практична реалізація онлайн-магазину з Telegram-сповіщеннями	01.04–24.04.26	
5.	Висновки по роботі	25.04–16.05.26	
6.	Розробка демонстраційних матеріалів, доповідь	19.05–20.05.26	
7.	Оформлення бакалаврської роботи	21.05–30.05.26	

Здобувач вищої освіти _____ Олександр КАРПЕНКО

(підпис)

(ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи _____

(підпис)

Оксана ТКАЛЕНКО

(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступня бакалавр: 62 стор., 18 табл., 19 рис., 25 джерел.

Мета роботи – розробка та обґрунтування інформаційної системи онлайн-магазину автозапчастин з інтеграцією Telegram-сповіщень.

Об'єкт дослідження – процес організації електронної торгівлі автозапчастинами засобами веборієнтованих інформаційних систем.

Предмет дослідження – методи проєктування і програмної реалізації онлайн-магазину автозапчастин з інтеграцією Telegram-сповіщень.

Короткий зміст роботи. У бакалаврській роботі досліджено передумови цифровізації торгівлі автозапчастинами та проведено огляд технологій створення веборієнтованих торговельних платформ. Визначено користувацькі сценарії оформлення замовлень, сформовано вимоги до каталогу товарів та адміністративної панелі, змодельовано архітектуру взаємодії клієнтської частини з сервером. Реалізовано серверну частину на базі NestJS з GraphQL API, PostgreSQL та Prisma ORM, клієнтський інтерфейс на Next.js з React та TypeScript, механізм JWT-авторизації та інтеграцію Telegram Bot API для сповіщення менеджера про нові замовлення. Проведено функціональне тестування основних сценаріїв роботи системи.

КЛЮЧОВІ СЛОВА: ЕЛЕКТРОННА КОМЕРЦІЯ, ОНЛАЙН-МАГАЗИН, АВТОЗАПЧАСТИНИ, ІНФОРМАЦІЙНА СИСТЕМА, TELEGRAM-СПОВІЩЕННЯ, NESTJS, GRAPHQL, POSTGRESQL, JWT-AВТОРИЗАЦІЯ.

ЗМІСТ

КВАЛІФІКАЦІЙНА РОБОТА.....	1
ВСТУП.....	8
1 ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ ОНЛАЙН-МАГАЗИНУ АВТОЗАПЧАСТИН.....	11
1.1 Передумови цифровізації торгівлі автозапчастинами в електронній комерції.....	11
1.2 Характеристика функціональних можливостей сучасних онлайн-магазинів автотоварів.....	14
1.3 Огляд технологій створення веборієнтованих торговельних платформ	19
2 АНАЛІТИЧНЕ ОБҐРУНТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН- МАГАЗИНУ	24
2.1 Визначення користувацьких сценаріїв оформлення замовлення автозапчастин	24
2.2 Формування вимог до каталогу товарів та адміністративної панелі.....	29
2.3 Моделювання архітектури взаємодії клієнтської частини з сервером.....	34
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ОНЛАЙН-МАГАЗИНУ З TELEGRAM- СПОВІЩЕННЯМИ	40
3.1 Побудова серверної частини програмного продукту онлайн-магазину.....	40
3.2 Розгортання клієнтського інтерфейсу для продажу автозапчастин	44
3.3 Тестування роботи замовлень з передаванням Telegram-повідомлень.....	50
ВИСНОВКИ	65
ПЕРЕЛІК ПОСИЛАНЬ	67
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	70

ВСТУП

Актуальність теми. Розвиток електронної комерції істотно змінив організацію продажу товарів, оскільки покупці дедалі частіше обирають онлайн-канали для пошуку, порівняння та замовлення продукції. Особливо важливим напрямом електронної торгівлі є продаж автозапчастин, моторних олив, фільтрів, присадок та супутніх товарів для обслуговування транспортних засобів. Покупець у сфері автотоварів потребує швидкого доступу до структурованого каталогу, зрозумілої інформації про характеристики товару, наявність, ціну, категорію, бренд, спосіб оплати та умови отримання замовлення.

Традиційна модель продажу автозапчастин часто пов'язана з потребою телефонного уточнення, ручного пошуку товарів менеджером, неузгодженістю інформації про залишки та затримками під час обробки заявок. Для малого або середнього магазину проблема посилюється через обмеженість персоналу, оскільки менеджер має одночасно консультувати клієнтів, оновлювати асортимент, приймати замовлення та контролювати їх виконання. Використання інформаційної системи онлайн-магазину дає змогу скоротити час взаємодії між покупцем та продавцем, впорядкувати облік товарів, автоматизувати приймання замовлень і забезпечити зручний інструмент адміністрування.

Актуальність кваліфікаційної роботи полягає в необхідності розробки інформаційної системи онлайн-магазину автозапчастин, яка поєднує каталог товарів, механізм оформлення замовлення, адміністративне керування, авторизацію користувачів і Telegram-інформування менеджера.

Мета роботи – розробка та обґрунтування інформаційної системи онлайн-магазину автозапчастин з інтеграцією Telegram-сповіщень.

Для досягнення мети потрібно виконати наступні завдання:

1. проаналізувати передумови цифровізації торгівлі автозапчастинами та визначити роль онлайн-магазинів в розвитку електронної комерції;
2. охарактеризувати функціональні можливості веборієнтованих торговельних платформ для продажу автотоварів;

3. розглянути технологічний стек, придатний для побудови клієнтської та серверної частин інформаційної системи;
4. визначити користувацькі сценарії оформлення замовлень та вимоги до каталогу товарів;
5. спроектувати архітектуру взаємодії клієнтської частини, серверного застосунку, бази відомостей і Telegram API;
6. реалізувати програмні компоненти онлайн-магазину автозапчастин з адміністративною панеллю;
7. перевірити працездатність оформлення замовлення, авторизації, керування товарами та надсилання Telegram-повідомлення.

Об'єкт дослідження – процес організації електронної торгівлі автозапчастинами засобами веборієнтованих інформаційних систем.

Предмет дослідження – методи проектування і програмної реалізації онлайн-магазину автозапчастин з інтеграцією Telegram-сповіщень.

Методи дослідження. Під час виконання кваліфікаційної роботи використано методи аналізу літературних джерел і програмних рішень для визначення вимог до онлайн-магазину автозапчастин; методи системного аналізу для встановлення взаємозв'язків між користувачем, каталогом, замовленням, адміністративною панеллю та сервісом сповіщень; методи об'єктно-орієнтованого моделювання для опису структури програмних компонентів; методи проектування баз відомостей для формування моделей товарів, користувачів, категорій і замовлень; методи програмної реалізації для створення клієнтської та серверної частин системи; методи функціонального тестування для перевірки коректності роботи основних сценаріїв.

Наукова новизна одержаних результатів полягає в удосконаленні структури інформаційної системи онлайн-магазину автозапчастин через поєднання веборієнтованого каталогу, адміністративного керування замовленнями, рольового доступу та механізму Telegram-сповіщення в єдиному програмному рішенні. Запропонована організація взаємодії компонентів дає змогу підвищити швидкість

отримання інформації про нові заявки та зменшити залежність процесу обробки замовлень від ручної перевірки адміністративного інтерфейсу.

Практична значущість одержаних результатів полягає у створенні програмного продукту, придатного для використання як прототип онлайн-магазину автозапчастин. Розроблена система забезпечує перегляд товарів, роботу з категоріями, оформлення замовлення, авторизацію, доступ адміністратора до керування торговельною інформацією та надсилання повідомлення менеджеру через Telegram.

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, переліку посилань та додатків. В першому розділі розглянуто теоретичні аспекти розробки онлайн-магазину автозапчастин. В другому розділі обґрунтовано вимоги до інформаційної системи та змодельовано її архітектуру. В третьому розділі описано практичну реалізацію програмного продукту, роботу клієнтської та серверної частин, інтеграцію Telegram-сповіщень і результати тестування.

Апробація результатів та публікації. Основні положення і результати бакалаврської роботи доповідались на науково практичних конференціях, що проходили на базі Державного університету інформаційно-комунікаційних технологій.

1. Карпенко О.О. «Особливості розробки онлайн-магазинів». III «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу», ДУІКТ, Київ, 18 листопада 2025р., С, 197-198.

2. Карпенко О.О. «Аналіз типів сповіщень у Telegram» VII Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT», ДУІКТ, Київ 20 квітня 2026р., С, 5-6.

1 ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ ОНЛАЙН-МАГАЗИНУ АВТОЗАПЧАСТИН

1.1 Передумови цифровізації торгівлі автозапчастинами в електронній комерції

Розвиток електронної комерції значно змінив організацію торговельної діяльності, оскільки взаємодія між продавцем та покупцем дедалі частіше переноситься до цифрового середовища. Для ринку автозапчастин дана зміна має особливе значення, адже покупець працює не лише з назвою товару, але й з великою кількістю параметрів, а саме, категорією запчастини, сумісністю з транспортним засобом, брендом, ціною, наявністю, умовами доставки та швидкістю оброблення замовлення. В традиційній торгівлі, значна частина операцій виконується вручну, що збільшує навантаження на менеджера і створює ризик помилок під час підбору або оформлення покупки [1].

Онлайн-магазин автозапчастин виконує роль інформаційної системи, яка поєднує електронний каталог, засоби пошуку, картку товару, кошик, механізм замовлення, адміністративне керування асортиментом і канали сповіщення відповідальних осіб. В сфері автотоварів, швидкість доступу до актуальної інформації має безпосередній вплив на рішення покупця, оскільки клієнт зазвичай порівнює кілька пропозицій та очікує швидкої відповіді від продавця.

Цифровізація торгівлі автозапчастинами зумовлена не лише загальним поширенням інтернет-продажів, але й специфікою предметної області. Автозапчастини мають широку номенклатуру, різні цінові групи, залежність від виробника, технічних характеристик і потреби покупця. Навіть невеликий магазин може мати значну кількість товарних позицій, через що ручне ведення каталогу стає неефективним. Інформаційна система дає змогу впорядкувати дані про товари, надати покупцеві зручний інтерфейс для перегляду асортименту та забезпечити менеджеру інструмент швидкого реагування на замовлення [2]. Передумови переходу торгівлі автозапчастинами до цифрового формату наведено в таблиці 1.1

Таблиця 1.1

Передумови переходу торгівлі автозапчастинами до цифрового формату

Передумова	Зміст впливу на торговельний процес	Значення для онлайн-магазину
Зростання кількості товарних позицій	Асортимент автозапчастин потребує структурованого подання за категоріями, брендами та призначенням	Каталог забезпечує впорядкований доступ до товарів
Потреба у швидкому оформленні заявки	Покупець очікує мінімальної кількості дій від вибору товару до надсилання замовлення	Кошик і форма замовлення скорочують час покупки
Необхідність оперативної реакції менеджера	Пропущене звернення може призвести до втрати клієнта	Сповіщення через Telegram інформує про нову заявку одразу після її створення
Конкуренція між продавцями	Клієнт порівнює ціни, доступність товару та зручність сервісу	Вебінтерфейс підвищує якість представлення магазину
Потреба в централізованому керуванні	Адміністратор має контролювати товари, категорії та замовлення в одному середовищі	Адміністративна панель спрощує супровід продажів

Онлайн-магазин автозапчастин потрібно розглядати як прикладну інформаційну систему, яка підтримує повний цикл взаємодії з покупцем, від перегляду асортименту до передавання інформації менеджеру [3].

Важливою особливістю продажу автозапчастин є необхідність точного представлення товарної інформації. Покупець повинен швидко зрозуміти, до якої категорії належить товар, яку функцію він виконує, скільки коштує і чи можна його замовити без додаткових уточнень.

Для продавця, цифровізація створює можливість скоротити кількість ручних операцій. Адміністратор отримує змогу додавати товари, редагувати інформацію, контролювати категорії та переглядати замовлення без прямого втручання в програмний код. Менеджер, зі свого боку, може оперативно отримувати

повідомлення про нову заявку, що особливо важливо за високої динаміки звернень. Поєднання адміністративної панелі та Telegram-сповіщень робить процес продажу більш керованим та менш залежним від постійного моніторингу сайту [4].

Для покупця головним результатом стає доступність асортименту, прозорість ціни та можливість оформити замовлення без зайвих контактів. Для продавця важливим є впорядкування даних, скорочення повторюваних дій і підвищення контрольованості торговельного процесу.

Покупець очікує, що магазин буде доступний незалежно від робочого часу, а інформація про товар не потребуватиме обов'язкового звернення до продавця. Дана вимога є досить важливою, оскільки рішення про покупку часто приймається в умовах обмеженого часу, наприклад під час ремонту або планового технічного обслуговування автомобіля [5]. Якщо магазин надає зручний каталог і швидкий канал оформлення заявки, ймовірність завершення покупки зростає.

Водночас, цифровізація не усуває потребу в участі менеджера, а змінює характер його роботи. Замість постійного приймання первинних звернень менеджер отримує вже сформоване замовлення з ключовою інформацією. В результаті, онлайн-магазин виконує не лише презентаційну функцію, а й забезпечує автоматизований перехід від дії покупця до реакції відповідальної особи [6].

Для побудови сучасної інформаційної системи онлайн-магазину автозапчастин важливо враховувати взаємозв'язок між клієнтською частиною, серверною логікою, базою відомостей та зовнішнім сервісом сповіщень. Клієнтський інтерфейс відповідає за зручність перегляду товарів та оформлення замовлення. Серверна частина забезпечує обробку запитів, перевірку даних, авторизацію та взаємодію з базою. База відомостей зберігає інформацію про користувачів, товари, категорії та замовлення. Зовнішній сервіс сповіщень доповнює систему оперативним каналом інформування. Роль основних компонентів цифрової торгівлі автозапчастинами наведена в таблиці 1.2.

Таблиця 1.2

Роль основних компонентів цифрової торгівлі автозапчастинами

Компонент	Призначення	Очікуваний результат використання
Електронний каталог	Подання автозапчастин у структурованому вигляді	Покупець швидше знаходить потрібний товар
Картка товару	Відображення ціни, назви, категорії та опису	Клієнт отримує інформацію для прийняття рішення
Кошик замовлення	Накопичення обраних позицій перед оформленням	Процес покупки стає послідовним і зрозумілим
Адміністративна панель	Керування товарами, категоріями та замовленнями	Продавець підтримує актуальність торговельних даних
Серверна логіка	Оброблення запитів і збереження операцій	Система працює як єдиний програмний комплекс
Telegram-сповіщення	Передавання повідомлення про нове замовлення	Менеджер отримує оперативний сигнал для реагування

Цифровізація торгівлі автозапчастинами створює передумови для переходу від розрізнених каналів продажу до єдиного інформаційного середовища [7]. Онлайн-магазин має забезпечувати не лише привабливе подання асортименту, але й надійну обробку замовлень, зручне адміністрування та своєчасне інформування відповідальної особи.

1.2 Характеристика функціональних можливостей сучасних онлайн-магазинів автотоварів

Сучасний онлайн-магазин автотоварів є не лише вебресурсом для демонстрації продукції, а комплексною інформаційною системою, яка забезпечує зв'язок між покупцем, товарним каталогом, замовленням, менеджером і внутрішніми засобами адміністрування. Для сфери продажу автозапчастин, функціональні можливості мають особливе значення, оскільки покупець працює з

технічною продукцією, де помилка у виборі може призвести до додаткових витрат часу, повернення товару або повторного звернення до продавця [8].

Функціональність онлайн-магазину автозапчастин формується на перетині потреб покупця та завдань адміністратора. Покупець очікує простого перегляду товарів, наявності опису, зрозумілої структури категорій, можливості додати позицію до кошика та оформити замовлення без зайвих дій. Адміністратор потребує засобів керування товарами, категоріями, замовленнями та користувачами. Основні групи функціональних можливостей онлайн-магазину автотоварів узагальнено в таблиці 1.3.

Таблиця 1.3

Основні групи функціональних можливостей онлайн-магазину автотоварів

Група функцій	Призначення	Практичне значення для роботи магазину
Каталог автотоварів	Подання товарних позицій за категоріями	Покупець отримує впорядкований доступ до асортименту
Картка товару	Відображення назви, опису, вартості, зображення й належності до категорії	Клієнт швидше приймає рішення щодо покупки
Кошик замовлення	Збереження обраних товарів перед оформленням заявки	Процес покупки стає послідовним і контрольованим
Форма оформлення	Збір контактних даних покупця та інформації про замовлення	Менеджер отримує дані, потрібні для подальшої обробки
Адміністративна панель	Керування асортиментом, категоріями та заявками	Продавець підтримує актуальність інформації без зміни коду
Авторизація користувачів	Розмежування доступу між звичайним користувачем і адміністратором	Система захищає службові функції від стороннього використання
Система сповіщень	Передавання повідомлення про нову заявку	Відповідальна особа швидше реагує на замовлення

Успішність роботи системи залежить не лише від наявності товарних сторінок, але й від узгодженості всіх етапів, а саме, перегляду продукції, вибору позиції, формування кошика, передавання заявки, збереження даних і реакції менеджера. Якщо хоча б один елемент працює незручно або ненадійно, загальна якість сервісу знижується [9].

Для автозапчастин каталог повинен мати зрозумілу логіку поділу, оскільки асортимент може охоплювати мастильні матеріали, фільтри, акумулятори, гальмівні компоненти, освітлення, аксесуари та інші групи товарів. Категоризація допомагає покупцеві не переглядати весь набір позицій, а перейти до потрібного типу продукції. Для адміністратора, категорії спрощують керування асортиментом, оскільки кожен товар прив'язується до певної групи.

Картка товару виконує інформаційну та комерційну функцію. Вона повинна містити достатньо даних для первинного вибору, до яких відносять назву, зображення, короткий опис, вартість і належність до категорії. Недостатність інформації змушує користувача звертатися до менеджера ще до оформлення заявки, що зменшує ефективність електронного каналу продажу.

Оформлення замовлення є центральним сценарієм роботи онлайн-магазину. Саме на даному етапі, інтерес покупця переходить в конкретну заявку, яку має отримати продавець. Форма замовлення повинна бути простою, але достатньою для подальшої обробки. Надмірна кількість полів ускладнює процес покупки, а нестача даних створює потребу в додатковому зв'язку з клієнтом. Оптимальна форма має збирати контактну інформацію, відомості про обрані товари та дані, які допомагають менеджеру швидко розпочати роботу з заявкою [9].

Кошик виконує роль проміжного простору між переглядом товарів та надсиланням замовлення. Для покупця він є засобом контролю обраних позицій, а для системи, механізмом формування структури майбутньої заявки. В магазині автозапчастин кошик особливо важливий, оскільки покупець може купувати не один товар, а набір позицій для обслуговування автомобіля.

Функціональність адміністративної панелі визначає зручність супроводу системи після її запуску. Адміністратор повинен мати змогу додавати нові товари,

змінювати наявні записи, працювати з категоріями, переглядати замовлення та контролювати користувачів. Без адміністративної панелі підтримка магазину потребувала б прямого редагування бази відомостей або коду, що є неприйнятним для реального використання. Панель керування перетворює онлайн-магазин з демонстраційного сайту на повноцінний інструмент торговельної діяльності [10]. Функції адміністративної панелі онлайн-магазину автотоварів наведено в таблиці 1.4

Таблиця 1.4

Функції адміністративної панелі онлайн-магазину автотоварів

Функція адміністратора	Зміст операції	Очікуваний результат
Додавання товару	Внесення нової позиції до каталогу	Асортимент поповнюється без втручання в програмний код
Редагування запису	Оновлення назви, опису, вартості або зображення	Інформація на сайті залишається актуальною
Керування категоріями	Створення та оновлення товарних груп	Каталог зберігає логічну структуру
Перегляд замовлень	Ознайомлення з даними заявок покупців	Менеджер контролює процес оброблення
Робота з користувачами	Контроль облікових записів і прав доступу	Службові операції захищаються від несанкціонованого використання
Перевірка повідомлень	Контроль факту надходження заявки	Зменшується ризик пропуску нового замовлення

Авторизація та рольовий доступ є обов'язковими для онлайн-магазину, який має адміністративну частину. Звичайний користувач повинен працювати з каталогом і замовленням, тоді як адміністратор отримує доступ до службових функцій. Розмежування прав захищає систему від небажаних змін, випадкового видалення інформації або стороннього перегляду внутрішніх даних. Рольовий доступ забезпечує базовий рівень безпеки та підтримує відповідальність за виконання адміністративних дій.

Окрему роль в сучасних онлайн-магазинах відіграють сповіщення. Для невеликого магазину автозапчастин, сповіщення часто мають навіть більше практичне значення, ніж складна аналітика, адже головним завданням є швидке реагування на заявку покупця. Якщо менеджер отримує повідомлення одразу після оформлення замовлення, зменшується проміжок між дією клієнта та відповіддю продавця. Telegram, в даному випадку, виступає зручним каналом службового інформування, оскільки повідомлення надходить в звичне середовище комунікації та не потребує постійного відкриття адміністративної панелі.

Функція Telegram-сповіщень доповнює класичну модель онлайн-магазину. Зазвичай замовлення зберігається в базі відомостей і стає доступним адміністратору через панель керування. Проте дана модель передбачає, що відповідальна особа регулярно перевіряє систему. Інтеграція месенджера змінює логіку взаємодії: магазин сам ініціює інформування менеджера після створення заявки [11].

Окрім основних функцій, онлайн-магазин автозапчастин повинен забезпечувати зручність користування. Інтерфейс має бути зрозумілим для покупця без додаткових пояснень, оскільки складна навігація знижує ймовірність завершення покупки. Для магазину автотоварів важливими є помітні категорії, читабельні картки товарів, зрозуміла кнопка додавання до кошика, проста форма замовлення та коректне відображення інформації на різних пристроях. Покупець може переглядати магазин зі смартфона, тому адаптивність інтерфейсу впливає на реальну доступність торговельної системи.

Функціональні можливості онлайн-магазину мають бути узгоджені з архітектурою програмного продукту. Клієнтська частина відповідає за візуальну взаємодію з користувачем, серверна частина обробляє запити та реалізує бізнес-логіку, база відомостей зберігає дані, а зовнішній сервіс сповіщень забезпечує передавання повідомлень.

Функціональні можливості сучасного онлайн-магазину автотоварів повинні охоплювати електронний каталог, картки товарів, кошик, оформлення замовлень, адміністративне керування, авторизацію, рольовий доступ і Telegram-сповіщення.

1.3 Огляд технологій створення веборієнтованих торговельних платформ

Розробка веборієнтованої торговельної платформи потребує узгодженого вибору технологій, які забезпечують зручний інтерфейс покупця, надійну серверну логіку, структуроване зберігання відомостей, захищену авторизацію та можливість інтеграції із зовнішніми сервісами. Для онлайн-магазину автозапчастин, технологічна основа має підтримувати роботу з каталогом товарів, категоріями, кошиком, замовленнями, адміністративною панеллю та службовими повідомленнями для менеджера.

Клієнтська частина онлайн-магазину відповідає за безпосередню взаємодію з покупцем. Саме на даному рівні користувач переглядає каталог, відкриває картки товарів, додає позиції до кошика й оформлює замовлення. Для реалізації клієнтської частини варто застосовувати фреймворк Next.js, який базується на React та підтримує створення вебзастосунків з файловою маршрутизацією, серверними компонентами та зручними засобами організації сторінок [12]. Призначення технологій клієнтського рівня онлайн-магазину узагальнено в таблиці 1.5.

Таблиця 1.5

Призначення технологій клієнтського рівня онлайн-магазину

Технологія	Призначення в інформаційній системі	Практичне значення для онлайн-магазину
React	Формування компонентної структури інтерфейсу	Картки товарів, кошик і сторінки каталогу можна створювати як окремі елементи
Next.js	Організація сторінок, маршрутизації та взаємодії з вебзастосунком	Покупець отримує швидкий і зрозумілий доступ до основних сторінок магазину
TypeScript	Типізація програмного коду клієнтської частини	Зменшується кількість помилок під час розроблення інтерфейсу

Продовження таблиці 1.5

CSS-модулі або стилізаційні бібліотеки	Оформлення зовнішнього вигляду сторінок	Інтерфейс набуває єдиного стилю та кращої читабельності
HTTP або GraphQL-клієнт	Передавання запитів до серверної частини	Дані каталогу та замовлень надходять до інтерфейсу через програмний API

Використання компонентної структури має важливе значення для торговельної платформи, оскільки інтерфейс онлайн-магазину складається з повторюваних блоків. До них належать картки товарів, кнопки додавання до кошика, форми введення даних, навігаційні панелі, повідомлення про результат операції та службові елементи адміністративної частини [13].

TypeScript доповнює клієнтську розробку за рахунок статичної типізації, що корисно під час роботи з об'єктами товарів, категорій, користувачів та замовлень. Помилка в структурі даних може призвести до некоректного відображення картки товару, втрати частини інформації в кошику або неправильного передавання заявки на сервер [14].

Серверна частина виконує центральну роль в роботі інформаційної системи, оскільки саме вона опрацьовує запити від клієнтського інтерфейсу, перевіряє коректність даних, взаємодіє з базою відомостей, формує відповіді для клієнта та запускає службові дії після створення замовлення. Для побудови серверного рівня доцільним є використання NestJS.

NestJS добре відповідає потребам онлайн-магазину через модульну структуру. Серверну логіку можна розділити на окремі функціональні частини, пов'язані з товарами, категоріями, користувачами, замовленнями, авторизацією та Telegram-інтеграцією. Модульність спрощує супровід програмного продукту, оскільки зміни в одному функціональному блоці не потребують повного перегляду всієї серверної частини [15].

Взаємодія між клієнтською та серверною частиною може бути організована за допомогою GraphQL. На відміну від класичної REST-моделі, GraphQL дає

можливість клієнтові запитувати саме той набір полів, який потрібен для певного інтерфейсного сценарію.

GraphQL особливо корисний в системах, де різні частини інтерфейсу потребують різних наборів даних. Наприклад, блок каталогу використовує назву, ціну та зображення товару, кошик потребує кількості підсумкової вартості, адміністративна панель працює з ширшим набором полів. Завдяки схемі GraphQL, серверна частина може чітко визначити доступні операції читання і зміни даних [16].

Для зберігання відомостей онлайн-магазину варто застосовувати реляційну базу даних PostgreSQL. Реляційна модель є зручною, оскільки товари, категорії, користувачі та замовлення мають чіткі зв'язки. Наприклад, товар належить до категорії, замовлення пов'язане з покупцем або контактними даними, а позиції замовлення містять зв'язок між конкретним товаром та кількістю.

Prisma ORM використовується як проміжний рівень між серверним кодом і базою даних. Він дає змогу описувати моделі даних, виконувати міграції та працювати з базою через типізований клієнт [17]. Документація Prisma зазначає, що Prisma ORM підтримує PostgreSQL та PostgreSQL-сумісні бази, зокрема локальні та хмарні рішення [18]. Технології зберігання та доступу до відомостей онлайн-магазину узагальнено в таблиці 1.6

Таблиця 1.6

Технології зберігання та доступу до відомостей онлайн-магазину

Технологія	Призначення	Приклад використання в онлайн-магазині
PostgreSQL	Збереження структурованих відомостей	Таблиці товарів, категорій, користувачів і замовлень
Prisma ORM	Опис моделей і виконання запитів з серверного коду	Створення товару, пошук замовлення, оновлення категорії
Міграції бази даних	Контроль зміни структури таблиць	Додавання нових полів до моделей без ручного переписування схеми

Продовження таблиці 1.6

Зв'язки між моделями	Відображення залежностей між об'єктами предметної області	Прив'язка товару до категорії та замовлення до переліку позицій
Типізований доступ	Узгодження структури даних з програмним кодом	Зменшення помилок під час оброблення серверних операцій

Авторизація є необхідною частиною торговельної платформи, оскільки адміністративна панель не повинна бути доступною звичайному відвідувачу. Для розмежування доступу може використовуватися JWT. Стандарт RFC 7519 визначає JSON Web Token як компактний URL-безпечний засіб передавання даних між двома сторонами, де дані кодуються у вигляді JSON-об'єкта та можуть бути підписані або захищені [19].

Рольовий доступ має практичне значення для захисту службових операцій. Покупець може переглядати каталог, додавати товари до кошика та оформляти замовлення. Адміністратор отримує розширені права, пов'язані з керуванням товарами, категоріями та замовленнями. Розділення прав зменшує ризик випадкового або навмисного втручання в торговельні дані.

Інтеграція Telegram-сповіщень розширює функціональність онлайн-магазину через зовнішній канал службової комунікації. Telegram Bot API є HTTP-інтерфейсом для створення ботів та взаємодії з Telegram з боку програмних систем. Бот може надсилати повідомлення менеджеру після створення замовлення. Повідомлення може містити номер заявки, перелік товарів, ім'я покупця, телефон та суму замовлення [20].

Розглянутий стек технологій відповідає потребам інформаційної системи онлайн-магазину автозапчастин, оскільки охоплює всі ключові рівні програмного продукту. Next.js забезпечує клієнтський інтерфейс і маршрутизацію сторінок. NestJS формує серверну основу з модульною структурою. GraphQL організовує взаємодію між клієнтом та сервером. PostgreSQL відповідає за збереження структурованих відомостей, а Prisma ORM спрощує роботу з моделями бази даних.

JWT підтримує авторизацію та рольовий доступ, тоді як Telegram Bot API забезпечує передавання службових повідомлень менеджеру.

В першому розділі розглянуто теоретичні аспекти розробки онлайн-магазину автозапчастин та визначено передумови переходу торговельної діяльності до цифрового формату. Ринок автотоварів потребує не лише вебпредставлення продукції, але й повноцінної інформаційної системи, здатної підтримувати каталог, замовлення, адміністративне керування та оперативне інформування відповідальної особи. Особливість предметної області полягає у великій кількості товарних позицій, потребі в логічній категоризації, точності опису продукції та швидкому реагуванні продавця на заявку покупця.

Ефективність торговельної платформи залежить від узгодженої роботи клієнтської та адміністративної частин. Для покупця ключовими є простота пошуку, зрозуміле представлення інформації та зручне оформлення заявки, а для продавця важливими залишаються актуальність асортименту, контроль замовлень та захист службових операцій від несанкціонованого доступу.

Огляд технологій створення веборієнтованих торговельних платформ показав доцільність використання сучасного стеку, який охоплює клієнтську частину, серверну логіку, API, базу відомостей, механізм авторизації та інтеграцію із зовнішнім каналом повідомлень. Next.js і React забезпечують побудову зручного інтерфейсу, NestJS дає змогу організувати модульну серверну архітектуру, GraphQL підтримує гнучкий обмін даними, PostgreSQL і Prisma ORM формують основу для зберігання та обробки структурованої інформації, а JWT використовується для розмежування доступу. Telegram Bot API доповнює функціональність системи засобом швидкого інформування менеджера про нове замовлення.

Матеріали першого розділу формують базу для подальшого аналітичного обґрунтування вимог до системи, моделювання її архітектури та опису практичної реалізації програмного продукту.

2 АНАЛІТИЧНЕ ОБҐРУНТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-МАГАЗИНУ

2.1 Визначення користувацьких сценаріїв оформлення замовлення автозапчастин

Проектування інформаційної системи онлайн-магазину автозапчастин потребує попереднього визначення користувацьких сценаріїв, які описують послідовність дій покупця, адміністратора та менеджера під час роботи з майбутнім програмним продуктом. Головним сценарієм для торговельної платформи є оформлення замовлення, оскільки саме він переводить перегляд асортименту в конкретну заявку на придбання товару. Якщо процес вибору продукції, додавання позицій до кошика та передавання контактних відомостей побудований незручно, електронний магазин не зможе повною мірою виконувати комерційну функцію.

Онлайн-магазин автозапчастин повинен враховувати специфіку предметної області, пов'язану з великою кількістю товарних позицій, різними категоріями автотоварів, потребою в точному описі продукції та швидкому зв'язку з покупцем. Користувач, який шукає автозапчастину або супутній товар, очікує зрозумілого каталогу, наявності ціни, короткого опису, зображення, категорії та можливості швидко сформулювати заявку [21]. Сценарій оформлення замовлення має будуватися як послідовний процес, де кожен етап логічно продовжує попередній і не створює зайвого навантаження для покупця. Учасники користувацьких сценаріїв онлайн-магазину автозапчастин наведені в таблиці 2.1.

Таблиця 2.1

Учасники користувацьких сценаріїв онлайн-магазину автозапчастин

Учасник процесу	Роль в інформаційній системі	Очікуваний результат взаємодії
Покупець	Переглядає каталог, обирає товар, формує кошик і надсилає заявку	Замовлення передане на опрацювання

Продовження таблиці 2.1

Адміністратор	Керує товарами, категоріями, обліковими записами та службовими даними	Асортимент і дані системи залишаються актуальними
Менеджер	Отримує інформацію про нову заявку та контактує з покупцем	Замовлення переходить до етапу опрацювання
Серверна частина	Перевіряє дані, зберігає заявку та запускає службові операції	Торговельний процес виконується без ручного втручання в базу
Telegram-бот	Передає повідомлення відповідальній особі після створення заявки	Менеджер швидко дізнається про нове звернення

Взаємодія зазначених учасників формує повний цикл роботи онлайн-магазину. Покупець ініціює процес через клієнтський інтерфейс, адміністратор підтримує коректність каталогу, серверна частина забезпечує обробку запиту, а менеджер приймає заявку до подальшої роботи. Telegram-бот в зазначеній логіці виконує роль службового каналу, який скорочує час між створенням замовлення та реакцією відповідальної особи.

Основний сценарій оформлення замовлення починається з відкриття сторінки каталогу. Покупець переглядає товарні позиції, порівнює вартість, аналізує опис та обирає потрібний автотовар. Після вибору позиції, користувач додає її до кошика, де може перевірити склад майбутньої покупки. Далі покупець переходить до форми оформлення, вводить контактні відомості, зазначає дані доставки, за потреби залишає коментар і підтверджує надсилання заявки. Після завершення операції система повинна зберегти замовлення та передати службове повідомлення менеджеру [21]. Основний сценарій оформлення замовлення автозапчастин наведено в таблиці 2.2.

Таблиця 2.2

Основний сценарій оформлення замовлення автозапчастин

Етап сценарію	Дія покупця	Очікувана реакція інформаційної системи
Перегляд каталогу	Користувач відкриває сторінку з автотоварами	Система відображає доступний асортимент
Аналіз товарної позиції	Покупець переглядає назву, опис, ціну й зображення	Інтерфейс надає достатню інформацію для вибору
Додавання до кошика	Користувач обирає потрібну позицію	Кошик оновлює склад майбутнього замовлення
Перевірка покупки	Покупець переглядає перелік обраних товарів	Система показує позиції, кількість і підсумкову інформацію
Заповнення форми	Користувач вводить контактні та адресні відомості	Інтерфейс готує дані для передавання на сервер
Підтвердження заявки	Покупець надсилає замовлення	Сервер перевіряє дані та створює запис
Службове інформування	Додаткових дій покупця не потрібно	Telegram-бот передає менеджеру повідомлення про нову заявку

Зазначений сценарій повинен бути простим для покупця, але достатньо інформативним для менеджера. Форма замовлення не має перевантажувати користувача зайвими полями, проте повинна містити відомості, необхідні для подальшого зв'язку та організації доставки. Оптимальний набір даних включає ім'я покупця, номер телефону, населений пункт, адресу, спосіб оплати та коментар. За наявності авторизації, система може пов'язувати заявку з обліковим записом користувача, що спрощує перегляд історії замовлень та повторне звернення до магазину.

Сценарій оформлення замовлення потребує перевірки введених даних. Якщо покупець не вказав ім'я, телефон або адресу, система не повинна створювати неповну заявку. Валідація допомагає уникнути ситуацій, коли менеджер отримує

повідомлення без необхідної інформації для зв'язку з клієнтом. Окрему увагу варто приділити складу кошика, оскільки замовлення без товарів не має комерційного змісту та не повинне потрапляти до бази даних [21].

Окремим напрямом проєктування є альтернативна поведінка покупця. Користувач може змінити склад кошика, видалити товар, повернутися до каталогу, відмовитись від покупки або помилитися під час заповнення форми. Інформаційна система повинна коректно підтримувати кожну з перелічених ситуацій, оскільки реальна взаємодія покупця з онлайн-магазином рідко відбувається за ідеально прямою траєкторією. Гнучкість сценарію підвищує зручність використання системи та зменшує ймовірність переривання покупки. Альтернативні сценарії поведінки покупця наведено в таблиці 2.3.

Таблиця 2.3

Альтернативні сценарії поведінки покупця

Ситуація	Причина виникнення	Очікувана поведінка системи
Зміна кількості товару	Покупець уточнює потребу в кількох одиницях продукції	Кошик оновлює кількість та підсумкову суму
Видалення позиції	Користувач відмовляється від частини покупки	Система прибирає товар з кошика
Повернення до каталогу	Покупець продовжує пошук автотоварів	Вміст кошика зберігається
Помилка у формі	Обов'язкове поле залишене порожнім або заповнене некоректно	Інтерфейс повідомляє про потребу виправлення
Спроба оформити порожній кошик	Користувач не додав жодної позиції	Система блокує створення заявки
Тимчасова відмова від покупки	Покупець не завершує процес оформлення	Замовлення не створюється в базі даних

Альтернативні сценарії мають важливе значення для якості користувацького досвіду. Якщо система зберігає вміст кошика під час повернення до каталогу, покупець не втрачає раніше обрані позиції. Якщо інтерфейс чітко вказує на

помилку у формі, користувач може швидко виправити дані без повторного проходження всього процесу. Якщо порожній кошик не допускається до оформлення, база даних не забивається непотрібними записами [21].

Після надсилання заявки починається внутрішній сценарій обробки замовлення. Серверна частина повинна прийняти дані з форми, перевірити обов'язкові поля, розрахувати підсумкову суму, створити запис про замовлення, зберегти товарні позиції та передати службову інформацію менеджеру. На відміну від зовнішнього сценарію, внутрішній процес не видимий покупцеві повністю, проте саме він визначає надійність роботи торговельної системи. Внутрішній сценарій обробки заявки наведено в таблиці 2.4.

Таблиця 2.4

Внутрішній сценарій обробки заявки

Етап внутрішньої обробки	Призначення	Очікуваний результат
Приймання даних	Отримання інформації з форми замовлення	Сервер має повний набір відомостей для перевірки
Валідація форми	Контроль заповнення обов'язкових полів	Некоректні заявки не зберігаються
Перевірка кошика	Контроль наявності товарних позицій	Порожнє замовлення не створюється
Розрахунок суми	Визначення підсумкової вартості покупки	Комерційні дані відповідають складу кошика
Збереження запису	Фіксація замовлення в базі даних	Адміністратор може переглянути заявку
Надсилання сповіщення	Передавання повідомлення менеджеру	Відповідальна особа дізнається про нове звернення

Telegram-сповіщення має бути частиною сценарію, який забезпечує швидку реакцію магазину на дії покупця. Повідомлення не повинне дублювати всю картку замовлення, оскільки надмірний обсяг тексту в месенджері ускладнює сприйняття. Для первинного інформування достатньо передати службовий номер заявки та контактний телефон покупця, а повні дані замовлення залишити в адміністративній частині системи. Дана організація дає менеджеру змогу швидко зрозуміти факт

появи нової заявки та перейти до детального перегляду в інформаційній системі [21].

Важливо враховувати адміністративний сценарій, який супроводжує оформлення замовлень. Адміністратор повинен мати змогу підтримувати актуальність каталогу, редагувати товарні позиції, оновлювати категорії, контролювати доступність продукції та переглядати заявки покупців. Без зазначених функцій сценарій оформлення замовлення може втратити практичну цінність, оскільки користувач бачитиме застарілу інформацію або намагатиметься придбати товар, який вже недоступний.

Сценарії оформлення замовлення автозапчастин створюють основу для подальшого формування вимог до інформаційної системи. Майбутній програмний продукт повинен забезпечувати перегляд асортименту, роботу з кошиком, оформлення заявки, перевірку обов'язкових даних, збереження замовлення, службове інформування менеджера та підтримку адміністративного керування. Узгодженість вказаних процесів дасть змогу побудувати онлайн-магазин, який виконуватиме не лише презентаційну функцію, але й забезпечуватиме повний цикл електронної взаємодії між покупцем та продавцем.

2.2 Формування вимог до каталогу товарів та адміністративної панелі

Формування вимог до каталогу товарів та адміністративної панелі є важливим етапом проєктування інформаційної системи онлайн-магазину автозапчастин, оскільки саме вказані компоненти забезпечують основну взаємодію між покупцем та продавцем. Каталог відповідає за представлення асортименту, навігацію, перегляд товарних позицій і підготовку користувача до оформлення замовлення. Адміністративна панель, зі свого боку, забезпечує керування торговельною інформацією, підтримку актуальності даних, контроль замовлень та розмежування службових операцій.

Для онлайн-магазину автозапчастин, каталог має виконувати роль структурованого інформаційного середовища, де покупець може швидко знайти

потрібну категорію продукції, переглянути короткий опис, оцінити вартість та додати позицію до кошика. Предметна область автотоварів характеризується значною кількістю найменувань, різними брендами, технічними характеристиками та відмінностями у призначенні продукції. Вимоги до каталогу мають охоплювати не лише зовнішнє оформлення, але й логіку групування, пошуку, відображення та актуалізації даних. Основні вимоги до каталогу товарів онлайн-магазину автозапчастин відображено в таблиці 2.5.

Таблиця 2.5

Основні вимоги до каталогу товарів онлайн-магазину автозапчастин

Група вимог	Зміст вимоги	Очікуваний результат
Структурованість асортименту	Товари мають бути розподілені за логічними категоріями	Покупець швидше знаходить потрібний тип автотовару
Повнота товарної інформації	Картка має містити назву, зображення, опис, ціну та належність до категорії	Користувач отримує достатні дані для вибору
Актуальність відомостей	Ціна, доступність і опис мають оновлюватися через адміністративну панель	Знижується ризик помилки під час оформлення заявки
Зручність перегляду	Інтерфейс має підтримувати зрозумілу навігацію між категоріями та товарами	Покупець не витрачає зайвий час на пошук
Підтримка кошика	Кожна доступна позиція повинна мати можливість додавання до замовлення	Каталог переходить від перегляду до комерційної дії
Адаптивність інтерфейсу	Сторінки мають коректно відображатися на різних пристроях	Магазин залишається доступним для користувачів смартфонів і комп'ютерів

Вимога структурованості каталогу має першочергове значення, оскільки асортимент автозапчастин може швидко збільшуватися. При відсутності категоризації, покупець змушений переглядати значну кількість позицій без чіткого орієнтиру. Для зручності варто передбачити розподіл товарів за

функціональним призначенням, наприклад моторні оливи, фільтри, акумулятори, гальмівні елементи, освітлення, аксесуари або інші групи продукції. Логічна структура каталогу допомагає покупцеві рухатись від загальної групи товарів до конкретної позиції.

Картка товару повинна надавати покупцеві інформацію, необхідну для прийняття рішення. Опис має бути лаконічним, але змістовним. Зображення товару покращує візуальне сприйняття, а ціна та статус доступності дають змогу оцінити реальну можливість покупки. Якщо частина відомостей відсутня, користувач буде змушений звертатися до менеджера ще до оформлення заявки, що зменшує ефективність електронного каналу продажу. Інформаційні поля картки товару представлено в таблиці 2.6.

Таблиця 2.6

Інформаційні поля картки товару

Поле картки	Призначення	Вплив на поведінку покупця
Назва товару	Ідентифікація позиції в каталозі	Користувач швидко розуміє предмет пропозиції
Категорія	Визначення належності до певної групи автотоварів	Полегшується навігація між спорідненими позиціями
Бренд	Відображення виробника або торговельної марки	Покупець може орієнтуватися на знайомого виробника
Опис	Пояснення призначення та особливостей товару	Зменшується потреба в додатковому уточненні
Ціна	Подання комерційної інформації	Користувач порівнює пропозицію з бюджетом
Зображення	Візуальне представлення продукції	Зростає довіра до товарної позиції
Доступність	Позначення можливості замовлення	Покупець уникає оформлення недоступної позиції

Вимоги до каталогу мають враховувати не лише інформаційне наповнення, але й поведінку користувача під час перегляду. Покупець може переходити між категоріями, відкривати детальні сторінки товарів, повертатися до попереднього перегляду, додавати продукцію до кошика та продовжувати пошук. Інтерфейс

повинен підтримувати вільний рух між зазначеними діями без втрати обраних позицій. Кошик має зберігати дані про товари, кількість та підсумкову суму, щоб користувач міг контролювати майбутню покупку перед оформленням.

Адміністративна панель повинна забезпечувати керування всіма даними, які впливають на роботу каталогу. Якщо адміністратор не має зручного інструменту для додавання та редагування продукції, магазин швидко втрачає актуальність. Для торговельної системи автозапчастин, особливо важливо оперативно оновлювати ціни, змінювати описи, додавати нові позиції, приховувати недоступні товари та підтримувати правильну структуру категорій. Всі зазначені операції мають виконуватися через захищений інтерфейс без прямої роботи з базою даних. Вимоги до адміністративної панелі онлайн-магазину автозапчастин наведено в таблиці 2.7.

Таблиця 2.7

Вимоги до адміністративної панелі онлайн-магазину автозапчастин

Група вимог	Зміст вимоги	Очікуваний результат
Керування товарами	Додавання, редагування, приховування й видалення товарних позицій	Асортимент підтримується в актуальному стані
Робота з категоріями	Створення та редагування груп автотоварів	Каталог зберігає логічну структуру
Контроль замовлень	Перегляд заявок покупців та зміна статусу обробки	Менеджер бачить стан виконання замовлень
Захищений доступ	Вхід до панелі лише для уповноважених осіб	Службові функції не доступні стороннім користувачам
Перевірка даних	Контроль обов'язкових полів під час створення товару	В каталог не потрапляють неповні записи
Зручність супроводу	Операції мають виконуватися через зрозумілі форми	Адміністратор працює без втручання в програмний код

Функція керування товарами має охоплювати повний життєвий цикл товарної позиції. На етапі додавання, адміністратор вводить назву, опис, ціну, категорію, бренд, зображення й статус доступності. Під час редагування він може

змінити вартість, уточнити опис або оновити візуальне представлення товару. У разі тимчасової відсутності позиції, вона не повинна вводити покупця в оману, тому панель має підтримувати зміну доступності або приховування товару з відкритого каталогу.

Категорії визначають логіку навігації та впливають на зручність пошуку продукції. Якщо назви груп нечіткі або товари розміщені в невідповідних розділах, користувач витрачає більше часу на перегляд і може не завершити покупку. Адміністратор повинен мати можливість створювати нові категорії, редагувати назви наявних груп та підтримувати коректне прив'язування товарів до відповідних розділів.

Контроль замовлень є наступною важливою частиною адміністративної панелі. Після того як покупець оформив заявку, менеджер або адміністратор повинен мати доступ до відомостей про покупця, телефон, адресу доставки, склад замовлення, суму та статус обробки. Адміністративний інтерфейс має показувати замовлення в зручному вигляді, щоб відповідальна особа могла швидко перейти від первинного перегляду до контакту з клієнтом. Зміна статусу замовлення допомагає контролювати етапи опрацювання: нова заявка, прийнято в роботу, виконано або скасовано.

Захищений доступ до адміністративної панелі є обов'язковою вимогою. Службові функції не можуть бути відкритими для звичайного покупця, оскільки це створює ризик несанкціонованої зміни цін, видалення товарів або перегляду персональних даних клієнтів. Під час проєктування системи необхідно розмежувати права доступу. Покупець повинен працювати з каталогом, кошиком і формою замовлення, а адміністратор має отримувати доступ до функцій керування даними лише після авторизації.

Каталог товарів та адміністративна панель мають бути логічно пов'язаними. Дані, які адміністратор вводить в службовій частині, повинні одразу впливати на відображення в клієнтському інтерфейсі. Зміна ціни має відображатися в картці товару, зміна доступності повинна впливати на можливість оформлення покупки, а редагування категорії має оновлювати навігацію каталогу.

Каталог повинен забезпечувати покупцеві зручний перегляд асортименту, доступ до повної товарної інформації, додавання позицій до кошика та перехід до оформлення замовлення. Адміністративна панель має забезпечувати службове керування товарами, категоріями, замовленнями та доступом.

2.3 Моделювання архітектури взаємодії клієнтської частини з сервером

Моделювання архітектури взаємодії клієнтської частини з сервером є необхідним етапом проєктування інформаційної системи онлайн-магазину автозапчастин, оскільки саме архітектурна модель визначає логіку передавання даних між користувацьким інтерфейсом, серверною частиною, базою даних та сервісом Telegram-сповіщень.

Клієнтська частина майбутньої системи має відповідати за взаємодію з покупцем та адміністратором. Покупець через вебінтерфейс переглядатиме каталог, відкриватиме картки товарів, додаватиме позиції до кошика та заповнюватиме форму замовлення. Адміністратор через захищену частину інтерфейсу працюватиме з товарами, категоріями, замовленнями та службовими параметрами. Серверна частина виконуватиме роль проміжного рівня між інтерфейсом і базою даних, оскільки саме вона повинна перевіряти коректність запитів, застосовувати правила доступу, виконувати операції з даними та запускати сповіщення після створення заявки [22].

Архітектура онлайн-магазину повинна будуватися за багаторівневою моделлю. На рівні представлення розміщується клієнтський вебінтерфейс, який забезпечує зручне відображення інформації для користувача. На рівні прикладної логіки працює серверний застосунок, який приймає запити та виконує основні правила обробки даних. На рівні зберігання розташовується база даних, де фіксуються відомості про користувачів, товари, категорії, замовлення та позиції замовлень. Окремий інтеграційний рівень утворює Telegram API, який використовується для передавання коротких службових повідомлень менеджеру.

Розподіл функцій між основними компонентами архітектури наведено в таблиці 2.8.

Таблиця 2.8

Розподіл функцій між основними компонентами архітектури

Компонент системи	Функціональне призначення	Результат використання
Клієнтський вебінтерфейс	Відображення каталогу, кошика, форми замовлення та адміністративної панелі	Користувач отримує зручний доступ до функцій магазину
Серверний застосунок	Оброблення запитів, перевірка даних, виконання бізнес-логіки	Торговельні операції виконуються за єдиними правилами
База даних	Збереження товарів, користувачів, категорій, замовлень і позицій покупки	Інформація зберігається в упорядкованій структурі
Модуль авторизації	Контроль входу та перевірка прав доступу	Адміністративні операції захищаються від стороннього використання
Модуль сповіщень	Передавання повідомлення про нову заявку через Telegram	Менеджер швидше отримує інформацію про замовлення

Клієнтський інтерфейс не повинен напряду взаємодіяти з базою даних, оскільки це створює загрози для цілісності та безпеки даних. Всі запити мають проходити через серверний застосунок, який перевіряє права користувача, аналізує вхідні дані та визначає, яку операцію можна виконати. Завдяки цьому, база даних залишається захищеним сховищем, а клієнтська частина працює лише з підготовленими відповідями серверу. Узагальнена архітектура взаємодії клієнтської частини з сервером наведена на рисунку 2.1.

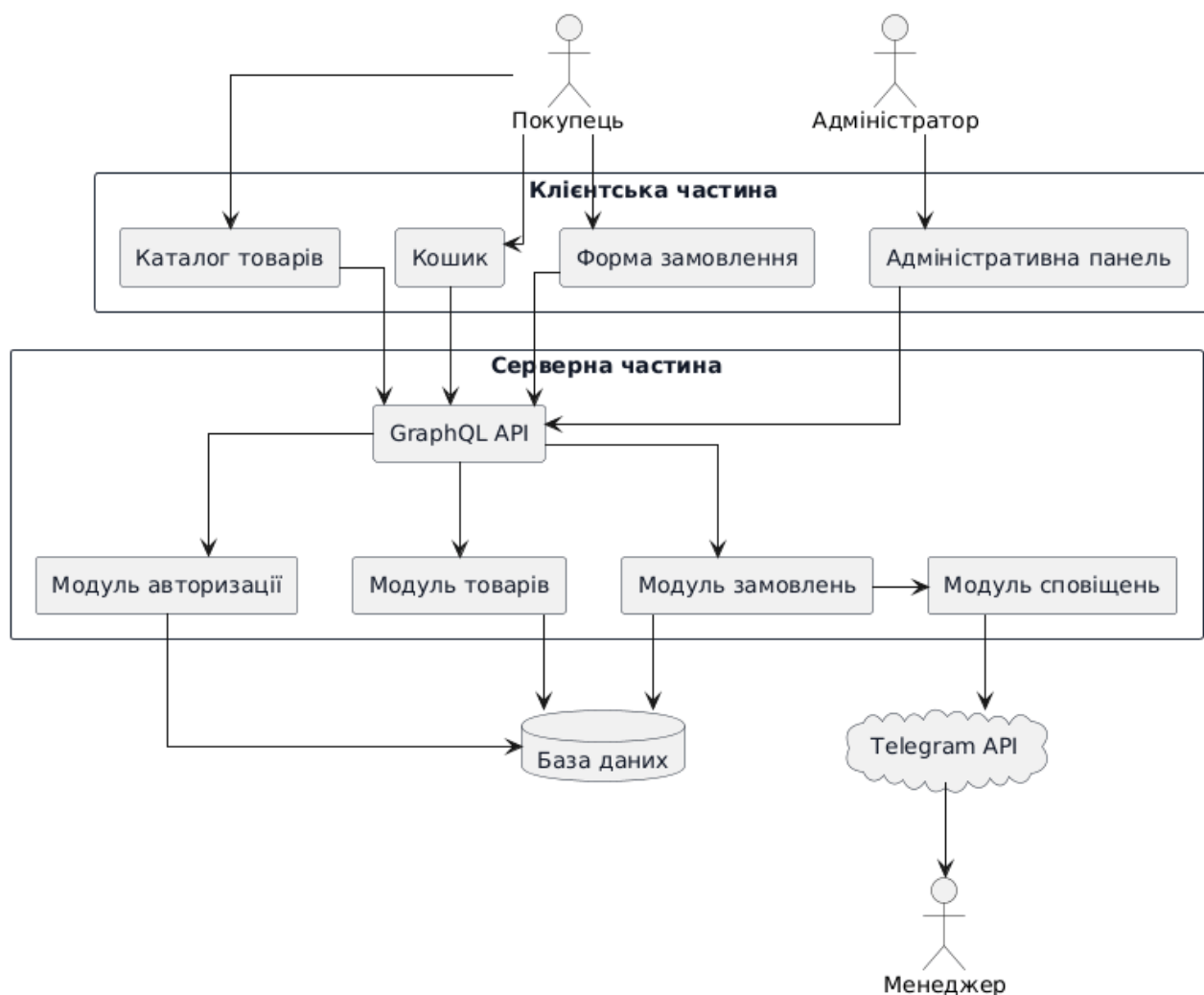


Рис. 2.1 Узагальнена архітектура взаємодії клієнтської частини з сервером

Покупець працює з відкритою частиною магазину, де переглядає товари, формує кошик і передає заявку. Адміністратор використовує окрему захищену панель, через яку виконує службові операції. Всі дії надходять до API взаємодії, після чого серверний застосунок передає запити до відповідних модулів. Модуль товарів відповідає за каталог, модуль замовлень обробляє заявки, модуль авторизації перевіряє права доступу, а модуль сповіщень забезпечує передавання повідомлення через Telegram.

Під час завантаження каталогу, клієнтська частина повинна надсилати запит до серверу, а у відповідь отримувати перелік товарів з базовою інформацією. Для сторінки товару потрібен розширений набір відомостей, оскільки покупець має бачити не лише назву та ціну, але й опис, зображення, категорію та доступність.

Сценарій оформлення замовлення в архітектурі має проходити через кілька взаємопов'язаних операцій. Спочатку клієнтська частина передає на сервер контактні відомості покупця, адресу доставки, склад кошика й обраний спосіб оплати. Далі сервер перевіряє обов'язкові поля, аналізує наявність товарних позицій, обчислює підсумкову суму та створює запис про замовлення. Після успішного збереження заявки серверний застосунок звертається до модуля сповіщень, який формує коротке повідомлення для менеджера та передає його через Telegram API. Основні потоки даних між клієнтською частиною та сервером узагальнено в таблиці 2.9.

Таблиця 2.9

Основні потоки даних між клієнтською частиною та сервером

Потік даних	Зміст передавання	Очікуваний результат
Запит каталогу	Отримання переліку товарних позицій	Покупець бачить актуальний асортимент
Запит картки товару	Отримання детальної інформації про продукцію	Користувач приймає рішення щодо покупки
Надсилання замовлення	Передавання контактних даних, адреси й складу кошика	Сервер створює заявку після перевірки
Запит авторизації	Передавання облікових даних користувача	Система визначає права доступу
Адміністративне редагування	Передавання змін щодо товарів або категорій	Дані каталогу оновлюються
Службове повідомлення	Передавання номера заявки й контакту покупця	Менеджер отримує сигнал про нове звернення

Окремої уваги потребує модель збереження даних. Для онлайн-магазину автозапчастин, базовими інформаційними об'єктами є користувач, категорія, товар, замовлення та позиція замовлення. Користувач може створювати заявки, товар належить до певної категорії, замовлення містить одну або кілька позицій, а кожна позиція пов'язує конкретний товар з кількістю та ціною на момент покупки.

Архітектура повинна враховувати обробку помилок, оскільки під час роботи онлайн-магазину можуть виникати неповні форми, порожній кошик, недоступність товару, завершення сесії користувача або тимчасовий збій зовнішнього сервісу. При введенні некоректних даних, сервер має відхиляти операцію до моменту збереження запису. Якщо помилка пов'язана лише з Telegram-сповіщенням, замовлення не повинно втрачатися, адже головним результатом сценарію є створення заявки в базі даних. Повідомлення менеджера в такій ситуації може бути повторене після відновлення доступу до зовнішнього сервісу [22].

Моделювання архітектури взаємодії клієнтської частини з сервером дає змогу визначити технічну основу майбутньої інформаційної системи онлайн-магазину автозапчастин. Клієнтський рівень забезпечує візуальну взаємодію з покупцем та адміністратором, серверний рівень виконує перевірку та бізнес-логіку, база даних зберігає торговельну інформацію, а Telegram-інтеграція підтримує оперативне службове інформування.

У другому розділі виконано аналітичне обґрунтування інформаційної системи онлайн-магазину автозапчастин з інтеграцією Telegram-сповіщень. Основну увагу зосереджено на визначенні користувацьких сценаріїв, формуванні вимог до каталогу товарів та адміністративної панелі, а також, моделюванні архітектури взаємодії клієнтської частини з сервером. Проведений аналіз дав змогу сформувати логіку майбутньої розробки без переходу до опису готового програмного продукту.

Центральним процесом для онлайн-магазину автозапчастин є оформлення замовлення. Сценарій має охоплювати перегляд каталогу, аналіз товарної позиції, додавання продукції до кошика, заповнення контактних і адресних відомостей, підтвердження заявки та службове інформування менеджера. Особливе значення має підтримка альтернативної поведінки покупця, зокрема зміна складу кошика, повернення до каталогу, виправлення помилок у формі та запобігання створенню порожнього замовлення.

Каталог товарів повинен забезпечувати структуроване представлення асортименту, логічний поділ за категоріями, наявність повної товарної інформації,

зручну навігацію, підтримку кошика та адаптивне відображення сторінок. Для автозапчастин важливими залишаються назва, опис, ціна, зображення, бренд, категорія та доступність продукції.

Адміністративна панель розглянута як службовий компонент, без якого неможливо підтримувати актуальність каталогу та контролювати процес опрацювання заявок. Вона має забезпечувати додавання, редагування, приховування та видалення товарів, керування категоріями, перегляд замовлень, зміну статусів і захищений доступ до службових функцій.

Під час моделювання архітектури взаємодії клієнтської частини із сервером обґрунтовано багаторівневу організацію майбутньої системи. Клієнтський вебінтерфейс має відповідати за відображення каталогу, кошика, форми замовлення та адміністративної панелі. Серверний застосунок повинен виконувати перевірку даних, авторизацію, оброблення замовлень, роботу з товарами та передавання інформації до модуля сповіщень. База даних має зберігати відомості про користувачів, товари, категорії, замовлення та позиції покупки, а Telegram API має забезпечувати швидке інформування менеджера про нову заявку.

Результати другого розділу сформували проєктну основу для практичної реалізації онлайн-магазину автозапчастин. Визначені сценарії, функціональні вимоги та архітектурна модель дають змогу перейти до створення програмного продукту, де клієнтська частина, серверна логіка, база даних, механізм авторизації та Telegram-сповіщення працюватимуть як єдина інформаційна система електронної торгівлі.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ОНЛАЙН-МАГАЗИНУ З TELEGRAM-СПОВІЩЕННЯМИ

3.1 Побудова серверної частини програмного продукту онлайн-магазину

Серверна частина програмного продукту онлайн-магазину автозапчастин реалізована як центральний рівень інформаційної системи, який забезпечує обробку запитів від клієнтського інтерфейсу, взаємодію з базою даних, перевірку прав доступу, опрацювання замовлень та підготовку службових повідомлень для менеджера. Backend-рівень побудовано на основі NestJS, GraphQL, Prisma ORM, PostgreSQL і JWT-авторизації, що відповідає вимогам до модульної веборієнтованої торговельної системи.

Головне призначення серверної частини полягає в тому, щоб відокремити бізнес-логіку онлайн-магазину від клієнтського інтерфейсу. Клієнтська частина відповідає за відображення сторінок, форм та товарних блоків, а серверна частина виконує перевірку запитів, звернення до бази даних та повернення структурованої відповіді. Покупець взаємодіє зі зручним вебінтерфейсом, тоді як всі критично важливі операції виконуються на backend-рівні [23].

Структура серверного застосунку побудована за модульним принципом. Головний модуль застосунку підключає GraphQLModule з ApolloDriver, PrismaModule, AuthModule, AdminModule, BannersModule, ProductsModule, OrdersModule і ShopSettingsModule. Основні модулі серверної частини онлайн-магазину наведено в таблиці 3.1.

Таблиця 3.1

Основні модулі серверної частини онлайн-магазину

Модуль	Призначення	Роль у роботі інформаційної системи
PrismaModule	Організація доступу до бази даних через Prisma ORM	Забезпечує роботу серверної логіки з таблицями PostgreSQL

Продовження таблиці 3.1

AuthModule	Авторизація, реєстрація та отримання даних поточного користувача	Дає змогу розмежувати звичайний і службовий доступ
ProductsModule	Робота з товарами, пошуком, додаванням, редагуванням і видаленням	Формує backend-основу електронного каталогу
OrdersModule	Створення, перегляд і оброблення замовлень користувача	Забезпечує перехід від кошика до збереженої заявки
AdminModule	Службові операції адміністративної частини	Підтримує керування внутрішніми функціями магазину
BannersModule	Робота з банерами клієнтського інтерфейсу	Дає змогу керувати візуальними промоблоками
ShopSettingsModule	Зберігання й отримання налаштувань магазину	Підтримує назву, контакти, валюту, часовий пояс і префікс замовлення

Модульність серверної частини полегшує супровід програмного продукту, оскільки кожен функціональний блок відповідає за окрему групу операцій. Логіка товарів не змішується з логікою замовлень, авторизація відокремлена від каталогу, а налаштування магазину винесені в окремий модуль. Компонентна діаграма серверної частини онлайн-магазину представлена на рисунку 3.1.

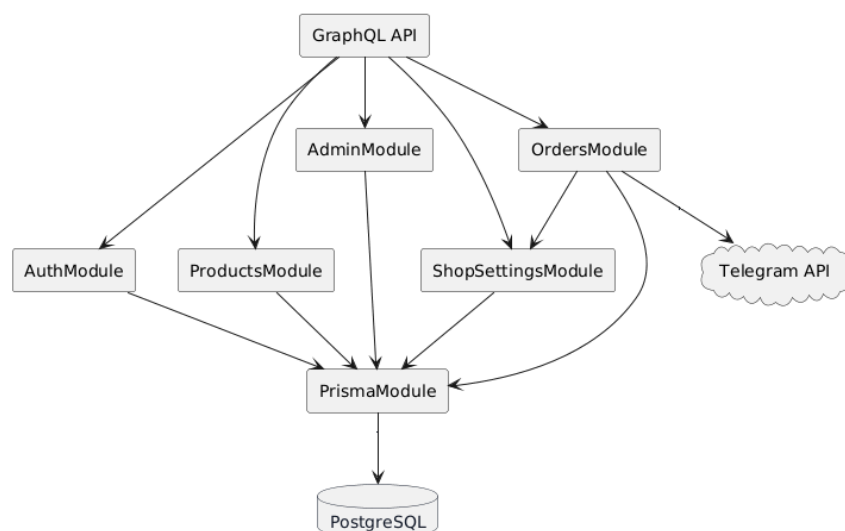


Рис. 3.1 Компонентна діаграма серверної частини онлайн-магазину

Наведена компонентна діаграма відображає загальну логіку backend-рівня. GraphQL API приймає запити від клієнтської частини та передає їх до відповідних модулів. Модуль авторизації відповідає за вхід користувача, модуль товарів обробляє каталог, модуль замовлень створює заявки, адміністративний модуль забезпечує службові операції, а модуль налаштувань надає параметри магазину. PrismaModule виконує роль проміжного рівня доступу до PostgreSQL, завдяки чому серверні модулі не працюють з базою даних напряду [24].

GraphQL в серверній частині використано як основний механізм програмної взаємодії між frontend та backend. Клієнтська частина надсилає запити або мутації, а сервер повертає структуровані відповіді відповідно до GraphQL-схеми. Для каталогу товарів в ProductsResolver передбачено запит products, який приймає параметри активності, кількості записів та пошукового рядка. Адміністративні мутації adminCreateProduct, adminUpdateProduct і adminDeleteProduct захищені GqlAuthGuard, RolesGuard та роллю ADMIN, що забезпечує відокремлення відкритого перегляду товарів від службового керування каталогом.

Авторизаційна частина backend реалізує операції реєстрації, входу та отримання поточного користувача. AuthResolver містить мутації register і login, а запит me захищено GqlAuthGuard, через що отримання персональних даних доступне лише після підтвердження облікового запису. Застосування JWT в загальній структурі програмного продукту дає змогу передавати відомості про користувача між клієнтською та серверною частинами без постійного повторного введення облікових даних.

База даних серверної частини побудована на PostgreSQL та описана через Prisma-схему. В схемі визначено користувача з унікальною електронною поштою, паролем, роллю та зв'язком з замовленнями. Роль користувача має два значення: CUSTOMER і ADMIN, що відповідає потребі розмежування покупця та адміністратора в онлайн-магазині.

Вагому роль в роботі backend відіграє модуль замовлень, оскільки OrdersResolver надає захищені GraphQL-операції myOrders, myOrder та createMyOrder, які працюють лише після проходження GqlAuthGuard. Відповідно,

створення замовлення та перегляд власних заявок пов'язані з авторизованим користувачем.

Сервіс замовлень виконує детальну перевірку даних перед створенням запису. Під час обробки заявки перевіряється наявність хоча б однієї товарної позиції, обов'язковість імені покупця, номера телефону, міста доставки та адреси. Кількість товару нормалізується в діапазоні від однієї до дев'яноста дев'яти одиниць, після чого сервер отримує активні товари з бази даних та блокує оформлення, якщо позиція відсутня, неактивна або має статус OUT_OF_STOCK [25].

Після успішної перевірки, замовлення зберігається разом з позиціями покупки. Для кожної позиції фіксується товар, кількість та вартість на момент оформлення. Загальна сума розраховується серверною частиною, що зменшує ризик підміни комерційних даних на стороні клієнта.

Після створення замовлення сервер формує службовий номер заявки. Для цього використовується префікс з налаштувань магазину, дата створення та фрагмент ідентифікатора замовлення. Далі OrdersService запускає надсилання Telegram-повідомлення, передаючи номер замовлення та телефон покупця. Якщо змінні оточення TELEGRAM_BOT_TOKEN або TELEGRAM_CHAT_ID не заповнені, надсилання не виконується, але створення замовлення не блокується. Текст повідомлення містить фразу про нове замовлення, номер заявки та телефон клієнта. Логіка серверної обробки замовлення узагальнена в таблиці 3.2.

Таблиця 3.2

Логіка серверної обробки замовлення

Етап backend-обробки	Призначення	Результат виконання
Отримання GraphQL-мутації	Приймання заявки від авторизованого користувача	Сервер отримує дані покупця та кошика
Перевірка обов'язкових полів	Контроль імені, телефону, міста та адреси	Некоректна заявка не зберігається
Нормалізація товарних позицій	Перевірка ідентифікаторів і кількості	Замовлення отримує коректний склад

Продовження таблиці 3.2

Контроль доступності	Перевірка активності товару й залишку	Недоступна позиція блокує оформлення
Розрахунок суми	Обчислення вартості на основі даних із бази	Комерційна частина формується на сервері
Збереження заявки	Створення замовлення та пов'язаних позицій	Дані стають доступними для подальшого перегляду
Формування номера	Створення службового ідентифікатора заявки	Менеджер отримує зручний номер замовлення
Telegram-інформування	Передавання короткого повідомлення менеджеру	Відповідальна особа бачить нове звернення

Важливою перевагою серверної реалізації є те, що критична логіка винесена за межі клієнтського інтерфейсу. Покупець може взаємодіяти з кошиком та формою, однак остаточна перевірка складу замовлення, доступності товарів і підсумкової суми виконується на backend-рівні.

Серверна частина забезпечує зв'язок між всіма ключовими елементами онлайн-магазину. Через GraphQL API клієнтська частина отримує дані каталогу та передає заявки, через Prisma ORM backend працює з PostgreSQL, через авторизаційні механізми захищаються службові операції, а через Telegram API менеджер отримує повідомлення про нове замовлення. В результаті backend-рівень виконує роль ядра програмного продукту, яке координує роботу клієнтської частини, бази даних, адміністративних функцій та зовнішнього сервісу сповіщень.

3.2 Розгортання клієнтського інтерфейсу для продажу автозапчастин

Клієнтська частина онлайн-магазину автозапчастин реалізована як вебінтерфейс, через який покупець переглядає асортимент, працює з кошиком, оформлює замовлення, авторизується в системі та переходить до особистого кабінету. Для адміністратора передбачено окрему службову частину, призначену для керування товарами, банерами, замовленнями та налаштуваннями магазину.

Frontend-рівень побудовано на Next.js, React, TypeScript, Tailwind, React Query і Zustand.

Розгортання клієнтського інтерфейсу виконано в структурі frontend, яка відокремлена від серверної частини, але взаємодіє з нею через GraphQL API. В корені клієнтської частини використано файлову маршрутизацію Next.js, де папка app містить маршрути для головної сторінки, каталогу, кошика, сторінки входу, особистого кабінету, порівняння, сторінки товару та адміністративного розділу. Наявність маршрутів cart, catalog, login, product/[slug], account, user, compare і (admin) показує, що інтерфейс охоплює як відкриту купівельну частину, так і захищені службові сторінки. Призначення основних частин клієнтського інтерфейсу наведено в таблиці 3.3.

Таблиця 3.3

Призначення основних частин клієнтського інтерфейсу

Частина інтерфейсу	Реалізаційне призначення	Практичний результат
Головна сторінка	Відображення банерів і популярних товарів	Покупець швидко переходить до товарного асортименту
Каталог	Подання продукції з фільтрацією та пошуком	Користувач добирає потрібні автотовари
Картка товару	Деталізація інформації про окрему позицію	Покупець оцінює товар перед додаванням до кошика
Кошик	Збереження вибраних товарів і оформлення заявки	Замовлення переходить до серверного оброблення
Сторінка входу	Авторизація користувача	Система отримує підставу для захищених операцій
Особистий кабінет	Перегляд користувацької інформації та заявок	Покупець контролює власні звернення
Адміністративна зона	Керування товарами, банерами, замовленнями й налаштуваннями	Працівник магазину підтримує актуальність торговельних даних

Головна сторінка клієнтської частини виконує роль вхідної точки для покупця. В її реалізації використано отримання банерів через `getBanners`, компонент `BannerSlider`, а також, блок `PopularProductsSection`, який формує стартове представлення магазину через промоматеріали та популярні товари.

Загальний каркас застосунку формується у файлі `layout.tsx`. В ньому підключаються глобальні стилі, `QueryProvider`, `Header`, `Footer` та серверне отримання налаштувань магазину для метаданих сторінки. Завдяки цьому, всі сторінки працюють в спільному візуальному оточенні, а шапка та футер залишаються наскрізними елементами інтерфейсу. Структура клієнтського інтерфейсу онлайн-магазину наведена на рисунку 3.2



Рис. 3.2 Структура клієнтського інтерфейсу онлайн-магазину

Клієнтська частина має багаторівневу структуру, де маршрути передають керування сторінковим представленням, представлення використовують віджети та функціональні модулі, а функціональні модулі працюють з предметними моделями та спільним GraphQL-клієнтом.

Каталог товарів реалізовано як сторінку, яка отримує продукцію через `getCatalogProducts`, застосовує пошук та фільтри, а потім передає результат в панель фільтрів та сітку товарів. В параметрах каталогу враховуються пошуковий запит, бренд, в'язкість, тип оливи, доступність, мінімальна та максимальна ціна.

Кошик є центральним елементом купівельного сценарію. На сторінці кошика використано `useCartStore`, `useSessionStore`, мутацію створення власного замовлення, налаштування магазину та форматування суми. Форма оформлення містить поля для ПІБ отримувача, телефону, міста, адреси або відділення, способу оплати та коментаря. Перед надсиланням перевіряється мінімальна довжина ключових полів, а за відсутності токена користувач перенаправляється на сторінку входу. Після успішного створення замовлення кошик очищається, а користувач переходить до особистого кабінету.

Стан кошика зберігається на клієнтському рівні через Zustand з використанням `persist-middleware` та `localStorage`. У сховищі передбачено додавання товару, видалення позиції, оновлення кількості, очищення кошика, підрахунок кількості товарів і загальної вартості. Кількість товару обмежується верхньою межею 99 одиниць, що запобігає випадковому введенню надмірного значення в інтерфейсі. Діаграма послідовності оформлення замовлення наведена на рисунку 3.3.

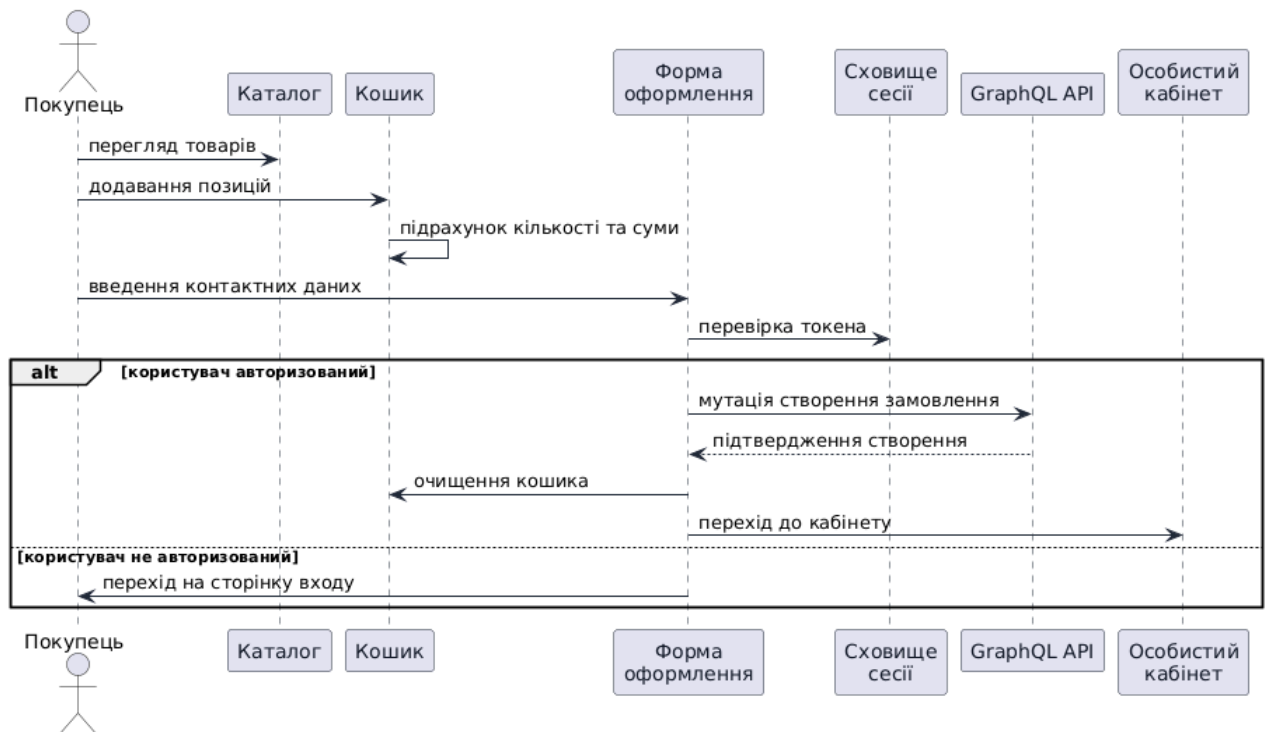


Рис. 3.3 Послідовність оформлення замовлення через клієнтський інтерфейс

Наведена діаграма відображає фактичну логіку купівельного сценарію в клієнтській частині. Покупець спочатку взаємодіє з каталогом, після чого товари накопичуються в кошику. Під час оформлення перевіряється наявність авторизаційного токена. Якщо користувач авторизований, форма передає дані замовлення через GraphQL API. Якщо авторизація відсутня, інтерфейс спрямовує користувача на сторінку входу, що захищає процес створення заявки від неідентифікованих дій.

Сторінка входу винесена в окреме представлення LoginPage, яке підключається через маршрут login, що відповідає загальному поділу frontend на маршрути та views, оскільки маршрут відповідає лише за відкриття сторінки, а основна логіка інтерфейсу зосереджується в окремому представленні.

Збереження сесії реалізовано через Zustand-сховище useSessionStore. В ньому зберігаються токен, дані користувача та стан гідратації, а persist-middleware записує токен і користувача до localStorage. Завдяки цьому після оновлення сторінки застосунок може відновити інформацію про активну сесію та продовжити роботу

без повторного введення облікових даних, якщо збережені відомості залишаються чинними.

Захист адміністративного інтерфейсу реалізовано через компонент AuthGate, який перевіряє наявність токена, виконує запит поточного користувача, очищає невалідну сесію та спрямовує користувача на сторінку входу або до особистого кабінету за відсутності потрібної ролі. Адміністративний layout підключає AdminSidebar, AdminHeader і AuthGate, через що службова зона не відкривається як звичайна сторінка без перевірки доступу. Організація доступу до адміністративного інтерфейсу наведена на рисунку 3.4.

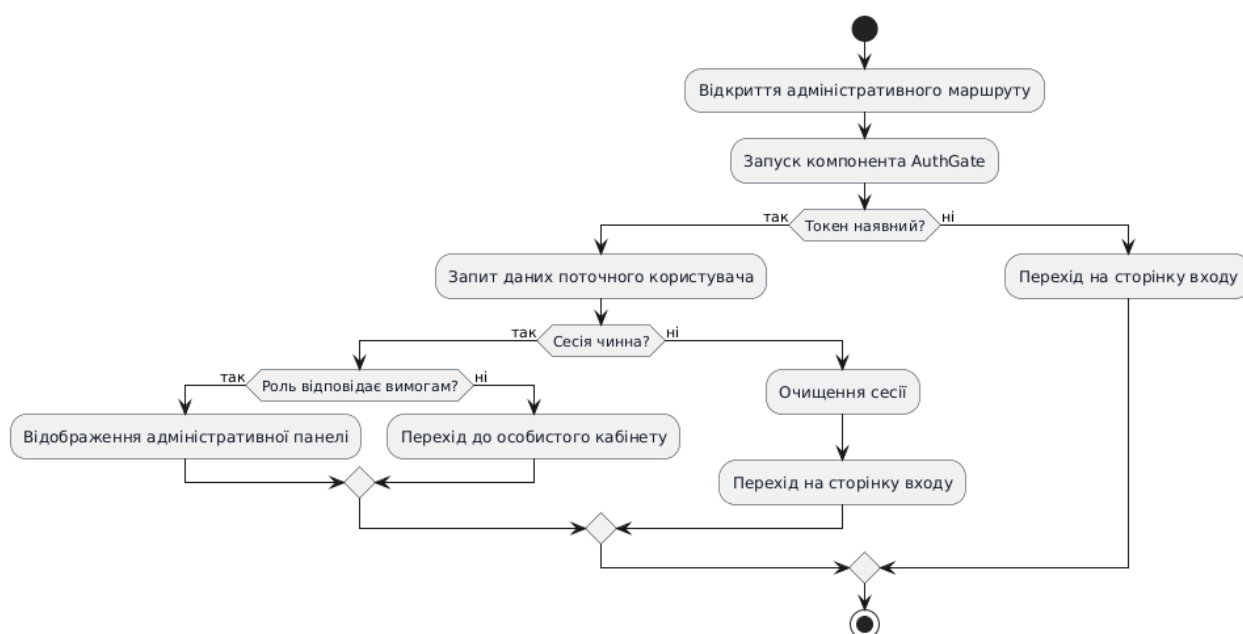


Рис. 3.4 Організація доступу до адміністративного інтерфейсу

Адміністративна зона клієнтської частини охоплює представлення для панелі керування, замовлень, товарів і банерів, що підтверджується структурою папки views. В складі features наявні функціональні частини для адміністративних замовлень, авторизації, банерів, кошика, порівняння, замовлень, товарів і налаштувань магазину.

GraphQL-взаємодія клієнтської частини з backend виконується через спільний клієнт graphqlRequest. Він формує POST-запит до endpoint, передає JSON-тіло з query та variables, автоматично додає заголовок Authorization за наявності

токену, обробляє мережеві помилки та помилки GraphQL-відповіді. В результаті функціональні модулі frontend можуть працювати з сервером через єдину точку доступу, не дублюючи логіку авторизаційного заголовку та перевірки відповіді.

React Query використано як провайдер клієнтських запитів. В QueryProvider створюється QueryClient, для якого встановлено параметри refetchOnWindowFocus: false і staleTime: 30_000. Завдяки цьому інтерфейс не перезавантажує дані під час кожного повернення фокусу до вікна, а отримані відомості вважаються актуальними протягом короткого проміжку часу, що покращує стабільність сприйняття сторінок каталогу, кабінету та адміністративних представлень.

Розгортання клієнтського інтерфейсу забезпечило повний зовнішній рівень онлайн-магазину автозапчастин. Покупець отримує головну сторінку, каталог, фільтрацію, кошик, форму оформлення, авторизацію та особистий кабінет. Адміністратор отримує захищену панель з окремими службовими представленнями. Взаємодія з сервером виконується через GraphQL-клієнт, стан кошика та сесії зберігається через Zustand, а серверні дані керуються через React Query.

3.3 Тестування роботи замовлень з передаванням Telegram-повідомлень

Тестування програмного продукту виконувалося для перевірки працездатності основних функціональних сценаріїв інформаційної системи онлайн-магазину автозапчастин. Перевірка охоплювала клієнтську частину, адміністративну панель, механізм оформлення замовлення, відображення товарів, роботу кошика, авторизацію користувача, перегляд особистого кабінету та надходження Telegram-сповіщення менеджеру. Окрема увага приділялась тому, щоб підтвердити узгоджену роботу frontend-рівня, серверної частини, бази даних та зовнішнього каналу службового інформування.

Перед початком тестування було запущено клієнтську та серверну частини програмного продукту в локальному середовищі. Клієнтський інтерфейс відкривався через веббраузер, а серверна частина забезпечувала обробку запитів,

доступ до даних та передавання повідомлень. Для коректної роботи системи попередньо було підключено базу даних PostgreSQL, виконано міграції Prisma та налаштовано змінні середовища для Telegram-бота. Після запуску, перевірка здійснювалася в послідовності, наближеній до реальної поведінки покупця та адміністратора онлайн-магазину.

Першим етапом стало відкриття головної сторінки магазину. Головна сторінка виконує роль початкової точки взаємодії з покупцем, тому її працездатність має важливе значення для сприйняття програмного продукту. На сторінці відображаються основні навігаційні елементи, пошуковий рядок, кнопка переходу до каталогу, банерна область та блок популярних товарів. Головна сторінка онлайн-магазину автозапчастин наведена на рисунку 3.5

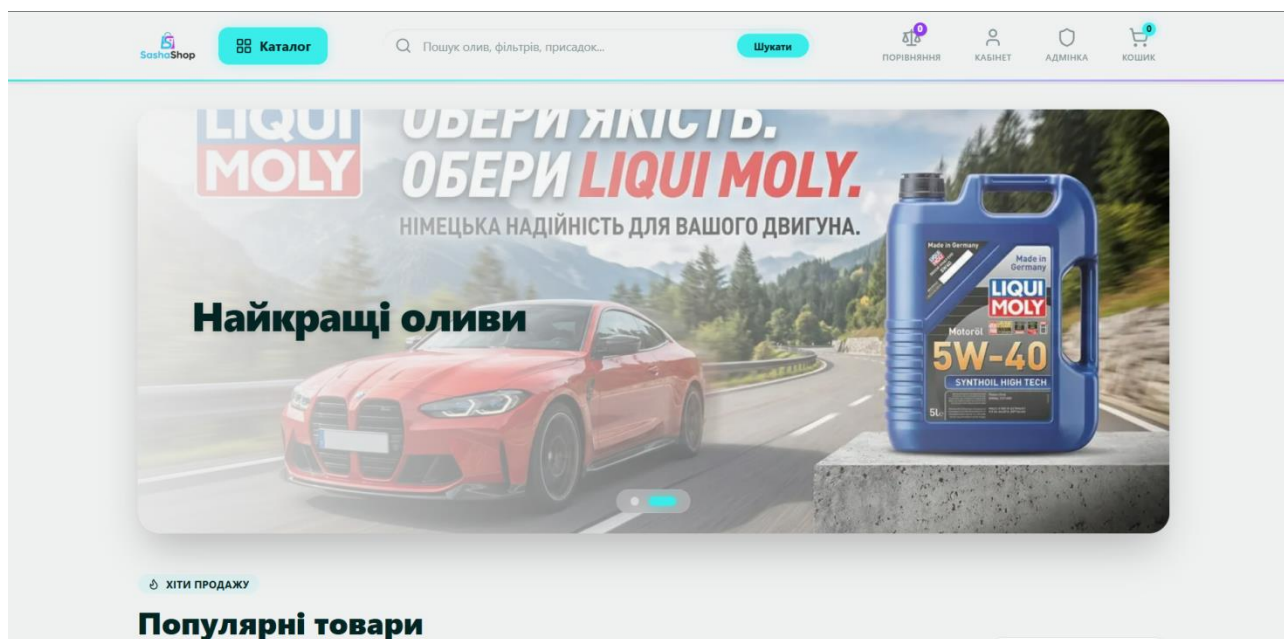


Рис. 3.5 Головна сторінка онлайн-магазину автозапчастин

Відображення головної сторінки підтверджує, що клієнтський інтерфейс успішно завантажує базову структуру застосунку. Наявність шапки сайту, банерної частини та товарних блоків засвідчує коректну роботу маршрутизації, спільних компонентів та візуального оформлення. Для користувача головна сторінка є орієнтиром в системі, тому вона повинна швидко пояснювати призначення магазину та пропонувати очевидний шлях до каталогу.

Після перевірки головної сторінки було здійснено перехід до каталогу товарів. Каталог є одним з ключових елементів онлайн-магазину автозапчастин, оскільки через нього покупець знайомиться з асортиментом та приймає рішення щодо подальшого оформлення покупки. На сторінці каталогу товари представлені у вигляді карток, які містять зображення, назву, вартість, рейтинг, статус доступності та кнопку придбання. Візуальне представлення товарів в картковому форматі спрощує перегляд асортименту та дає змогу швидко порівнювати позиції між собою. Відображення товарів у каталозі онлайн-магазину наведено на рисунку 3.6.

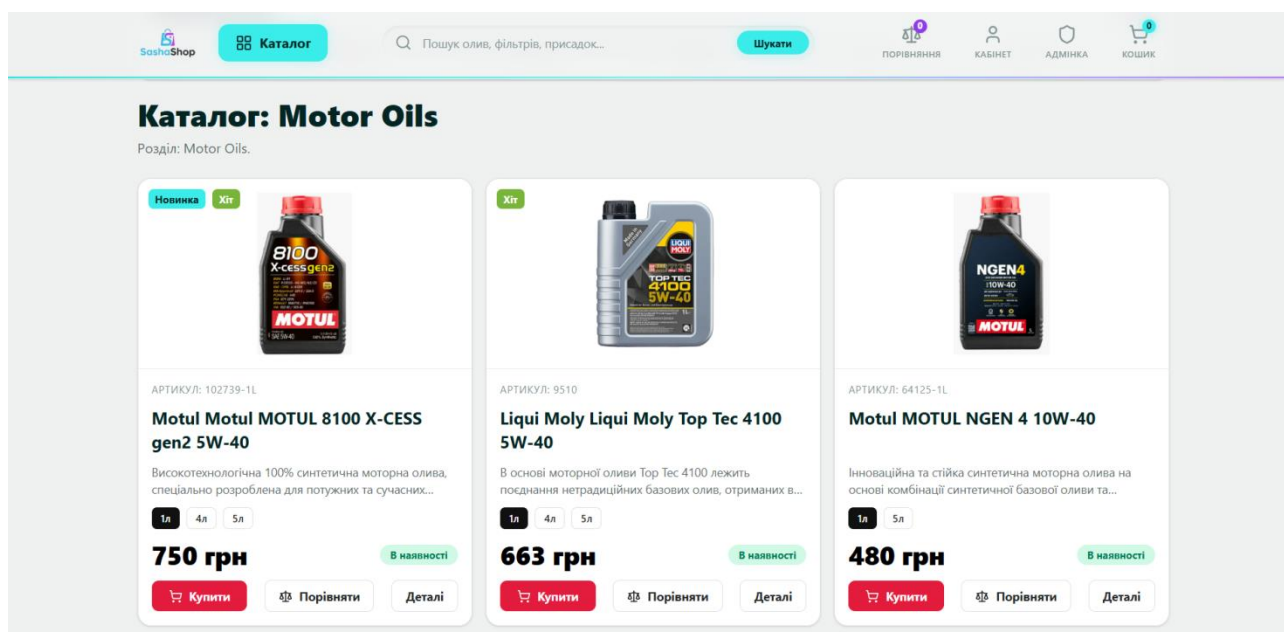


Рис. 3.6 Відображення товарів в каталозі онлайн-магазину

Результат перевірки каталогу показує, що система коректно відображає товарні позиції та забезпечує перехід від інформаційного перегляду до купівельної дії. В розробленому інтерфейсі картка товару одразу надає покупцеві основні відомості, необхідні для первинного вибору.

Окремо було перевірено роботу випадаючого меню каталогу. Навігаційне меню має особливе значення для онлайн-магазину автозапчастин, оскільки асортимент може включати значну кількість категорій і підкатегорій. Меню дає змогу користувачеві перейти до потрібного напряму товарів без перегляду всіх позицій магазину. Випадаюче меню каталогу товарів наведено на рисунку 3.7.

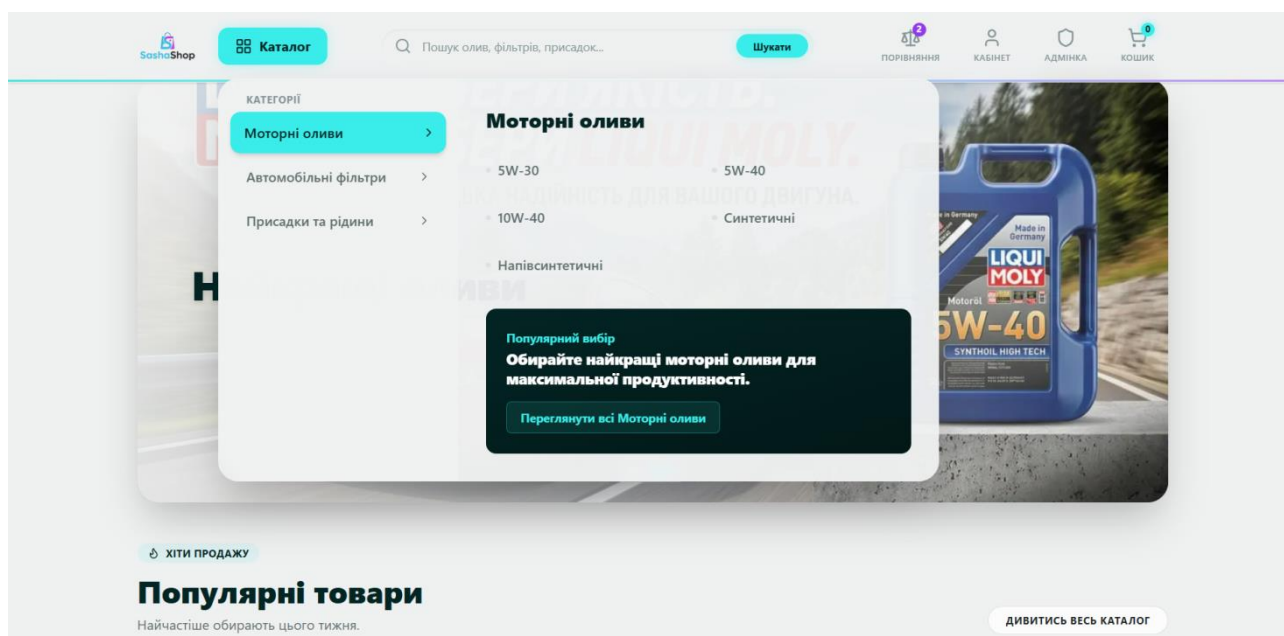


Рис. 3.7 Випадаюче меню каталогу товарів

Наявність випадаючого меню підтверджує, що інтерфейс підтримує логічну організацію асортименту. Для покупця представлена навігація скорочує шлях до потрібної товарної групи, а для магазину створює передумови для подальшого розширення каталогу. Якщо асортимент зростатиме, структуроване меню дозволить зберегти зручність користування без істотного ускладнення головної сторінки.

Після перевірки навігації було протестовано блок популярних товарів. Його наявність на головній сторінці допомагає покупцеві швидше перейти до актуальних або найбільш затребуваних позицій. Блок популярних товарів виконує не лише інформаційну, але й маркетингову функцію, оскільки концентрує увагу користувача на товарах, які магазин прагне показати першими. Кожна позиція в блоці має назву, ціну, статус доступності та кнопку для додавання до покупки. Блок популярних товарів онлайн-магазину наведено на рисунку 3.8.

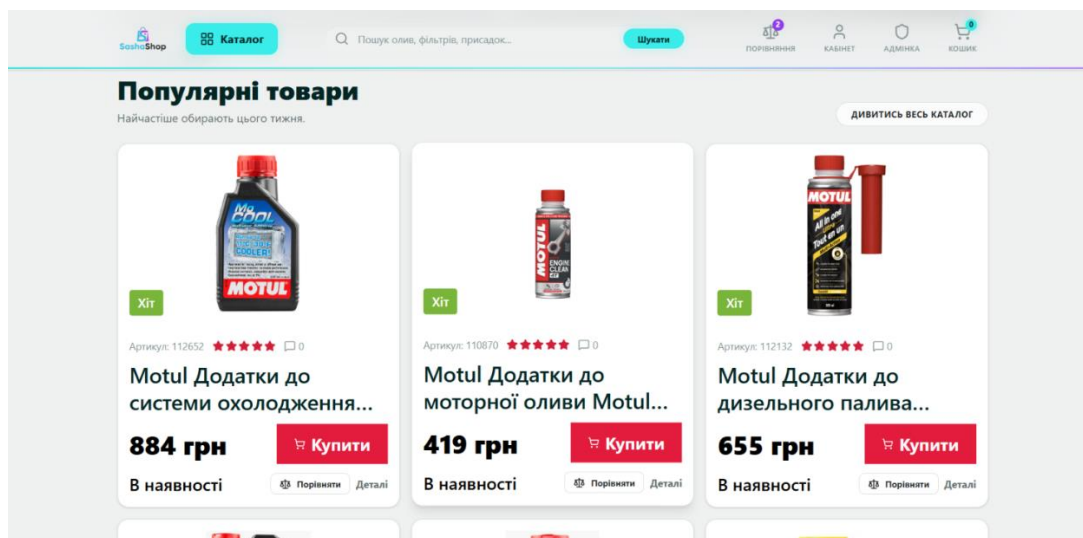


Рис. 3.8 Блок популярних товарів онлайн-магазину

Перевірка блоку популярних товарів показала, що система може формувати окремі товарні добірки для швидкого доступу з головної сторінки. Для покупця це зменшує кількість дій до початку купівельного сценарію. Для адміністратора або власника магазину наявний блок є інструментом просування певних позицій.

Наступним етапом стало тестування сторінки порівняння товарів. В сфері автозапчастин та супутніх товарів покупець часто обирає між кількома позиціями, які мають близьке призначення, але відрізняються ціною, брендом, характеристиками або статусом наявності. Сторінка порівняння дає змогу в одному інтерфейсі переглянути важливі параметри двох товарів та прийняти обґрунтоване рішення щодо покупки. Сторінка порівняння товарів наведена на рисунку 3.9.

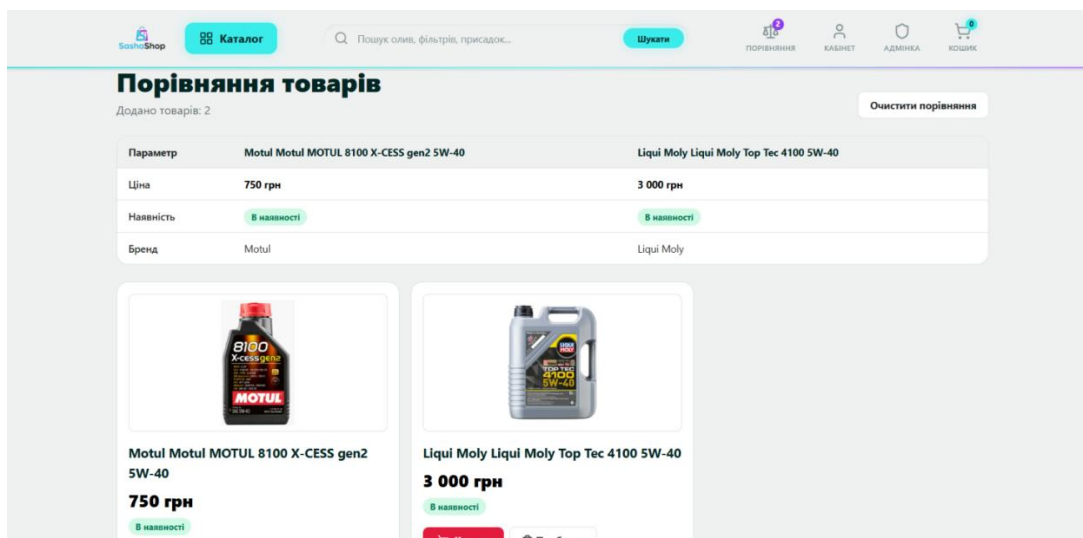


Рис. 3.9 Сторінка порівняння товарів

Результат тестування підтверджує, що порівняння товарів є корисним доповненням до каталогу. Покупець може оцінити різницю між позиціями без перемикання між кількома сторінками товарів. Наявність порівняльного функціоналу підвищує інформативність магазину та зменшує потребу в додатковому зверненні до менеджера.

Після перевірки каталогу та порівняння було протестовано роботу кошика. Кошик є проміжним етапом між вибором товару та створенням замовлення, тому його працездатність безпосередньо впливає на завершення купівельного сценарію. В кошику відображаються додані товари, кількість одиниць, ціна, підсумкова сума та форма оформлення замовлення. Покупець може перевірити склад майбутньої покупки перед надсиланням заявки. Кошик покупця з формою оформлення замовлення наведено на рисунку 3.10.

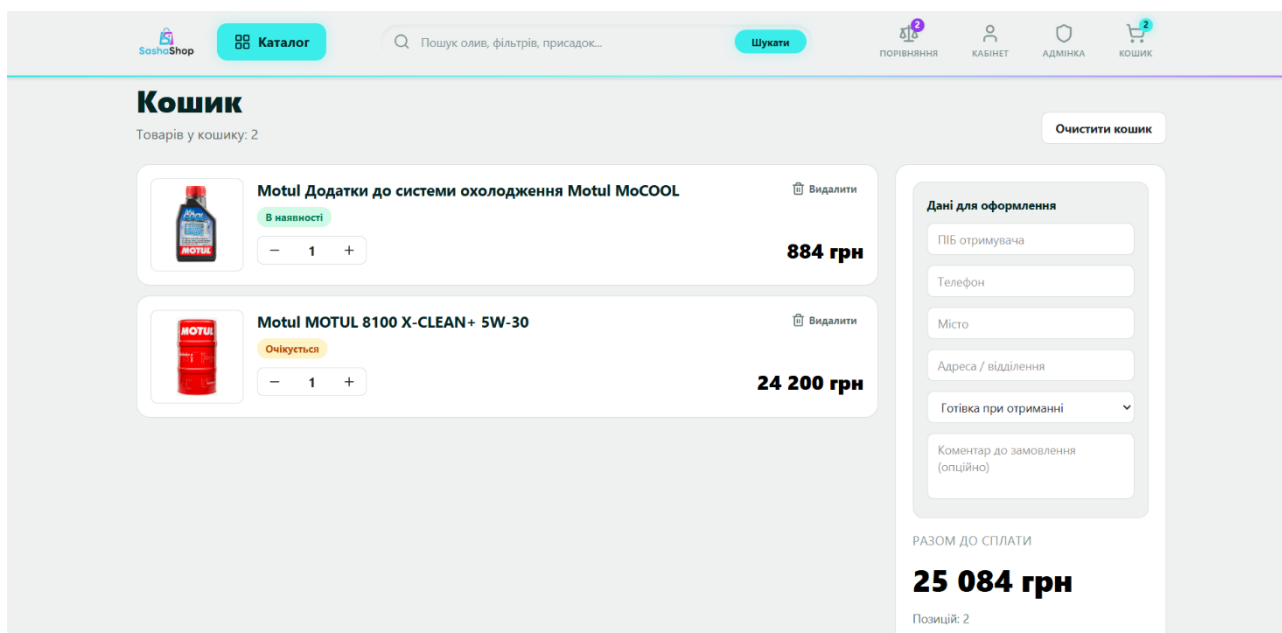


Рис. 3.10 Кошик покупця з формою оформлення замовлення

Перевірка кошика показала, що клієнтський інтерфейс підтримує не лише збереження обраних товарів, а й перехід до оформлення заявки. Наявність форми безпосередньо в кошику скорочує кількість переходів між сторінками та робить процес купівлі більш послідовним. Користувач бачить склад замовлення та одразу вводить контактні та адресні дані.

Під час тестування форми замовлення перевірялися поля для введення контактної інформації, міста, адреси або відділення доставки, способу оплати та коментаря. Важливо, щоб форма не була перевантажена, але водночас містила відомості, достатні для подальшого опрацювання заявки менеджером. Після заповнення форми покупець надсилає замовлення, а серверна частина приймає дані, перевіряє їх та створює запис у базі даних.

Після оформлення замовлення було перевірено роботу Telegram-сповіщення. Інтеграція з Telegram має важливе практичне значення, оскільки менеджер не повинен постійно оновлювати адміністративну панель для виявлення нових заявок. Після створення замовлення серверна частина формує коротке повідомлення та надсилає його у визначений чат. В повідомленні відображається факт появи нового замовлення, його номер та телефон покупця. Telegram-сповіщення про нове замовлення представлено на рисунку 3.11.

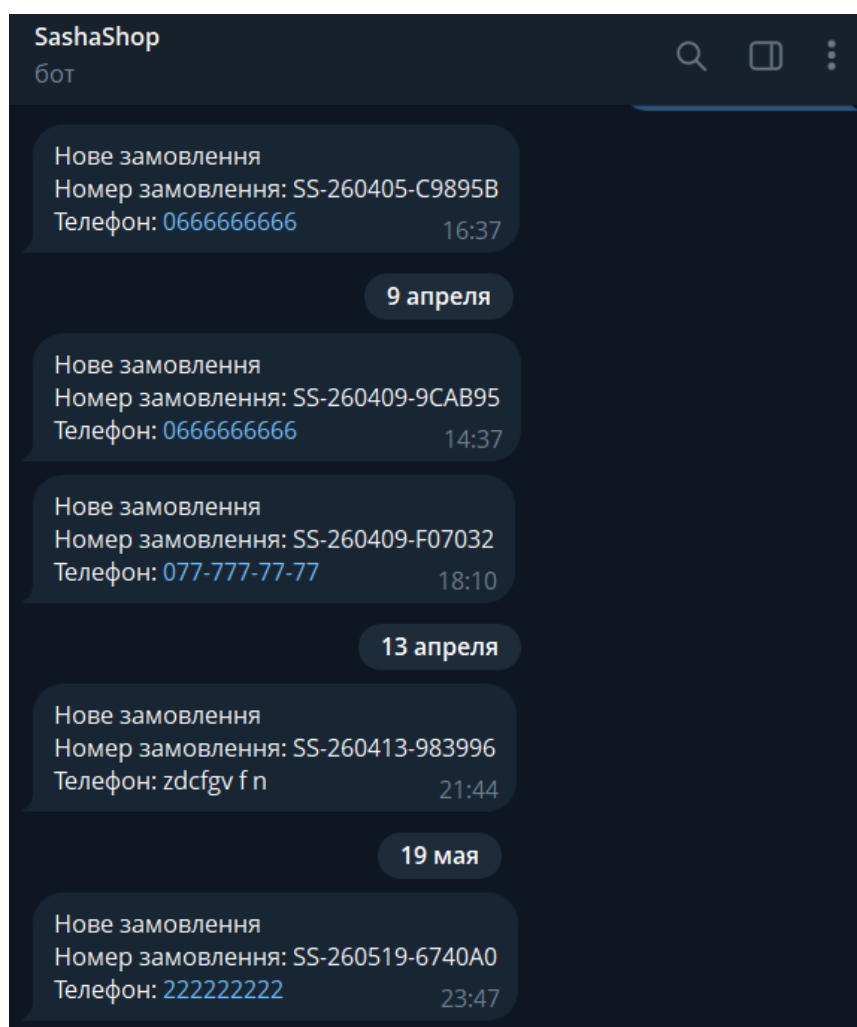


Рис. 3.11 Telegram-сповіщення про нове замовлення

Надходження повідомлення в Telegram підтверджує, що серверна частина успішно взаємодіє із зовнішнім сервісом сповіщень. Для менеджера дане повідомлення є сигналом до швидкої реакції на звернення покупця. Повний склад замовлення може переглядатися в адміністративній частині, а Telegram використовується як швидкий канал первинного інформування.

Далі було перевірено особистий кабінет користувача. Особистий кабінет потрібен для того, щоб покупець мав доступ до власних даних і міг переглядати створені замовлення. На сторінці кабінету відображається інформація про користувача, його роль, доступні дії та перелік останніх заявок. Наявність замовлень в кабінеті підтверджує, що заявка пов'язується з обліковим записом користувача та може бути переглянута після оформлення. Особистий кабінет користувача онлайн-магазин наведено на рисунку 3.12.

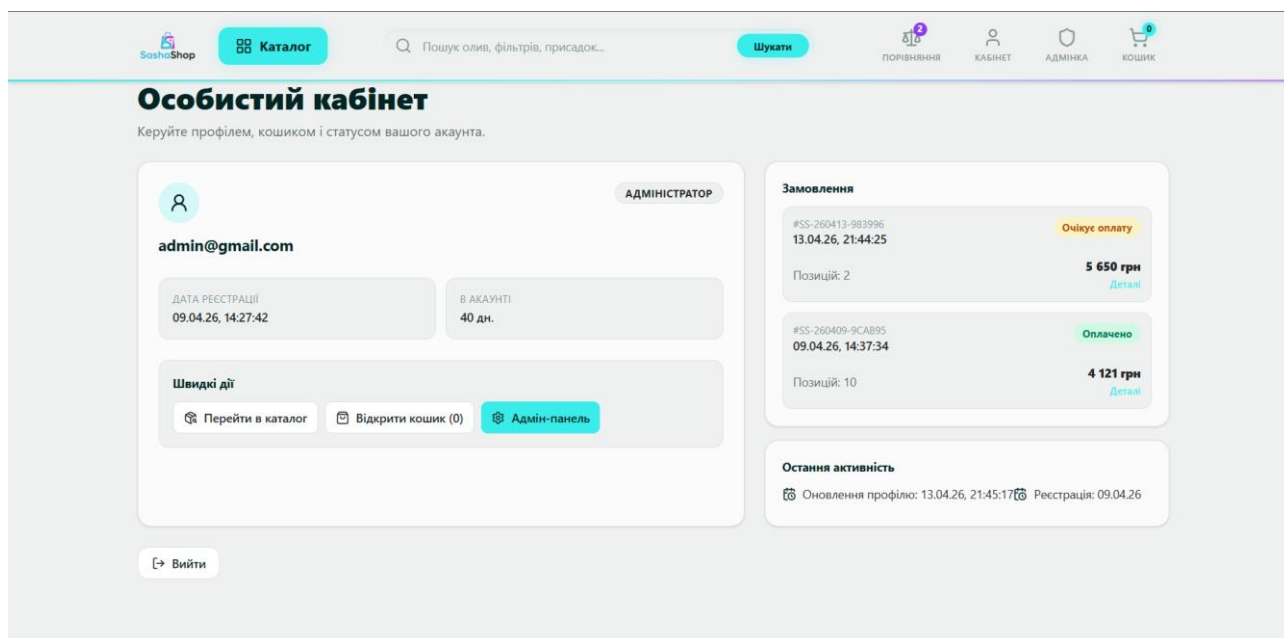


Рис. 3.12 Особистий кабінет користувача онлайн-магазину

Перевірка особистого кабінету показала, що система підтримує персоналізований сценарій роботи з покупцем. Користувач після входу отримує не лише можливість оформити замовлення, але й доступ до історії власних звернень. Для онлайн-магазину це важливо, оскільки покупець може повернутися до попередніх заявок, перевірити суму або статус замовлення. Наявність особистого

кабінету також створює основу для майбутнього розширення функціональності, наприклад повторного замовлення або персональних рекомендацій.

Адміністративна частина призначена для внутрішньої роботи працівника магазину та не повинна бути доступною звичайному покупцеві. Під час тестування було перевірено вхід до адміністративної зони та відображення головного екрана панелі. На головній сторінці адміністративного розділу відображаються узагальнені показники магазину, зокрема кількість користувачів, замовлень, товарів і загальна сума продажів. Головна сторінка адміністративної панелі наведена на рисунку 3.13.

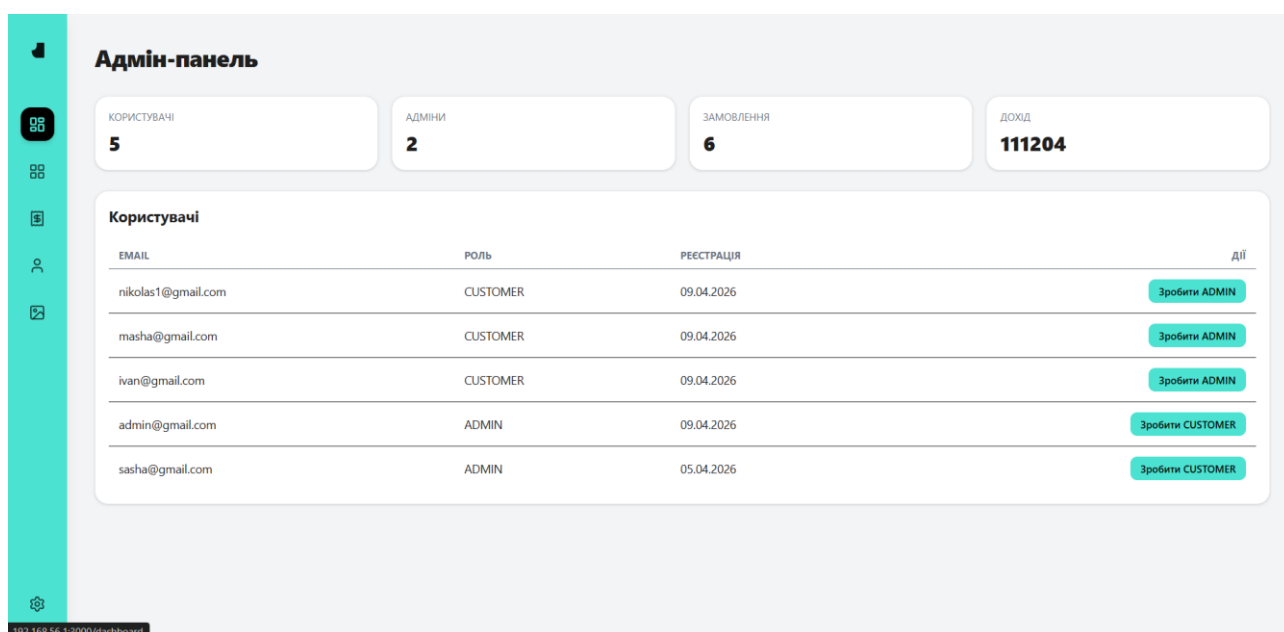
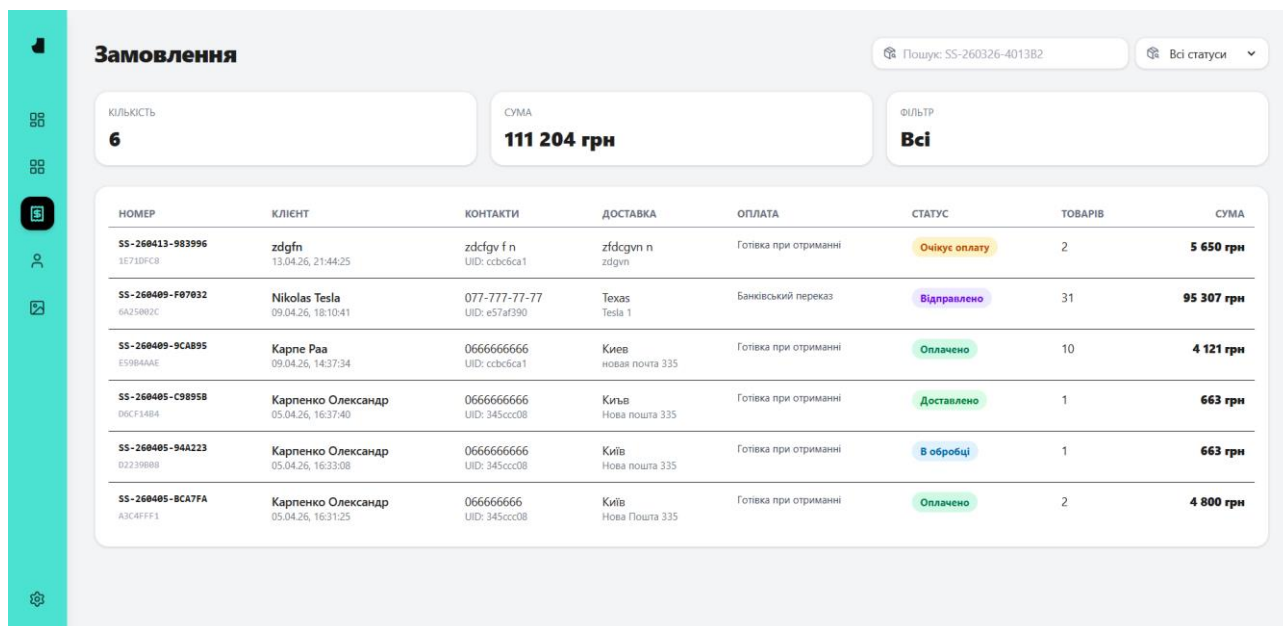


Рис. 3.13 Головна сторінка адміністративної панелі

Головна сторінка адміністративної панелі виконує функцію службового огляду стану системи. Вона дає адміністратору змогу швидко оцінити кількість основних об'єктів магазину та перейти до потрібного розділу. Наявність узагальнених показників робить панель не лише засобом редагування, а й інструментом первинного контролю роботи торговельної платформи.

Після цього було перевірено розділ замовлень в адміністративній частині. В ньому відображаються заявки покупців з номерами, іменами клієнтів, телефонами, містами доставки, способами оплати, статусами, кількістю товарів і сумами. Для

менеджера розділ замовлень є основним інструментом подальшого опрацювання заявок. Перегляд замовлень в адміністративній частині наведено на рисунку 3.14.



НОМЕР	КЛІЄНТ	КОНТАКТИ	ДОСТАВКА	ОПЛАТА	СТАТУС	ТОВАРІВ	СУМА
SS-260413-983996 16710PC8	zdgfn 13.04.26, 21:44:25	zdcfgv f n UID: ccbcbca1	zfdcgvn n zdgvn	Готівка при отриманні	Очікує оплати	2	5 650 грн
SS-260409-F07032 0A25002C	Nikolas Tesla 09.04.26, 18:10:41	077-777-77-77 UID: e57af390	Texas Tesla 1	Банківський переказ	Відправлено	31	95 307 грн
SS-260409-9CAB95 E59B4AAE	Карпе Раа 09.04.26, 14:37:34	0666666666 UID: ccbcbca1	Київ нова пошта 335	Готівка при отриманні	Оплачено	10	4 121 грн
SS-260405-C98958 06CF34B4	Карпенко Олександр 05.04.26, 16:37:40	0666666666 UID: 345ccc08	Київ Нова пошта 335	Готівка при отриманні	Доставлено	1	663 грн
SS-260405-94A223 02239008	Карпенко Олександр 05.04.26, 16:33:08	0666666666 UID: 345ccc08	Київ Нова пошта 335	Готівка при отриманні	В обробці	1	663 грн
SS-260405-8CA7FA A3C9FF1	Карпенко Олександр 05.04.26, 16:31:25	0666666666 UID: 345ccc08	Київ Нова Пошта 335	Готівка при отриманні	Оплачено	2	4 800 грн

Рис. 3.14 Перегляд замовлень в адміністративній частині

Перевірка розділу замовлень підтвердила, що дані, введені покупцем в клієнтському інтерфейсі, успішно зберігаються та стають доступними для адміністратора. Покупець створює заявку в кошику, сервер обробляє її, база даних зберігає результат, а адміністративна панель відображає створене замовлення. Пошук і фільтрація за статусами підвищують зручність роботи з великим обсягом заявок.

Наступним етапом було тестування розділу клієнтів. В ньому адміністратор бачить зареєстрованих користувачів, кількість їхніх замовлень, дату реєстрації, дату оновлення та кнопку переходу до пов'язаних заявок. Вказаний розділ допомагає аналізувати клієнтську базу та швидко переходити від користувача до його історії замовлень. Для торгівельної системи це важливо, оскільки робота з клієнтами не завершується моментом створення першої заявки. Розділ клієнтів в адміністративній панелі представлено на рисунку 3.15.

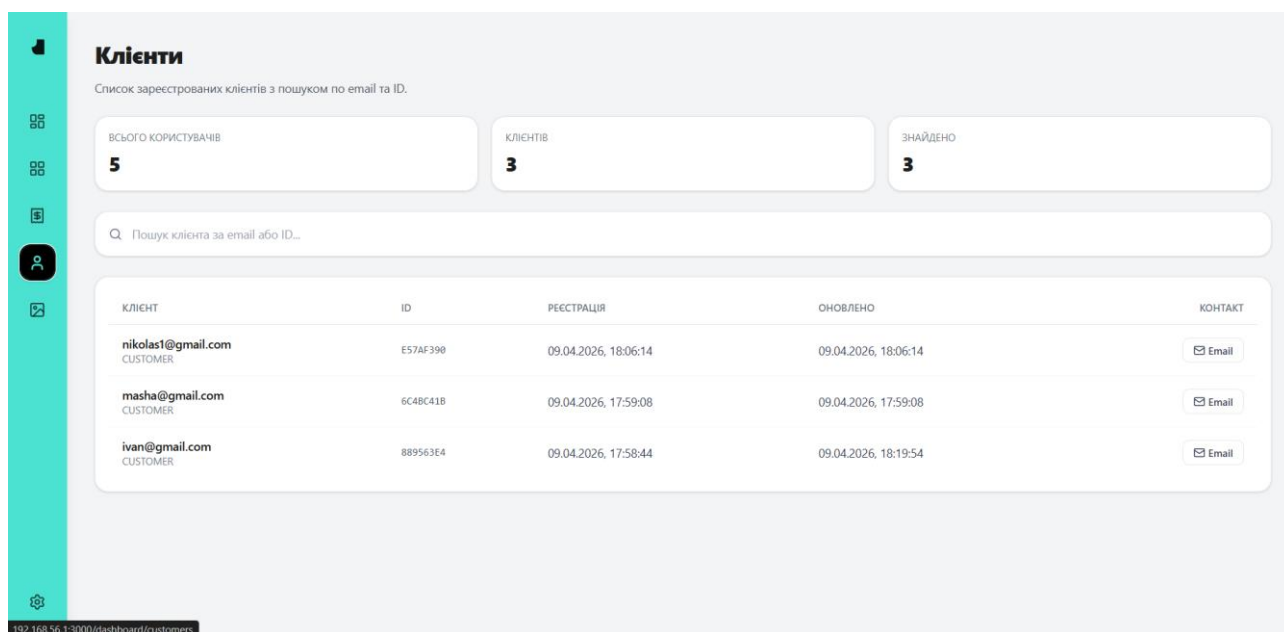


Рис. 3.15 Розділ клієнтів в адміністративній панелі

Наявність розділу клієнтів підтверджує, що адміністративна частина підтримує не лише товарні та замовні операції, але й роботу з користувацькими обліковими записами. Адміністратор може бачити, скільки замовлень має певний користувач, і переходити до їхнього перегляду. Дана функція корисна для контролю повторних звернень, аналізу активності покупців і супроводу клієнтів, які вже взаємодіяли з магазином.

Після перевірки клієнтської бази було протестовано розділ керування товарами. В даному розділі адміністратор бачить перелік товарних позицій, їх бренди, ціни, статуси доступності та службові кнопки редагування або видалення. Нижче розміщено форму редагування товару, через яку можна змінити назву, опис, ціну, категорію, бренд, зображення, активність і статус наявності. Розділ товарів є одним з ключових елементів адміністративної панелі, оскільки саме через нього підтримується актуальність каталогу. Редагування товарної позиції в адміністративній панелі наведено на рисунку 3.16

The screenshot shows the 'Редагування товару' (Edit Item) form in an administrative panel. The form is for the item 'Liqui Moly Top tec 4200 5W-30'. It includes the following fields and options:

- Item Name:** Liqui Moly Top tec 4200 5W-30
- Price:** 83 000
- Availability:** € в наявності (In stock)
- Additives and fluids:** additives-and-fluids
- Group variants:** 0.5
- Antifreeze / Cooling liquid:** ☒ Активний (Active)
- Novinka (New):** ☒ Новинка
- Популярний товар (Popular item):** ☒ Популярний товар

A warning message is displayed at the bottom: 'Призначений для всіх систем охолодження мотоциклів, автомобілів, квадроциклів... коли спостерігається перегрів двигуна або в умовах перегонів для зниження робочої температури двигуна. УВАГА! Деякі види перегонів передбачають використання лише чистої води в якості охолоджуючої рідини, тому перед використанням необхідно проконсультуватися з організаторами перегонів. Компанія Motul не несе відповідальності у випадку дискваліфікації у перегонах через неналежне використання продукту. MoCool не забезпечує захист від замерзання.'

Рис. 3.16 Редагування товарної позиції в адміністративній панелі

Перевірка редагування товару показала, що адміністратор може змінювати основні характеристики товарної позиції без втручання в програмний код або базу даних напряму. Завдяки адміністративній формі працівник магазину може швидко оновити інформацію про найменування, вартість, наявність чи інші параметри, а покупець одразу побачить актуальні відомості в каталозі. Це суттєво спрощує підтримку актуальності асортименту та знижує залежність бізнесу від технічних спеціалістів у повсякденній роботі.

Додатково було перевірено функціональність створення нового товару. Форма додавання містить поля для назви, бренду, ціни, групи товару, кількості, опису, зображення, статусу активності, ознаки популярності та статусу наявності. Кожне поле виконує чітку функцію та забезпечує повноту даних, необхідних для коректного відображення товару в клієнтській частині магазину. Наявність окремої форми створення товару дає змогу адміністратору самостійно розширювати асортимент онлайн-магазину без технічної участі розробника, що є важливою перевагою з точки зору операційної гнучкості. Після заповнення всіх обов'язкових полів та збереження нового запису товар має з'являтися в каталозі клієнтської частини та бути доступним для перегляду і придбання покупцями. Форма створення нового товару наведена на рисунку 3.17.

Управління Товарами
Додавайте та редагуйте товари каталогу.

+ Додати товар

Назва SKU Ціна Є в наявності

Категорія (назва) Категорія (slug) Інше Slug (optional)

Група варіантів (optional, напр. top-tec-4100-5v) Об'єм, літрів (optional) Бренд (optional) В'язкість (optional)

Тип оливи (optional) ☒ Активний ☐ Новинка ☐ Популярний товар

Фото товару (optional)
Выберите файл Файл не выбран

☐ Створити одразу варіанти об'єму (окремі товари з різними SKU: 1л/4л/5л)

Опис Заповніть це поле.

Створити товар

Пошук за назвою, SKU, брендом...

Рис. 3.17 Форма створення нового товару

Результат перевірки створення товару підтверджує, що адміністративна панель забезпечує повний цикл роботи з асортиментом. Адміністратор може не лише редагувати наявні позиції, але й додавати нові товари до системи без залучення розробників. Можливість швидкого додавання позицій підвищує гнучкість програмного продукту та дозволяє оперативно реагувати на зміни в асортименті чи появу нових товарів.

Останнім адміністративним сценарієм стало тестування функціональності керування банерами. Банери є важливим інструментом візуального наповнення головної сторінки та слугують для просування актуальних пропозицій і нових надходжень. В розділі банерів передбачено форму створення банера з полями для введення заголовка, підзаголовка, посилання, завантаження зображення та встановлення статусу активності. Нижче форми відображається перелік наявних банерів з можливістю їх редагування або видалення, що забезпечує повний контроль над контентом головної сторінки. Керування банерами головної сторінки наведено на рисунку 3.18.

Управління Банерами
Створюйте, редагуйте та видаляйте банери для головної сторінки.

Заголовок

Зображення (JPG/PNG/WEBP, до 20MB)
Виберите файл Файл не выбран

Link URL (optional)

Опис (optional)

☒ Активний банер

Створити банер

ЗОБРАЖЕННЯ	ЗАГОЛОВОК	СТАТУС	Дії
	Найкраща якість	Активний	Редагувати Видалити
	Найкращі оливи	Активний	Редагувати Видалити

Рис. 3.18 Керування банерами головної сторінки

Перевірка розділу банерів показала, що адміністративна панель дозволяє змінювати промоматеріали головної сторінки без редагування коду. В рамках онлайн-магазину, дана функція доповнює товарний каталог та робить головну сторінку більш гнучкою для комерційного використання.

Результати тестування підтвердили, що програмний продукт забезпечує повний цикл роботи онлайн-магазину автозапчастин. Покупець може відкрити головну сторінку, перейти до каталогу, скористатися меню категорій, переглянути популярні товари, порівняти позиції, додати продукцію до кошика, заповнити форму замовлення, отримати доступ до особистого кабінету та переглянути власні заявки. Серверна частина приймає замовлення, зберігає його в базі даних і передає службове повідомлення через Telegram.

Адміністративна частина підтвердила працездатність внутрішніх сценаріїв керування магазином. Адміністратор може переглядати загальні показники, контролювати замовлення, працювати з клієнтами, редагувати наявні товари, створювати нові товарні позиції та змінювати банерне наповнення головної сторінки.

В третьому розділі виконано практичний опис реалізації інформаційної системи онлайн-магазину автозапчастин з інтеграцією Telegram-сповіщень.

Розглянуто серверну частину програмного продукту, клієнтський інтерфейс та результати тестування основних сценаріїв роботи. Практична реалізація підтвердила, що система побудована як цілісний вебзастосунок, в якому клієнтська частина, backend-рівень, база даних, адміністративна панель і зовнішній сервіс повідомлень працюють узгоджено.

Під час опису серверної частини визначено роль backend-рівня як центрального компонента програмного продукту. Серверна частина забезпечує оброблення GraphQL-запитів, роботу з товарами, користувачами, замовленнями, авторизацією, налаштуваннями магазину й Telegram-інформуванням. Використання NestJS, Prisma ORM, PostgreSQL і JWT-авторизації дало змогу організувати модульну структуру серверної логіки, відокремити відкриті функції покупця від службових операцій адміністратора та забезпечити надійне збереження торговельних відомостей.

Тестування програмного продукту підтвердило працездатність основних сценаріїв роботи системи. Було перевірено відкриття головної сторінки, відображення каталогу, роботу випадаючого меню, блок популярних товарів, сторінку порівняння, кошик, форму оформлення замовлення, особистий кабінет, адміністративну панель, розділи замовлень, клієнтів, товарів і банерів. Окремо підтверджено надходження Telegram-повідомлення після створення заявки, що засвідчує працездатність інтеграції серверної частини з зовнішнім каналом службового інформування.

Результати практичного розділу свідчать, що розроблений програмний продукт відповідає поставленій меті кваліфікаційної роботи. Онлайн-магазин забезпечує електронне представлення автотоварів, оформлення замовлень, збереження заявок у базі даних, захищене адміністративне керування та оперативне сповіщення менеджера через Telegram.

ВИСНОВКИ

В кваліфікаційній роботі виконано обґрунтування та практичну реалізацію інформаційної системи онлайн-магазину автозапчастин з інтеграцією Telegram-сповіщень. Актуальність розробки зумовлена потребою торговельних підприємств в цифровому представленні асортименту, швидкому прийманні заявок, зручному адмініструванні товарних позицій та своєчасному інформуванні менеджера про нові замовлення.

Онлайн-магазин автозапчастин має розглядатися не лише як вебсайт з товарами, а як повноцінна інформаційна система, яка поєднує електронний каталог, кошик, форму замовлення, авторизацію, адміністративну панель, базу даних та зовнішній канал службового повідомлення. Для покупця важливими є зрозумілий інтерфейс, структурований каталог, доступність інформації про продукцію та швидке оформлення заявки. Для продавця значення мають актуальність асортименту, контроль замовлень, захищене керування даними та оперативна реакція на звернення клієнта.

Під час аналітичного обґрунтування сформовано користувацькі сценарії роботи майбутньої системи, визначено вимоги до каталогу товарів, адміністративної панелі та архітектури взаємодії клієнтської частини з сервером. Передбачено, що покупець повинен мати можливість переглядати товари, користуватися категоріями, додавати продукцію до кошика, заповнювати форму замовлення та отримувати доступ до особистого кабінету. Адміністратор повинен мати змогу керувати товарами, замовленнями, клієнтами, банерами та службовими параметрами магазину.

Практична реалізація програмного продукту підтвердила доцільність обраного технологічного стеку. Серверна частина забезпечує обробку запитів, роботу з базою даних, авторизацію, перевірку замовлень та надсилання повідомлень через Telegram. Клієнтська частина забезпечує візуальну взаємодію користувача з магазином, відображення каталогу, роботу кошика, авторизацію, оформлення замовлень, особистий кабінет та адміністративний інтерфейс.

В результаті тестування перевірено основні сценарії роботи онлайн-магазину, а саме, відкриття головної сторінки, перегляд каталогу, використання меню категорій, роботу блоку популярних товарів, порівняння позицій, додавання продукції до кошика, оформлення заявки, надходження Telegram-повідомлення, перегляд особистого кабінету та роботу адміністративної панелі. Перевірка підтвердила, що створене замовлення зберігається в системі, стає доступним для адміністратора та супроводжується службовим повідомленням менеджеру.

Розроблена інформаційна система має практичну цінність як завершений прототип онлайн-магазину автозапчастин. Вона забезпечує електронне представлення асортименту, приймання замовлень, захищене адміністрування торговельних даних та швидке інформування відповідальної особи через Telegram. В подальшому програмний продукт може бути розширений модулем онлайн-оплати, розширеною фільтрацією за параметрами автомобіля, аналітикою продажів, системою знижок, історією повторних покупок та персоналізованими рекомендаціями для користувачів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Про електронну комерцію : Закон України від 03.09.2015 № 675-VIII // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/675-19>
2. Про захист прав споживачів : Закон України від 10.06.2023 № 3153-IX // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/3153-20>
3. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/2297-17>
4. Іпполітова І. Перспективи розвитку електронної торгівлі в Україні в умовах цифровізації економіки. Економіка та суспільство. 2023. № 47. DOI: <https://doi.org/10.32782/2524-0072/2023-47-18>
5. Іванов К. Електронна комерція: функції, форми та сучасний стан. Актуальні проблеми економіки. 2024. № 6 (276). С. 183–194. DOI: <https://doi.org/10.32752/1993-6788-2024-1-276-183-194>
6. Дубель М. В., Барвінченко О. О. Особливості розвитку електронної комерції в Україні в умовах війни. Підприємництво та інновації. 2023. № 27. С. 16-22.
7. OECD. Enhancing Resilience by Boosting Digital Business Transformation in Ukraine. Paris : OECD Publishing, 2024. 111 p. URL: https://www.oecd.org/en/publications/enhancing-resilience-by-boosting-digital-business-transformation-in-ukraine_4b13b0bb-en.html
8. UNCTAD. Digital Economy Report 2024: Shaping an Environmentally Sustainable and Inclusive Digital Future. Geneva : United Nations, 2024. URL: <https://unctad.org/publication/digital-economy-report-2024>
9. Шостак Л., Ліпич Л., Павлова С. Вплив електронної комерції на інновації бізнес-моделей та нові технології продажу на ринку. Київський економічний науковий журнал. 2025. № 9. С. 302-309.

10. Nielsen Norman Group. UX Guidelines for Ecommerce Product Pages. URL: <https://www.nngroup.com/articles/ecommerce-product-pages/>
11. Baymard Institute. E-Commerce Cart & Checkout Usability Research. URL: <https://baymard.com/research/checkout-usability>
12. Next.js Documentation. App Router. URL: <https://nextjs.org/docs/app>
13. React Documentation. Passing Props to a Component. URL: <https://react.dev/learn/passing-props-to-a-component>
14. TypeScript Documentation. The TypeScript Handbook. URL: <https://www.typescriptlang.org/docs/handbook/intro.html>
15. NestJS Documentation. Authentication. URL: <https://docs.nestjs.com/security/authentication>
16. GraphQL Foundation. Queries and Mutations. URL: <https://graphql.org/learn/queries/>
17. Prisma Documentation. PostgreSQL Database Connector. URL: <https://www.prisma.io/docs/orm/core-concepts/supported-databases/postgresql>
18. PostgreSQL Global Development Group. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/current/>
19. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT). RFC 7519. Internet Engineering Task Force, 2015. URL: <https://datatracker.ietf.org/doc/html/rfc7519>
20. Telegram. Telegram Bot API. URL: <https://core.telegram.org/bots/api>
21. Zustand Documentation. Persist Middleware. URL: <https://zustand.docs.pmnd.rs/reference/middlewares/persist>
22. TanStack Query Documentation. Overview. URL: <https://tanstack.com/query/latest/docs/framework/react/overview>
23. Tailwind CSS Documentation. Styling with Utility Classes. URL: <https://tailwindcss.com/docs/styling-with-utility-classes>
24. SashaShop : e-commerce project repository / GitHub. URL: <https://github.com/ant0-nn/Sasha-Shop>

25. ISO/IEC/IEEE 29119-1:2022. Software and systems engineering – Software testing – Part 1: General concepts. Geneva : International Organization for Standardization, 2022. URL: <https://www.iso.org/standard/81291.html>

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ
(Презентація)

ІНФОРМАЦІЙНА СИСТЕМА ОНЛАЙН-МАГАЗИНУ АВТОЗАПЧАСТИН З ІНТЕГРАЦІЄЮ TELEGRAM-СПОВІЩЕНЬ

Виконав: Олександр КАРПЕНКО гр. ІСД-41

Керівник: Оксана ТКАЛЕНКО, к.т.н., доцент

Київ, 2026

АКТУАЛЬНІСТЬ ТА МЕТА РОБОТИ

Слайд 2 з 11

ПРОБЛЕМА

- Ручне оброблення замовлень — великий ризик помилок
- Менеджер змушений постійно моніторити адмін-панель
- Неузгодженість інформації про залишки товарів
- Затримки у відповідях клієнтам — втрата продажів
- Відсутність централізованого керування каталогом

МЕТА РОБОТИ

Розробка інформаційної системи онлайн-магазину автозапчастин з інтеграцією Telegram-сповіщень

ЗАВДАННЯ

- Аналіз цифровізації торгівлі автозапчастинами
- Визначення вимог та сценаріїв роботи
- Проектування архітектури системи
- Реалізація frontend, backend, БД + Telegram
- Функціональне тестування системи

ВИХІДНІ ДАНІ ТА ЗМІСТ РОБОТИ

Слайд 3 з 11

Вихідні дані кваліфікаційної роботи

- 1 Принципи побудови та функціонування онлайн-магазинів автотоварів
- 2 Принципи інтеграції месенджерів у вебзастосунки
- 3 Науково-технічна література

ЗМІСТ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

- розділ 1** Аналіз передумов цифровізації торгівлі автозапчастинами та огляд технологій розробки веборієнтованих платформ
- розділ 2** Аналітичне обґрунтування вимог до інформаційної системи та моделювання архітектури
- розділ 3** Практична реалізація онлайн-магазину з Telegram-сповіщеннями та тестування

Структура: 3 розділи · 18 таблиць · 19 рисунків · 25 джерел · 62 сторінок · Демонстраційні матеріали

ТЕХНОЛОГІЧНИЙ СТЕК

Слайд 4 з 11

FRONTEND

- Next.js + React
- TypeScript
- Tailwind CSS
- React Query
- Zustand

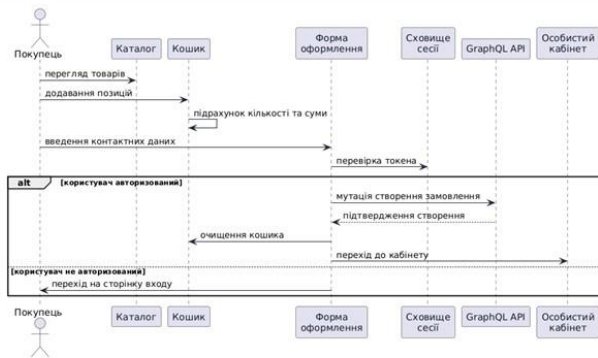
BACKEND

- NestJS (Node.js)
- GraphQL + Apollo
- JWT авторизація
- Prisma ORM
- PostgreSQL

ІНТЕГРАЦІЯ

- Telegram Bot API
- HTTP POST запит
- Миттєві сповіщення
- Env-конфігурація
- Graceful fallback

Повний стек від клієнтського інтерфейсу до зовнішнього каналу сповіщень — єдиний програмний комплекс



Багаторівнева архітектура: клієнт → GraphQL API → бізнес-логіка → БД → Telegram

КОМПОНЕНТИ АРХИТЕКТУРИ

Клієнтський вебінтерфейс

Каталог, кошик, форма замовлення, адмін-панель

Серверний застосунок

GraphQL API, перевірка прав, бізнес-логіка

База даних PostgreSQL

Товари, категорії, замовлення, користувачі

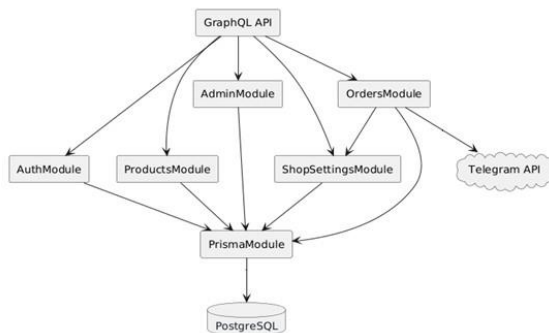
Модуль авторизації JWT

Захист службових функцій, ролі

Telegram Bot API

Миттєве сповіщення менеджера

СЕРВЕРНА ЧАСТИНА (BACKEND)



Лоріка Telegram-сповіщення:

Замовлення збережено → Формується номер заявки → OrdersService → Telegram Bot API → Чат менеджера (якщо TELEGRAM_BOT_TOKEN не налаштований — замовлення НЕ блокується)

ОСНОВНІ МОДУЛІ

AuthModule — реєстрація, вхід, JWT

ProductsModule — каталог товарів

OrdersModule — замовлення

AdminModule — службові операції

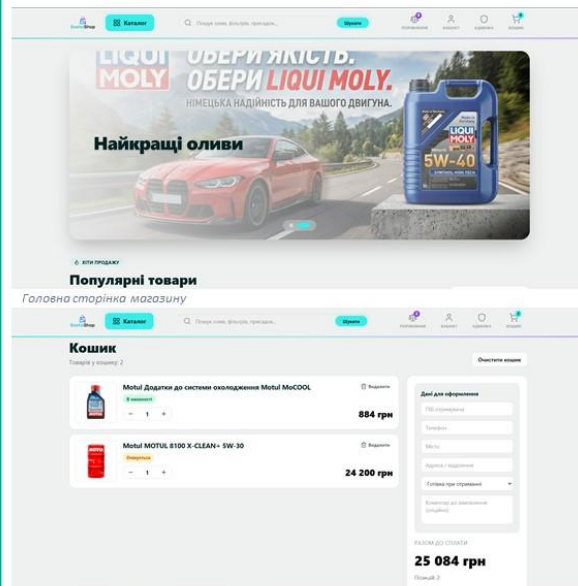
BannersModule — банери сторінок

ShopSettingsModule — налаштування

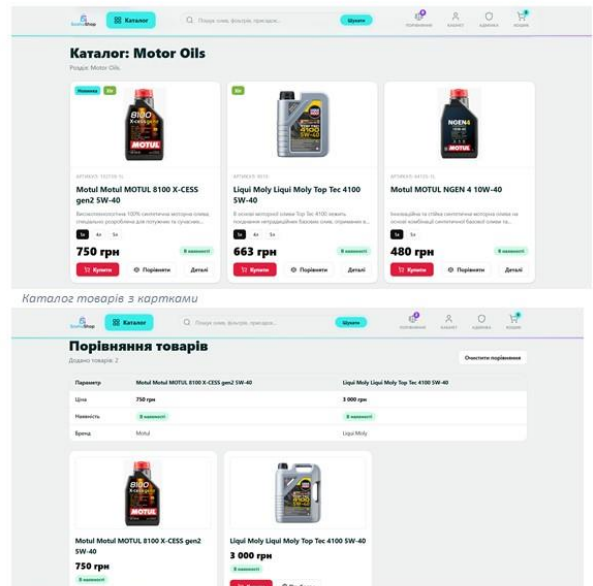
PrismaModule — доступ до PostgreSQL

КЛІЄНТСЬКИЙ ІНТЕРФЕЙС

Слайд 7 з 11



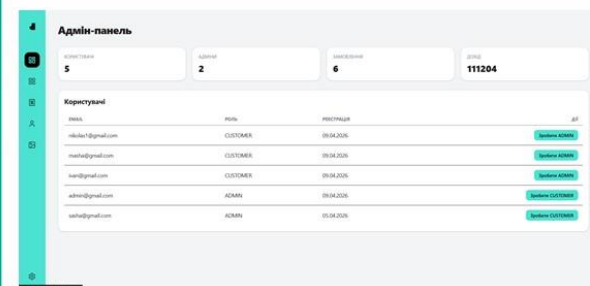
Кошик та форма оформлення замовлення



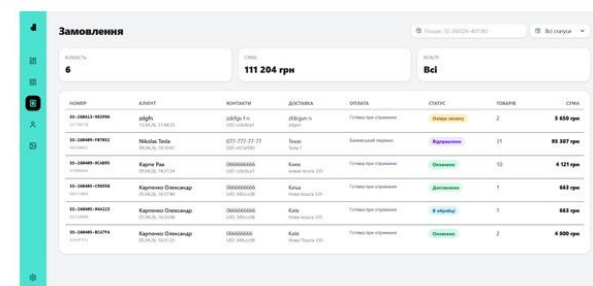
Сторінка порівняння товарів

АДМІНІСТРАТИВНА ПАНЕЛЬ

Слайд 8 з 11



Головна сторінка адмін-панелі (статистика)



Перегляд замовлень в адмін-панелі

ФУНКЦІЇ АДМІНІСТРАТОРА:

Товари

Додавання, редагування, приховування, видалення позицій каталогу

Категорії

Створення та редагування груп автозапчастин

Замовлення

Перегляд, завантаження, зміна статусів замовлення

Клієнти

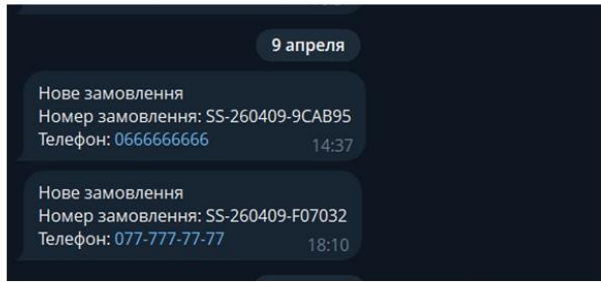
Контроль облікових записів, перегляд замовлень клієнта

Банери

Керування промоматеріалом головної сторінки

Налаштування

Назва магазину, контакти, валюта, часовий пояс



Telegram-сповіщення після оформлення замовлення

ПЕРЕВІРЕНІ СЦЕНАРІЇ

- ✓ Відкриття головної сторінки
- ✓ Перегляд каталогу та фільтрація
- ✓ Порівняння товарів
- ✓ Кошик та форма замовлення
- ✓ Авторизація та особистий кабінет
- ✓ Надходження Telegram-повідомлення
- ✓ Адміністративна панель (всі розділи)

ПРОЦЕС ОФОРМЛЕННЯ ЗАМОВЛЕННЯ:



ВИСНОВКИ

Результати кваліфікаційної роботи

01

Теоретична база

Досліджено передумови цифровізації торгівлі автозапчастинами, охарактеризовано функціональні можливості онлайн-магазинів та сучасний технологічний стек для їх розробки.

02

Аналіз та проектування

Визначено користувацькі сценарії, сформовано вимоги до каталогу та адмін-панелі, змодельовано багаторівневу архітектуру системи.

03

Практична реалізація

Розроблено повноцінний програмний продукт: Next.js frontend, NestJS backend (GraphQL), PostgreSQL + Prisma ORM, JWT авторизація, Telegram Bot API.

04

Тестування

Підтверджено коректну роботу всіх сценаріїв: каталог, кошик, замовлення, авторизація, адмін-панель та Telegram-сповіщення менеджера.

Розроблена ІС є завершеним прототипом онлайн-магазину автозапчастин, придатним для подальшого розширення модулями онлайн-оплати, аналітики продажів та персоналізованих рекомендацій.

Апробація результатів

Слайд 11 з 11

01

III «ТЕХНОЛОГІЧНІ ГОРИЗОНТИ: ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ТЕХНОЛОГІЧНОГО ПРОГРЕСУ УКРАЇНИ І СВІТУ»

Особливості розробки онлайн-магазинів

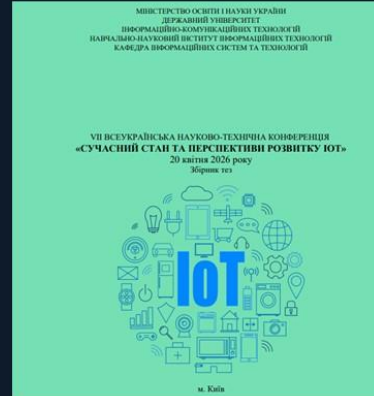


ДУІКТ, Київ, 18 листопада 2025р., С. 197-198.

02

VII МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ»

Аналіз типів сповіщень у Telegram



ДУІКТ, Київ 20 квітня 2026р., С. 5-6.

Дякую за увагу