

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Відокремлений структурний підрозділ
«Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»
Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

**на тему: Розробка інформаційної системи електронної комерції
для реалізації товарів для кемпінгу «PeakPoint»**

Виконав: студент IV курсу, групи КН-423

спеціальності: 122 Комп'ютерні науки

Олександр ГУЦАЛ

Керівник

Галина РАДЧИК

Рецензент

(ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук
Освітньо-професійний ступінь «фаховий молодший бакалавр»
Спеціальність 122 Комп'ютерні науки
Галузь знань 12 Інформаційні технології

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук

_____ Галина МАРЦІЯШ

« 02 » березня 2026 року

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

_____ Гуцалу Олександрю Арсеновичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка інформаційної системи електронної комерції для реалізації товарів для кемпінгу «PeakPoint»

керівник роботи _____ Радчик Галина Іванівна,

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студентом роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мови програмування: Java, JavaScript; фреймворки: Spring Boot, Hibernate; бібліотека React, Vite, SCSS, стандарти IEEE 29148-2018, IEEE 29119, ДСТУ 8302:2015.

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

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу інформаційної системи

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

- 2 Розробка технічного та робочого проєкту
 - 2.1 Розробка структури інформаційної системи
 - 2.2 Створення та верстка сторінок інформаційної системи
 - 2.3 Розробка структури бази даних інформаційної системи
 - 2.4 Програмування інформаційної системи
 - 2.4.1 Написання клієнтської частини
 - 2.4.2 Написання серверної частини
 - 2.5 Тестування інформаційної системи
- 3 Спеціальний розділ
 - 3.1 Інструкція з розгортання інформаційної системи
 - 3.2 Інструкція з наповнення інформаційної системи
 - 3.3 Інструкція з популяризації та підтримки інформаційної системи
- 4 Економічний розділ
 - 4.1 Визначення стадій технологічного процесу та загальної тривалості

проведення НДР

- 4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи
- 4.3 Розрахунок витрат на електроенергію
- 4.4 Розрахунок суми амортизаційних відрахувань
- 4.5 Обчислення накладних витрат
- 4.6 Складання кошторису витрат та визначення собівартості вебсайту
- 4.7 Розрахунок ціни НДР
- 4.8 Визначення економічної ефективності і терміну окупності капітальних

вкладень

- 5 Охорона праці, техніка безпеки та екологічні вимоги
 - 5.1 Гарантія прав на охорону праці
 - 5.2 Засоби і заходи забезпечення пожежної безпеки виробничого об'єкта
- 6 Висновки

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – інформаційної системи «PeakPoint».

5. Перелік графічного матеріалу:

- 1. Схема структурна інформаційної системи
- 2. UML-діаграма варіантів використання
- 3. ER-діаграма бази даних інформаційної системи
- 4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Генадій ГОРЯЧЕК		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	
2	Збір і узагальнення інформації	01.05.2026	
3	Написання першого розділу	15.05.2026	
4	Розробка технічного та робочого проекту	29.05.2026	
5	Написання спеціального розділу	05.06.2026	
6	Розрахунок економічної частини	08.06.2026	
7	Написання розділу охорони праці	09.06.2026	
8	Виконання графічної частини	10.06.2026	
9	Оформлення пояснювальної записки	11.06.2026	
10	Погодження нормоконтролю	12.06.2026	
11	Попередній захист кваліфікаційної роботи	.06.2026	
12	Захист кваліфікаційної роботи	.06.2026	

7. Дата видачі завдання: 05 березня 2026 р.

Студентка

(підпис)

Олександр ГУЦАЛ

Керівник роботи

(підпис)

Галина РАДЧИК

ЗМІСТ

Анотація	8
Abstract	9
Вступ	10
1 Загальний розділ.....	12
1.1 Аналітичний огляд існуючих рішень	12
1.1.1 Інформаційна система «Gorgany».....	12
1.1.2 Інформаційна система «Alp»	13
1.1.3 Інформаційна система «Tramp».....	14
1.1.4 Інформаційна система «Перевал»	14
1.2 Технічне завдання	15
1.2.1 Найменування та область застосування.....	15
1.2.2 Призначення розробки	16
1.2.3 Вимоги до функціоналу інформаційної системи.....	16
1.2.4 Вимоги до програмної документації	19
1.2.5 Техніко-економічні показники	19
1.2.6 Стадії та етапи розробки інформаційної системи	20
1.2.7 Порядок тестування та прийому.....	20
2 Розробка технічного та робочого проєкту	21
2.1 Розробка структури інформаційної системи.....	21
2.2 Створення та верстка сторінок інформаційної системи.....	23
2.3 Розробка структури бази даних інформаційної системи	24
2.4 Програмування інформаційної системи.....	33
2.4.1 Написання клієнтської частини.....	33
2.4.2 Написання серверної частини.....	34
2.5 Тестування інформаційної системи.....	37
2.5.1 Тестування автентифікації та управління профілем	37
2.5.2 Тестування публічного каталогу та пошуку	38

					2026.КВР.122.421.07.00.00 ПЗ			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Гуцал О.А.</i>			Розробка інформаційної системи електронної комерції для реалізації товарів для кемпінгу «PeakPoint» Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Радчик Г.І.</i>				5	107	
<i>Реценз.</i>						ВСП ТФК ТНТУ КН-421 м. Тернопіль		
<i>Н. Контр.</i>		<i>Приймак В.А.</i>						
<i>Затверд.</i>								

2.5.3	Тестування кошика та оформлення замовлення.....	41
2.5.4	Тестування адміністративної панелі	43
2.5.5	Тестування списку бажань та відгуків та адаптивності	44
3	Спеціальний розділ	46
3.1	Інструкція з розгортання інформаційної системи	46
3.1.1	Вимоги до серверного середовища	46
3.1.2	Підготовка серверного середовища.....	47
3.1.3	Створення бази даних та користувача PostgreSQL.....	47
3.1.4	Конфігурація та розгортання серверної частини.....	48
3.1.5	Збірка та розгортання клієнтської частини.....	50
3.2	Інструкція з наповнення інформаційної системи	52
3.2.1	Управління брендами.....	52
3.2.2	Додавання та редагування товарів	54
3.2.3	Обробка замовлень.....	56
3.2.4	Управління туристичними маршрутами	57
3.2.5	Управління промокодами	58
3.3	Інструкція з популяризації та підтримки інформаційної системи.....	59
4	Економічний розділ.....	61
4.1	Визначення стадій технологічного процесу та загальної тривалості проведення розробки інформаційної системи	61
4.2	Визначення витрат на оплату праці та відрахувань на соціальні заходи	62
4.3	Розрахунок витрат на електроенергію	65
4.4	Розрахунок суми амортизаційних відрахувань розробки інформаційної системи.....	65
4.5	Обчислення накладних витрат.....	66
4.6	Складання кошторису витрат та визначення собівартості розробки інформаційної системи.....	66
4.7	Розрахунок ціни розробки інформаційної системи	67
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень	68
5	Охорона праці, техніка безпеки та екологічні вимоги	70

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

5.1	Гарантія прав на охорону праці.....	70
5.2	Засоби і заходи забезпечення пожежної безпеки виробничого об'єкта..	72
	Висновки	75
	Перелік посилань.....	77
	Додатки	79
	Додаток А. Лістинг файлу «RouteGuards.jsx»	79
	Додаток Б. Лістинг файлу «api.js»	82
	Додаток В. Лістинг файлу «AuthContext.jsx»	86
	Додаток Г. Лістинг файлу «CartContext.jsx»	91
	Додаток Д. Лістинг файлу «JwtAuthFilter.java»	97
	Додаток Е. Лістинг файлу «ProductSpec.java»	99
	Додаток Ж. Лістинг файлу «OrderService.java»	100
	Додаток И. Лістинг файлу «GlobalExceptionHandler.java».....	106

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи на тему «Розробка інформаційної системи електронної комерції для реалізації товарів для кемпінгу «PeakPoint»» містить 73 сторінки, 20 таблиць, 16 рисунків та перелік посилань.

Об'єктом дослідження є процеси електронної комерції, управління контентом та онлайн-взаємодії користувачів у сфері роздрібної торгівлі туристичними товарами.

Метою роботи є проектування та розробка високопродуктивної інформаційної системи електронної комерції «PeakPoint».

Пояснювальна записка налічує 5 розділів.

В першому розділі роботи проведено аналітичний огляд чотирьох існуючих платформ у сфері туристичного спорядження, на основі якого виявлено технологічні та функціональні прогалини ринку та сформовано технічне завдання на розробку.

Другий розділ роботи містить опис розробки інформаційної системи: розробки структури вебзастосунку, створення дизайну клієнтської частини, проектування бази даних, написання клієнтської та серверної частин вебзастосунку, проведено мануальне тестування розробленого програмного забезпечення. В третьому розділі розроблено інструкції з розгортання системи на сервері Ubuntu 24.04 LTS, наповнення та адміністрування контенту, а також рекомендації з популяризації та підтримки.

Розрахунок вартості розробки та економічної ефективності приведено в економічній частині в четвертому розділі. Питання охорони праці та техніки безпеки розглянуто в п'ятому розділі.

До складу кваліфікаційної роботи входить графічна частина, яка складається з структурної схеми інформаційної системи, діаграми варіантів використання, діаграми бази даних, а також техніко-економічних показників, що виконані на окремих аркушах формату А1.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

ABSTRACT

The explanatory note for the qualification work on the topic «Development of an E-Commerce Information System for Camping Goods Retail «PeakPoint»» contains 73 pages, 20 tables, 16 figures, and a list of references.

The object of research is the processes of e-commerce, content management, and online user interaction in the field of outdoor and camping goods retail.

The purpose of the work is to design and develop a high-performance e-commerce information system «PeakPoint».

The explanatory note consists of 5 sections.

The first section provides an analytical review of four existing platforms in the field of outdoor equipment, on the basis of which technological and functional gaps in the market were identified and a technical specification for development was formulated.

The second section describes the development of the information system: designing the web application structure, creating the client-side interface, database design, implementation of both client-side and server-side components of the web application, and manual testing of the developed software. The third section presents deployment instructions for the Ubuntu 24.04 LTS server environment, content management and administration guidelines, as well as recommendations for promotion and ongoing support.

The calculation of development costs and economic efficiency is presented in the economic part in the fourth section. Occupational health and safety issues are addressed in the fifth section.

The qualification work includes a graphic part consisting of a structural diagram of the information system, a use case diagram, a database diagram, and technical and economic indicators, all produced on separate A1-format sheets.

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

ВСТУП

Стрімкий розвиток цифрових технологій та глобальна трансформація бізнес-процесів зумовили перехід значної частини роздрібною торгівлі у сферу електронної комерції (e-commerce). Сучасні споживачі вимагають швидкого, безпечного та інтуїтивно зрозумілого доступу до товарів і послуг у будь-який час та з будь-якого пристрою. Особливо відчутною ця тенденція стала в сегменті товарів для активного відпочинку, туризму та кемпінгу. Інтерес до внутрішнього туризму, здорового способу життя та відпочинку на природі стабільно зростає, що створює високий попит на якісне туристичне спорядження.

Традиційні підходи до організації продажів у цій ніші вже не задовольняють потреби ринку в повній мірі. Покупці шукають не просто перелік товарів, а комплексне інформаційне рішення. Наприклад, початківцям у кемпінгу складно самостійно підібрати намет, спальний мішок чи газове обладнання під конкретні погодні умови або складність маршруту. Відтак, інтеграція в систему електронної комерції додаткових інтерактивних модулів, таких як рекомендаційні карти туристичних маршрутів, що підказують необхідний набір спорядження, є надзвичайно актуальною та конкурентною перевагою.

З технічної точки зору, розробка сучасних вебплатформ вимагає використання надійних, масштабованих та швидкодіючих технологій. Застосування архітектурного підходу рознесення клієнтської та серверної частин за допомогою побудови REST API на базі фреймворку Java / Spring Boot та реляційної СКБД PostgreSQL у поєднанні з динамічним інтерфейсом користувача (SPA) на базі бібліотеки React є галузевим стандартом, що забезпечує високу швидкість завантаження сторінок, гнучкість розробки та стійкість до високих навантажень.

Метою кваліфікаційної роботи є проєктування та розробка високопродуктивної інформаційної системи електронної комерції для реалізації товарів для кемпінгу «PeakPoint», яка об'єднує функціонал сучасного інтернет-магазину та рекомендаційного сервісу туристичних маршрутів для покращення

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

користувачького досвіду та оптимізації процесів онлайн-продажів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- 1) Провести аналітичний огляд існуючих вебплатформ у сфері туристичного спорядження для виявлення їхніх переваг та недоліків.
- 2) Сформулювати вимоги технічного завдання до архітектури, функціональності та безпеки інформаційної системи.
- 3) Проектувати структуру бази даних для надійного збереження даних про користувачів, замовлення, маршрути та товари.
- 4) Розробити ергономічний, адаптивний користувачький інтерфейс (UI/UX) із використанням сучасних CSS-фреймворків.
- 5) Реалізувати серверну частину системи у вигляді REST API з чітким розмежуванням прав доступу на основі рольової моделі.
- 6) Реалізувати інтерактивний клієнтський застосунок, інтегрований із серверними ендпоінтами.
- 7) Провести комплексне тестування розробленого програмного забезпечення на предмет функціональності та надійності.
- 8) Описати інструкції з розгортання, наповнення та подальшої підтримки.

Об'єкт дослідження – процеси електронної комерції, управління контентом та онлайн-взаємодії користувачів у сфері роздрібної торгівлі туристичними товарами.

Предмет дослідження – методи, архітектурні патерни, інструменти програмної інженерії та фреймворки (Java/Spring Boot, React, PostgreSQL) для побудови інформаційної системи інтернет-магазину товарів для кемпінгу.

Практичне значення отриманих результатів полягає у створенні готового до розгортання та експлуатації вебзастосунку «PeakPoint». Система дозволяє автоматизувати роботу контент-менеджерів та складського персоналу через адмін-панель, забезпечує клієнтам зручний процес вибору товарів через інтерактивні маршрути, швидке оформлення замовлень та безпечний доступ до особистих даних, що в комплексі знижує операційні витрати бізнесу та підвищує лояльність клієнтів.

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

Для створення конкурентоспроможної інформаційної системи електронної комерції «PeakPoint» необхідно провести детальний аналіз існуючих на ринку України рішень у сегменті роздрібної торгівлі туристичним спорядженням та товарами для кемпінгу. Об'єктами дослідження було обрано провідні галузеві вебплатформи: «Gorganu» [1], «Alp» [2], «Tramp» [3] та «Перевал» [4]. Аналіз проводився за ключовими критеріями: структура каталогу, реалізація системи фільтрації, логіка кошика та оформлення замовлення, наявність особистого кабінету та інтерактивні інструменти підтримки вибору клієнта.

1.1.1 Інформаційна система «Gorganu»

Платформа «Gorganu» [1] є одним із лідерів українського ринку аутдор-спорядження, це велика мультибрендова екосистема електронної комерції, орієнтована на роздрібну торгівлю високоякісним туристичним спорядженням, одягом та взуттям для екстремального спорту. Система має розвинену структуру, що інтегрує вебплатформу з широкою мережею фізичних магазинів у 12 містах України. Перевагами даної платформи є:

- високий рівень структуризації каталогу (поділ на туризм, вело, одяг, взуття тощо);
- інтерактивна карта фізичних точок;
- наявність додаткових сервісів, таких як «Прокат» та інформаційний блог «Журнал PRO» для контент-маркетингу;
- добре оптимізована система фільтрації товарів за технічними характеристиками (вага, матеріали, сезонність) та брендами;
- інтегрована багаторівнева програма лояльності в особистому кабінеті користувача.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

До недоліків можна віднести:

- надмірну кількість категорій та великий обсяг скриптів, інтерфейс платформи може перевантажувати пристрої користувачів із низькою продуктивністю;
- процес оформлення замовлення (Checkout) містить багато обов'язкових кроків, що знижує конверсію;
- головний недолік — відсутність прямого взаємозв'язку між інформаційним контентом (статтями блогу про маршрути) та комерційною частиною: користувач не може автоматично сформувати набір спорядження під обраний у журналі маршрут.

1.1.2 Інформаційна система «Alp»

Мультибрендовий інтернет-магазин спорядження для туризму, альпінізму та кемпінгу «Alp» [2] має ряд переваг:

- потужний функціонал порівняння товарів («Порівняння») та модуль додавання в обране («Бажання»);
- каталог покриває специфічні ніші (дайвінг, промисловий альпінізм, тактичне спорядження);
- функція формування «Списку бажань» (Wishlist);
- наявність гнучкого модуля «Питання/відповідь» безпосередньо у картках товарів та розвинена дисконтна програма.

Проте володіє також наступними недоліками реалізації:

- вебплатформа має класичну монолітну архітектуру, що негативно позначається на швидкості динамічного оновлення кошика без перезавантаження сторінок.
- інтерфейс особистого кабінету є застарілим і надає користувачеві лише базову історію замовлень, без можливості гнучкого керування персональними рекомендаціями.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

1.1.3 Інформаційна система «Tramp»

Фірмовий інтернет-магазин «Tramp» [3], що спеціалізується виключно на продукції бренду Tramp (намети, спальні мішки, посуд, кемпінгові меблі) має ряд переваг та недоліків. До переваг можна віднести:

- вузька спеціалізація дозволяє максимально детально описати картки товарів;
- поділ каталогу здійснено за чіткими категоріями бренду (намети, килимки, спальники, газове обладнання тощо);
- є виділена лінійка товарів із пожиттєвою гарантією (Expedition Line);
- реалізовано зручні модулі замовлення зворотного дзвінка та швидкої онлайн-консультації;
- структура кошика проста, без зайвих відволікаючих елементів.

До недоліків можна зарахувати:

- обмеженість асортименту лише одним брендом;
- фільтрація в каталозі є базовою (переважно за ціною та типом), відсутні складні комбіновані фільтри за специфічними параметрами (наприклад, водостійкість тенту або температурний режим спальника).

1.1.4 Інформаційна система «Перевал»

Регіональна вебплатформа «Перевал» [4] поєднує інтернет-магазин, блог та послуги прокату туристичного інвентарю, має простий, неперевантажений інтерфейс, а також такі переваги:

- окремий акцент на товари для захисників та тактичне спорядження;
- наявність модуля відгуків на головній сторінці, що підвищує рівень довіри користувачів.

Проте система «Перевал» містить ряд недоліків:

- оформлення замовлення часто потребує обов'язкового телефонного підтвердження менеджера через відсутність інтеграції зі складськими системами

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

обліку в реальному часі;

- особистий кабінет клієнта має обмежений функціонал.

Для наочного представлення результатів аналізу сформовано порівняльну таблицю 1.1.

Таблиця 1.1 – Функціональні можливості існуючих інформаційних систем

Функціональний модуль / Критерій	Gorgany	Alp	Tramp	Перевал
Каталог із багаторівневою фільтрацією	Частково (Важкий UI)	Так	Базовий	Базовий
Особистий кабінет покупця	Так	Базовий	Базовий	Ні
Адмін-панель керування (товари/замовлення)	Так	Так	Так	Базова
Динамічний кошик без перезавантаження (SPA)	Ні	Ні	Ні	Ні
Рекомендація спорядження під маршрут	Ні	Ні	Ні	Ні

Проведений аналіз виявив суттєву технологічну та функціональну прогалину на ринку e-commerce товарів для кемпінгу. Усі існуючі системи побудовані за класичним принципом статичних інтернет-магазинів-вітрин, де вибір спорядження повністю покладається на компетенцію покупця.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

Найменування розробки: Інформаційна система електронної комерції для реалізації товарів для кемпінгу «PeakPoint».

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

Областю застосування інформаційної системи «PeakPoint» є сфера електронної комерції, роздрібна торгівля туристичним спорядженням, одягом та аксесуарами для активного відпочинку, а також надання інформаційно-рекомендаційних послуг для планування походів і підбору екіпірування. Платформа орієнтована на кінцевих споживачів (B2C) — туристів, альпіністів, прихильників кемпінгу, а також на внутрішній персонал компанії (менеджери складу, адміністратори контенту) для оптимізації бізнес-процесів.

1.2.2 Призначення розробки

Інформаційна система «PeakPoint» призначена для забезпечення повноцінного циклу онлайн-продажів туристичних товарів через мережу Інтернет (вибір, додавання у кошик, оформлення замовлення). Також для надання користувачам зручного персоналізованого інструментарію через особистий кабінет (перегляд історії замовлень, збереження профілю), оптимізації роботи адміністративного персоналу щодо управління каталогом товарів, категоріями та замовленнями клієнтів.

1.2.3 Вимоги до функціоналу інформаційної системи

1.2.3.1 Вимоги до функціональних характеристик

Система повинна забезпечувати розмежування прав доступу на основі рольової моделі для таких категорій користувачів: Гість, Авторизований покупець, Адміністратор.

Користувачі з роллю «Гість» повинні мати можливість виконання наступних дій:

- перегляд головної сторінки, каталогу товарів з багаторівневою фільтрацією та сортуванням;
- перегляд детальної картки товару, технічних характеристик, описів та відгуків;

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

- пошук товарів за ключовими словами;
- додавання товарів до кошика (зберігання стану на стороні клієнта);
- реєстрація та автентифікація в системі.

Користувачі з роллю «Авторизований покупець» повинні мати усі можливості ролі «Гість», а також можливість виконання наступних дій:

- доступ до особистого кабінету з можливістю редагування персональних даних;
- перегляд історії власних замовлень та їх поточних статусів;
- оформлення замовлення (Checkout) із зазначенням даних доставки та оплати;
- додавання відгуків та оцінок до придбаних товарів.

Для керування інформаційною системою «PeakPoint» потрібно розробити роль «Адміністратор», які дає забезпечити можливість виконання таких операцій:

- доступ до спеціалізованого інтерфейсу адмін-панелі;
- перегляд списку всіх замовлень у системі;
- зміна статусів замовлень (наприклад: «Очікує відправки», «Відправлено», «Виконано»);
- моніторинг та редагування залишків товарів на складі.
- керування користувачами (блокування, зміна ролей);
- управління каталогом: додавання, редагування, видалення товарів, брендів та категорій;
- управління контентом модуля туристичних маршрутів: створення нових маршрутів та прив'язка до них списку рекомендованого спорядження.

1.2.3.2 Вимоги до надійності та безпеки

Система повинна коректно реагувати на помилкові дії користувачів, обробляти помилки з'єднання з базою даних, відсутність ендпоінтів (404) чи внутрішні помилки сервера (500) із видачею інформативних повідомлень на інтерфейс.

Захист конфіденційної інформації користувачів, авторизація та

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

автентифікація повинні здійснюватися на основі сучасного галузевого стандарту Spring Security з використанням токенів безпеки JWT (JSON Web Tokens). Паролі користувачів у БД PostgreSQL повинні зберігатися виключно в хешованому вигляді за допомогою стійкого алгоритму (наприклад, BCrypt).

Усі вхідні дані повинні проходити дворівневу валідацію: на стороні клієнта (React-форми) для покращення UX та обов'язкову строгу валідацію на стороні сервера (Spring Boot Validation / JSR 380) для запобігання SQL-ін'єкціям та XSS-атакам.

1.2.3.3 Умови експлуатації та вимоги до інтерфейсу

Взаємодія з системою має здійснюватися через веббраузер у режимі реального часу. Інтерфейс користувача повинен бути реалізований як Single Page Application (SPA) на базі React.

Верстка вебсторінок має бути повністю адаптивною (компоненти Tailwind CSS або Material UI) для коректного відображення на мобільних пристроях, планшетах та десктопах.

Забезпечення стабільної роботи платформи у всіх сучасних браузерах (Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge) є обов'язковою вимогою до розробки інформаційної системи.

1.2.3.4 Вимоги до складу та параметрів технічних засобів

Серверна частина (Backend + DB):

- процесор: мінімум 2-ядерний з частотою від 2.0 ГГц;
- оперативна пам'ять (RAM): не менше 4 ГБ (для паралельної роботи Spring Boot додатка та СКБД PostgreSQL);
- дисковий простір: від 10 ГБ вільного місця на SSD.

Клієнтська частина (для користувача) представляє собою будь-який пристрій (ПК, смартфон) з доступом до мережі Інтернет та встановленим браузером.

1.2.3.5 Вимоги до інформаційної та програмної сумісності

На стороні сервера повинна використовуватись операційна система сімейства Linux (Ubuntu/Debian); встановлене середовище виконання Java (у

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

межах JDK 17 або вище), СКБД PostgreSQL (версії 15 або вище).

На стороні клієнта повинне використовуватись програмне забезпечення (браузер) з підтримкою виконання JavaScript.

1.2.4 Вимоги до програмної документації

Розробка інформаційної системи повинна супроводжуватися створенням таких документів:

- пояснювальна записка кваліфікаційної роботи (опис архітектури, структури БД, опис алгоритмів та результатів тестування);
- інструкція з розгортання та інсталяції інформаційної системи (для розробників/системних адміністраторів);
- інструкція з наповнення інформаційної системи;
- інструкція з популяризації та підтримки (SEO, технічне обслуговування коду та БД). [5]

1.2.5 Техніко-економічні показники

Економічний ефект від впровадження інформаційної системи «PeakPoint» досягається завдяки:

- зниженню витрат на утримання фізичних торгових точок шляхом повної міграції бізнес-процесів в онлайн-сегмент;
- підвищенню конверсії сайту та середнього чека покупця;
- мінімізації людського фактора та помилок в обліку залишків товарів внаслідок автоматизації складських операцій в адмін-панелі.

Розробка виконується на базі програмного забезпечення з відкритим вихідним кодом (Open Source: Java, Spring Boot, PostgreSQL, React, Tailwind CSS), що повністю виключає витрати на придбання коштовних комерційних ліцензій.

Проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки буде представлено в четвертому розділі.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

1.2.6 Стадії та етапи розробки інформаційної системи

Процес проєктування та реалізації системи розбито на такі етапи відповідно до затвердженого плану роботи:

- 1) Передпроектне дослідження та аналіз аналогів та конкурентів на ринку, формування мети та завдань проєкту, затвердження технічного завдання.
- 2) Технічне проєктування, що поєднує в собі розробку загальної структури системи, проєктування логіки взаємодії клієнта та сервера через REST API, створення та верстку сторінок інформаційної системи (UI/UX макети).
- 3) Проєктування бази даних, де буде виконано розробку структури бази даних PostgreSQL, визначення сутностей, зв'язків, побудова ER-діаграми.
- 4) Програмування системи, яке складається з написання клієнтської частини (React, react-router-dom, інтеграція з API) та написання серверної частини (Java, Spring Boot, Spring Security, Spring Data JPA).
- 5) Тестування, впровадження та документування – проведення модульного та інтеграційного тестування, написання інструкції з розгортання та експлуатації.

1.2.7 Порядок тестування та прийому

Контроль за якістю та відповідністю розробки вимогам ТЗ здійснюється на етапі тестування інформаційної системи. Прийом готового програмного продукту включає перевірку:

- коректності виконання всіх функціональних прецедентів для кожної з чотирьох рольових моделей (Гість, Покупець, Менеджер, Адмін);
- працездатності REST API за допомогою автоматизованих сценаріїв (Postman / Unit-тести Spring);
- відсутності критичних дефектів верстки на різних типах екранів;
- наявності повного пакета документації та інструкцій.

Готовий програмний комплекс демонструється керівнику кваліфікаційної роботи для підтвердження виконання технічного завдання в повному обсязі. [5]

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури інформаційної системи

Розробка інформаційної системи «PeakPoint» розпочалась із визначення її функціонального призначення та формування загальної архітектури вебресурсу. Відповідно до технічного завдання, система є повнофункціональним інтернет-магазином туристичного спорядження, що поєднує комерційний каталог товарів з унікальним модулем рекомендацій спорядження на основі туристичних маршрутів. Це визначило дворівневу природу контенту платформи: комерційний (товари, замовлення, кошик) та інформаційно-рекомендаційний (маршрути і прив'язане до них спорядження).

На етапі проєктування було визначено повний перелік сторінок застосунку та логіку навігації між ними. На головній сторінці розміщено: банер-герой із закликом до дії (перейти в каталог), блок товарів зі знижками, секцію «Популярні категорії» та блок маршрутів як точок входу до рекомендаційного модуля. Сторінка каталогу спроектована з бічною панеллю фільтрів та основною сіткою карток товарів, що забезпечує зручне звуження асортименту за категорією, брендом та ціновим діапазоном. Картка товару містить галерею зображень, технічні характеристики, блок відгуків і кнопку додавання до кошика та в «улюблене». Сторінкову організацію клієнтської частини та взаємозв'язки між основними модулями інтерфейсу відображено на схемі структурній, наведеній на плакаті 2026.КВР. 122.421.07.00.00 СС.

Відповідно до рольової моделі, описаної у технічному завданні, частина сторінок є публічно доступною, а частина — захищеною та вимагає авторизації з певними правами доступу. Для графічного відображення взаємодії між акторами системи (Гість, Покупець, Адміністратор) та доступними їм прецедентами розроблено UML-діаграму варіантів використання, що наведена на плакаті 2026.КВР. 122.421.07.00.00 ДВ. ВОпис сторінок та маршрутів інформаційної системи «PeakPoint» наведено у таблиці 2.1.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

Таблиця 2.1 – Структура сторінок та маршрутів інформаційної системи «PeakPoint»

Сторінка/	URL	Доступ	Компонент	Опис
Головна	/	Всі	HomePage	Банер, акції, маршрути
Каталог	/catalog	Всі	CatalogPage	Фільтрація, пошук, картки
Картка товару	/product/:id	Всі	ProductPage	Деталі, відгуки, кошик
Кошик	/cart	Всі	CartPage	Перегляд, редагування
Оформлення замовлення	/checkout	Покупець.	CheckoutPage	Дані доставки й оплати
Підтвердження	/confirm	Покупець.	ConfirmPage	Квитанція замовлення
Кабінет користувача	/profile	Покупець	ProfilePage	Профіль, замовлення
Маршрути	/routes	Всі	RoutesPage	Туристичні маршрути
Адмін: товари	/admin/products	Адмін	ProductsAdminPage	CRUD товарів
Адмін: маршрути	/admin/routes	Адмін	RoutesConfigPage	Керування маршрутами
Аналітика	/admin/analytics	Адмін	AnalyticsPage	Статистика системи

2.2 Створення та верстка сторінок інформаційної системи

На рівні проектування інтерфейсу була сформована концепція дизайну, що базується на принципах сучасного мінімалізму з акцентом на природну тематику. Основна палітра кольорів поєднує відтінки зеленого та бежевого, які асоціюються з природою та кемпінгом. Для досягнення консистентності візуального стилю на всіх сторінках сформовано єдину дизайн-систему на основі Tailwind CSS: визначено набір типографічних стилів, кольорових констант, відступів та розмірів елементів.

Верстка клієнтської частини інформаційної системи виконана з використанням бібліотеки React 18 та утиліт Tailwind CSS версії 3.4. Збірка проекту здійснюється інструментом Vite, який забезпечує миттєве оновлення під час розробки (Hot Module Replacement) та оптимізовану продуктивну збірку з tree-shaking і code-splitting.

Архітектура компонентів відповідає принципу атомарного дизайну. На найнижчому рівні розташовані атомарні UI-компоненти (директорія `src/components/ui/`): `Pagination`, `ProductCard`, `StarRating`, `Toast` — перевикористовувані елементи без прив'язки до бізнес-логіки. На наступному рівні — складові компоненти (`src/components/product/`, `src/components/cart/`): `ProductCard`, `ProductGrid`, `CartLine`, `CartSummary`. Найвищий рівень — компоненти-сторінки (`src/pages/`), що компонують складові елементи та підключаються до відповідних сервісів через хуки React.

Для стилізації відмовились від будь-яких сторонніх UI-бібліотек на користь утиліт Tailwind CSS. Такий підхід виключає конфлікти між стилями сторонніх компонентів та власною дизайн-системою, а також значно зменшує розмір підсумкового CSS-бандлу завдяки автоматичному видаленню невикористаних класів при збірці. Іконки реалізовано через бібліотеку Lucide React, що надає уніфікований набір SVG-іконок.

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

Верстка усіх сторінок є повністю адаптивною (responsive). Точки зламу (breakpoints) Tailwind CSS (sm: 640px, md: 768px, lg: 1024px, xl: 1280px) застосовуються до розміщення елементів сітки каталогу, навігації Header та структури сторінки оформлення замовлення. На мобільних пристроях бічна панель фільтрів каталогу трансформується у модальне вікно, яке активується кнопкою, що зберігає зручність взаємодії на малих екранах.

2.3 Розробка структури бази даних інформаційної системи

Реляційна база даних інформаційної системи «PeakPoint» спроектована засобами СКБД PostgreSQL версії 15. Вибір реляційної моделі зумовлений природою предметної області: між сутностями системи існують чіткі, задокументовані зв'язки, цілісність яких критично важлива для коректного функціонування комерційних процесів — оформлення замовлень, обліку залишків та нарахування рейтингів товарів. Реляційна СКБД гарантує дотримання цих обмежень на рівні ядра бази даних через механізми зовнішніх ключів і транзакцій стандарту ACID.

Схема бази даних налаштована у режимі ddl-auto=update, що дозволяє Hibernate автоматично приводити DDL-структуру до поточного стану JPA-сутностей при кожному старті застосунку. Така конфігурація прийнятна на стадії активної розробки і тестування. При першому запуску компонент DataInitializer (реалізація CommandLineRunner) перевіряє наявність даних у таблицях і, якщо база порожня, заповнює її початковим набором: 8 категорій, 6 брендів, 12 тестових товарів, 4 типи маршрутів та промокод PEAK10 зі знижкою 10%.

База даних системи «PeakPoint» складається з 15 таблиць, що охоплюють чотири функціональні домени: управління користувачами і безпекою, товарний каталог, обробку замовлень та інформаційно-рекомендаційний модуль маршрутів. Загальний перелік таблиць з їхнім призначенням наведено у таблиці 2.2.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

Таблиця 2.2 – Перелік таблиць бази даних системи «PeakPoint»

Таблиця БД	Призначення
1	2
users	Облікові записи користувачів: ім'я, email, хеш пароля (BCrypt), роль (CUSTOMER / ADMIN), прапор підтвердження email
refresh_tokens	Токени оновлення сесії для реалізації refresh token rotation: значення токена, зовнішній ключ до users, термін дії
email_verification_tokens	Одноразові токени підтвердження email при реєстрації: значення токена, зовнішній ключ до users, термін дії (24 год)
categories	Ієрархічний довідник категорій каталогу: назва, slug; підтримує самоприєднання (self-join) для дерева підкатегорій
brands	Довідник брендів: назва, автоматично генерований slug (lower-kebab-case), URL логотипу, опис
products	Центральна таблиця товарного каталогу: назва, опис, ціна (ціле число, гривні), залишок на складі, зв'язки з categories та brands, набір сезонів (enum), бейдж (SALE / NEW / тощо), рейтинг, лічильник відгуків
product_images	Зображення товарів: URL файлу на сервері, порядок відображення (sortOrder), зовнішній ключ до products
product_specs	Технічні характеристики товарів у форматі «ключ–значення»: назва характеристики, значення, зовнішній ключ до products
orders	Замовлення: зовнішній ключ до users (може бути NULL для гостьового checkout), загальна сума, статус (PENDING / SHIPPED / DELIVERED / CANCELLED), спосіб доставки, спосіб оплати, адреса, номер відстеження

Продовження таблиця 2.2

1	2
wishlist_items	Список бажань покупця: зовнішні ключі до users та products, дата додавання
promo_codes	Промокоди: код, відсоток знижки, прапор активності, дата закінчення терміну дії
reviews	Відгуки покупців: зовнішні ключі до users та products, оцінка (1–5), текст, дата; унікальна пара (user_id, product_id); зміна відгуку автоматично перераховує Product.rating та Product.reviewCount
route_types	Типи туристичних маршрутів («Піший», «Гірський», «Водний» тощо): назва, опис, URL зображення, тривалість, список рекомендованих товарів
contact_messages	Звернення з форми зворотнього зв'язку: ім'я, email, текст повідомлення, статус обробки, можливість відповіді адміністратора через email

Домен управління користувачами і безпекою охоплює три таблиці:

- таблиця users є центральною обліковою сутністю;
- таблиця refresh_tokens реалізує механізм rotation токенів оновлення: при кожному успішному refresh-запиті старий токен відкликається і замінюється новим, що унеможлиблює повторне використання перехопленого токена;
- таблиця email_verification_tokens зберігає одноразові посилання для підтвердження email при реєстрації з терміном дії 24 години.

Детальну структуру таблиці users наведено у таблиці 2.3.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

Таблиця 2.3 – Структура таблиці users

№	Поле	Тип даних	Опис
1	id	BIGSERIAL (PK)	Унікальний ідентифікатор
2	name	VARCHAR(100)	Ім'я користувача
3	email	VARCHAR(255), UNIQUE	Електронна пошта — унікальний ідентифікатор входу
4	password	VARCHAR(255)	Хеш пароля (BCrypt, cost factor 10)
5	role	VARCHAR(30)	Роль: CUSTOMER або ADMIN
6	email_verified	BOOLEAN	Прапор підтвердження email; false — вхід заблоковано (за умови увімкненого SMTP)
7	created_at	TIMESTAMP	Дата та час реєстрації

Домен товарного каталогу є найбільш розгалуженим і включає п'ять таблиць: categories, brands, products, product_images та product_specs.

Таблиця categories підтримує ієрархічну структуру через self-join: поле parent_id посилається на id у тій же таблиці, що дозволяє формувати дерево категорій довільної глибини.

Таблиця brands забезпечена захистом від видалення бренду, що має прив'язані товари: сервісний шар перевіряє наявність записів у products перед виконанням DELETE.

Таблиця product_specs реалізує паттерн «Entity-Attribute-Value» у спрощеному вигляді: кожна характеристика зберігається як окремий рядок з полями key та value, що надає гнучкість для опису різномірних товарів без зміни схеми БД.

Детальну структуру центральної таблиці products наведено у таблиці 2.4.

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Таблиця 2.4 – Структура таблиці products

№	Поле	Тип даних	Опис
1	id	BIGSERIAL (PK)	Унікальний ідентифікатор товару
2	name	VARCHAR(255)	Повна назва товару
3	short_description	VARCHAR(500)	Короткий опис для картки у каталозі та повнотекстового пошуку
4	description	TEXT	Детальний опис для сторінки товару
5	price	INTEGER	Роздрібна ціна у гривнях (ціле число)
6	old_price	INTEGER	Стара ціна для відображення закресленої ціни при акції; NULL — акції немає
7	stock_quantity	INTEGER	Залишок одиниць товару на складі
8	category_id	BIGINT (FK)	Посилання на таблицю categories
9	brand_id	BIGINT (FK)	Посилання на таблицю brands
10	seasons	VARCHAR[] / JSON	Набір сезонів придатності товару: WINTER, SPRING, SUMMER, AUTUMN
11	badge	VARCHAR(30)	Бейдж-мітка товару: SALE, NEW, BESTSELLER тощо; NULL — без мітки
12	rating	DOUBLE PRECISION	Середній рейтинг на основі відгуків; автоматично перераховується
13	review_count	INTEGER	Кількість відгуків; автоматично оновлюється синхронно з полем rating
15	created_at	TIMESTAMP	Дата та час додавання запису

Домен обробки замовлень включає таблиці orders, order_items, wishlist_items та promo_codes. Ключовою архітектурною особливістю є підтримка гостьового оформлення замовлення: поле user_id у таблиці orders допускає значення NULL, що дозволяє покупцям завершувати покупку без реєстрації в системі. При цьому контактні дані отримувача (first_name, last_name, email, phone) зберігаються безпосередньо у записі замовлення. Детальну структуру таблиці orders наведено у таблиці 2.5.

Таблиця 2.5 – Структура таблиці orders

№	Поле	Тип даних	Опис
1	2	3	4
1	id	BIGSERIAL (PK)	Унікальний ідентифікатор замовлення
2	user_id	BIGINT (FK, NULL)	Покупець; NULL допускається для гостьового оформлення замовлення
3	total_amount	INTEGER	Загальна сума замовлення у гривнях (розраховується виключно на сервері)
4	status	VARCHAR(30)	Статус: PENDING → SHIPPED → DELIVERED або CANCELLED
5	delivery_method	VARCHAR(50)	Спосіб доставки (enum DeliveryMethod): кур'єр, Нова Пошта тощо
6	payment_method	VARCHAR(50)	Спосіб оплати (enum PaymentMethod): накладений платіж, онлайн тощо
7	delivery_address	TEXT	Адреса доставки, зазначена покупцем при оформленні
8	tracking_no	VARCHAR(100)	Номер відстеження відправлення (заповнюється при відправці)

Продовження таблиці 2.5

1	2	3	4
8	tracking_no	VARCHAR(100)	Номер відстеження відправлення (заповнюється менеджером при відправці)
9	promo_code	VARCHAR(50)	Застосований промокод; зберігається як текст для аудиту навіть після деактивації коду
10	first_name	VARCHAR(100)	Ім'я отримувача (гостьового замовлення)
11	last_name	VARCHAR(100)	Прізвище отримувача
12	email	VARCHAR(255)	Email для надсилання підтвердження замовлення
13	phone	VARCHAR(20)	Номер телефону отримувача
14	created_at	TIMESTAMP	Дата та час оформлення замовлення

Ціна кожного товару замовлення фіксується у таблиці order_items на момент оформлення і ніколи не переобчислюється на основі поточної ціни, що гарантує незмінність фінансових зобов'язань за раніше оформленими замовленнями. Детальну структуру таблиці order_items наведено у таблиці 2.6.

Таблиця 2.6 – Структура таблиці order_items

№	Поле	Тип даних	Опис
1	id	BIGSERIAL (PK)	Унікальний ідентифікатор рядка
2	order_id	BIGINT (FK)	Посилання на замовлення
3	product_id	BIGINT (FK)	Посилання на товар
4	quantity	INTEGER	Кількість одиниць товару у замовленні
5	price	INTEGER	Ціна одиниці товару на момент оформлення

Таблиця reviews реалізує систему рейтингів із вбудованим механізмом автоматичного перерахунку. Унікальне обмеження на пару (user_id, product_id) гарантує, що кожен покупець може залишити лише один відгук на конкретний товар. Після кожного збереження або оновлення відгуку сервісний шар ReviewService автоматично перераховує поля rating та review_count у таблиці products, що забезпечує актуальність агрегованих показників без необхідності виконання окремих агрегатних запитів при кожному відображенні картки товару. Детальну структуру таблиці reviews наведено у таблиці 2.7.

Таблиця 2.7 – Структура таблиці reviews

№	Поле	Тип даних	Опис
1	id	BIGSERIAL (PK)	Унікальний ідентифікатор відгуку
2	user_id	BIGINT (FK)	Автор відгуку
3	product_id	BIGINT (FK)	Товар, якому залишено відгук
4	rating	INTEGER	Оцінка від 1 до 5; унікальна комбінація (user_id, product_id) — один відгук на користувача
5	comment	TEXT	Текст відгуку
6	created_at	TIMESTAMP	Дата публікації відгуку

Домен рекомендаційного модуля представлений таблицею route_types. Її особливістю є зв'язок «багато до багатьох» (M:N) з таблицею products через JPA-анотацію @ManyToMany, що Hibernate матеріалізує у вигляді зв'язкової таблиці route_type_products. Цей зв'язок є технічною основою ключової конкурентної переваги платформи: адміністратор формує перелік рекомендованого спорядження для кожного типу маршруту через адмін-панель, а ендпоінт GET /api/v1/products/route-recommend повертає відфільтрований список товарів на

основі обраного routeType та тривалості походу duration, що надає покупцям персоналізовану підбірку екіпірування. Детальну структуру таблиці route_types наведено у таблиці 2.8.

Таблиця 2.8 – Структура таблиці route_types

№	Поле	Тип даних	Опис
1	id	BIGSERIAL (PK)	Унікальний ідентифікатор типу маршруту
2	name	VARCHAR(100)	Назва типу маршруту (наприклад, «Піший», «Гірський»)
3	description	TEXT	Опис маршруту та необхідного рівня підготовки
4	image_url	VARCHAR(500)	URL зображення для картки маршруту
5	duration_days	INTEGER	Орієнтовна тривалість походу у днях
6	recommended_products	@ManyToMany → products	Список рекомендованих товарів для даного типу маршруту (зв'язкова таблиця M:N)

Таблиця contact_messages забезпечує роботу форми зворотнього зв'язку і не пов'язана зовнішніми ключами з іншими таблицями, оскільки надсилати звернення можуть як зареєстровані, так і незареєстровані відвідувачі. Адміністратор отримує доступ до повного списку звернень через захищений ендпоінт, може змінювати статус обробки та надсилати відповідь безпосередньо на email відправника через сервіс EmailService.

Розроблену ER-діаграму бази даних інформаційної системи «PeakPoint»

представлено на плакаті 2026.KBP.122.421.07.00.00 БД.

Доступ серверної частини до бази даних здійснюється виключно через Spring Data JPA з Hibernate як ORM-провайдером. Кожна таблиця відповідає анотованому Java-класу-сутності у пакеті `ua.reactpoint.entity`. Для складних запитів — зокрема фільтрації каталогу за комбінацією категорії, брендів, сезонів, цінового діапазону та ключових слів — використовується Specification API з `JpaSpecificationExecutor`, що дозволяє динамічно компонувати предикати JPA без написання вручну JPQL-рядків. Повнотекстовий пошук реалізований нативним PostgreSQL-запитом `websearch_to_tsquery("simple", ...)` по полях `name`, `brand` та `short_description`, який інтегрується у загальний ланцюжок Specification і поєднується з іншими фільтрами.

2.4 Програмування інформаційної системи

2.4.1 Написання клієнтської частини

Клієнтська частина інформаційної системи реалізована як SPA-застосунок на базі React 18 з використанням виключно функціональних компонентів та хуків [6]. Вибір функціонального підходу обумовлений його сучасністю, кращою підтримкою з боку екосистеми React та більш лаконічним кодом порівняно з класовими компонентами.

Маршрутизація клієнтської частини побудована на React Router DOM v6 [7]. Кореневий компонент `App.jsx` визначає дерево маршрутів за допомогою компонентів `BrowserRouter`, `Routes` та `Route`. Захищені маршрути обгорнуто у компонент-обгортку `PrivateRoute`, який перевіряє наявність автентифікованої сесії та відповідність ролі користувача вимогам маршруту. За відсутності прав доступу компонент виконує перенаправлення на сторінку входу засобами хука `useNavigate`. Лістинг компонента файлу `RouteGuards.jsx` наведено у додатку А.

Комунікація з REST API серверної частини організована через сервісний шар (`src/services/`). Кожен сервіс відповідає певному домену: `authService.js` —

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

реєстрація та вхід, `productService.js` — каталог і пошук товарів, `orderService.js` — оформлення та відстеження замовлень. Всі сервіси використовують єдиний налаштований екземпляр `Axios (api.js)`, що автоматично прикріплює JWT-токен до кожного запиту [8]. `Axios`-інтерсептор відповіді перехоплює HTTP 401 та автоматично виконує `refresh`-запит; при невдачі — генерує подію `auth:unauthorized`, що ініціює `logout` та перенаправлення на сторінку входу. Лістинг файлу `api.js` наведено у додатку Б.

Форми реєстрації, входу та оформлення замовлення реалізовані через контрольовані компоненти `React` з локальним станом `useState`. Валідація вхідних даних на стороні клієнта виконується утилітами з модуля `src/utills/validators.js`: перевіряється формат `email`, мінімальна довжина пароля, наявність обов'язкових полів форми доставки. Усі асинхронні операції супроводжуються індикаторами завантаження, а результати — `toast`-сповіщеннями через глобальний хук `useToast`.

Управління станом автентифікації реалізовано через `AuthContext`, який зберігає дані поточного користувача, роль та методи `login()` і `logout()`. Лістинг компонента `AuthContext` наведено у додатку В. `CartContext` управляє станом кошика з операціями додавання, видалення та зміни кількості товарів; стан кошика персистується у `localStorage` браузера між сесіями. Лістинг компонента `CartContext` наведено у додатку Г.

HTTP-комунікація між клієнтом та сервером організована через єдиний сервісний шар (`src/services/`). Бібліотека `Axios` налаштована як централізований екземпляр (`api.js`) з інтерсепторами запитів і відповідей. Інтерсептор запитів автоматично додає JWT-токен з `localStorage` до заголовку `Authorization` кожного захищеного запиту. Інтерсептор відповідей перехоплює помилки з HTTP-кодом 401 та ініціює оновлення токена або перенаправлення на сторінку входу, що забезпечує прозорість роботи сесії для користувача.

2.4.2 Написання серверної частини

Серверна частина системи «PeakPoint» реалізована мовою `Java` 17 з

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

використанням фреймворку Spring Boot 3.3.4 [6]. Архітектурний підхід — REST API зі stateless-сесіями та чітким розподілом відповідальності за трьома шарами: Controller приймає HTTP-запити та делегує обробку; Service містить бізнес-логіку; Repository реалізує доступ до бази даних через Spring Data JPA [9].

REST API системи складається з дев'яти контролерів пакету ua.peakpoint.controller. Усі маршрути починаються з префіксу /api/v1. API задокументований за стандартом OpenAPI 3.0 за допомогою бібліотеки springdoc-openapi та доступний через Swagger UI за адресою /swagger-ui.html із підтримкою JWT Bearer-авторизації [10]. За рівнем доступу всі ендпоінти поділяються на три групи: публічні (без токена), захищені рівня CUSTOMER+ (вимагають дійсного JWT) та адміністративні рівнів MANAGER+ і ADMIN (перевіряються анотаціями @PreAuthorize).

AuthController реалізує реєстрацію та вхід з видачею пари токенів (access — 15 хв, refresh — 7 діб), механізм refresh token rotation (при кожному оновленні старий токен відкликається), підтвердження email через одноразові токени, отримання та оновлення профілю поточного користувача і logout з інвалідацією refresh-токена на сервері [10]. Автентифікація на основі JWT реалізована через фільтр JwtAuthenticationFilter, лістинг якого наведено у додатку Д.

ProductController обслуговує як публічний каталог, так і адміністративний CRUD. Публічна частина надає пагінований список товарів з динамічною фільтрацією через Specification API (за категорією, брендами, сезонами, ціновим діапазоном), повнотекстовий пошук засобами PostgreSQL-функції websearch_to_tsquery, детальну картку товару з характеристиками та зображеннями, рекомендації за категорією і рекомендації спорядження за типом маршруту (ендпоінт /route-recommend). Реалізацію класу ProductSpecification, що забезпечує динамічну фільтрацію, наведено у додатку Е. Адміністративна частина дозволяє створювати та редагувати товари, завантажувати зображення через FileStorageService і видаляти записи з каскадним видаленням файлів.

CategoryController та BrandController надають публічний доступ до довідників категорій і брендів, а також захищені CRUD-операції для менеджерів.

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

BrandService автоматично генерує slug з назви бренду у форматі lower-kebab-case; видалення бренду блокується, якщо до нього прив'язані товари.

OrderController реалізує оформлення замовлення як для авторизованих покупців, так і в гостьовому режимі (без реєстрації). Ключовою особливістю є серверне обчислення суми: OrderService завжди зчитує актуальні ціни товарів з бази даних, ігноруючи будь-які ціни від клієнта. При застосуванні промокоду сервіс валідує його і розраховує знижку. Після оформлення та при кожній зміні статусу замовлення EmailService надсилає покупцю повідомлення на основі Thymeleaf-шаблону. Лістинг класу OrderService наведено у додатку Ж.

WishlistController надає авторизованим покупцям можливість додавати товари до списку бажань та видаляти їх. Система відгуків реалізована в рамках ProductController: унікальне обмеження на пару (userId, productId) гарантує один відгук на покупця, а після кожного збереження ReviewService автоматично перераховує поля rating та reviewCount у таблиці products.

PromoController розділений на два рівні: публічний ендпоінт /promo/validate дозволяє покупцю перевірити промокод під час оформлення замовлення, тоді як повний CRUD промокодів доступний лише адміністратору. ContactController зберігає звернення з форми зворотнього зв'язку, надає адміністратору список звернень з можливістю зміни статусу та надсилання відповіді на email через EmailService. RouteTypeController повертає публічний список типів маршрутів з прив'язаними товарами для рекомендаційного блоку головної сторінки.

Обробка помилок у всіх контролерах централізована через компонент GlobalExceptionHandler (анотація @RestControllerAdvice). Він перехоплює виключення ResourceNotFoundException (HTTP 404), BadRequestException (HTTP 400) та помилки валідації Spring Validation (HTTP 400 з деталями по кожному полю), формуючи уніфіковану структуру JSON-відповіді з полями message, status та timestamp. Це забезпечує передбачувану поведінку API та спрощує обробку помилок на стороні клієнта. Лістинг класу GlobalExceptionHandler наведено у додатку И.

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

2.5 Тестування інформаційної системи

Тестування інформаційної системи «PeakPoint» проводилось методом ручного (мануального) тестування клієнтської частини. Метою тестування було підтвердити відповідність реалізованого функціоналу вимогам технічного завдання, виявити відхилення у поведінці інтерфейсу та переконатися у коректному відображенні системи у цільових браузерних середовищах і на різних типах пристроїв. Тестування охоплювало п'ять функціональних блоків: автентифікацію та управління профілем, публічний каталог і пошук, кошик і оформлення замовлення, адміністративну панель, а також список бажань, відгуки та крос-браузерну адаптивність.

2.5.1 Тестування автентифікації та управління профілем

Перший блок тестування охоплює реєстрацію нового облікового запису, вхід до системи, перегляд і редагування профілю, а також перевірку захисту маршрутів від неавторизованого доступу. Сторінки реєстрації та входу реалізовані у вигляді модальних вікон, що забезпечує зручність взаємодії без переходу на окремі сторінки. Зовнішній вигляд форми реєстрації наведено на рисунку 2.1.

Рисунок 2.1 — Форма реєстрації нового облікового запису

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

Зовнішній вигляд форми входу наведено на рисунку 2.2, особистого кабінету покупця — на рисунку 2.3

Рисунок 2.2 — Форма входу до системи

Рисунок 2.3 — Особистий кабінет авторизованого покупця

2.5.2 Тестування публічного каталогу та пошуку

Другий блок охоплює перевірку головної сторінки, сторінки каталогу з фільтрацією та сортуванням, детальної картки товару, повнотекстового пошуку та рекомендаційного модуля маршрутів. Особлива увага приділялась коректності комбінованої фільтрації — одночасному застосуванню фільтрів за категорією,

брендом, сезоном та ціновим діапазоном. Зовнішній вигляд головної сторінки наведено на рисунку 2.4, сторінки каталогу з фільтрами – на рисунку 2.5

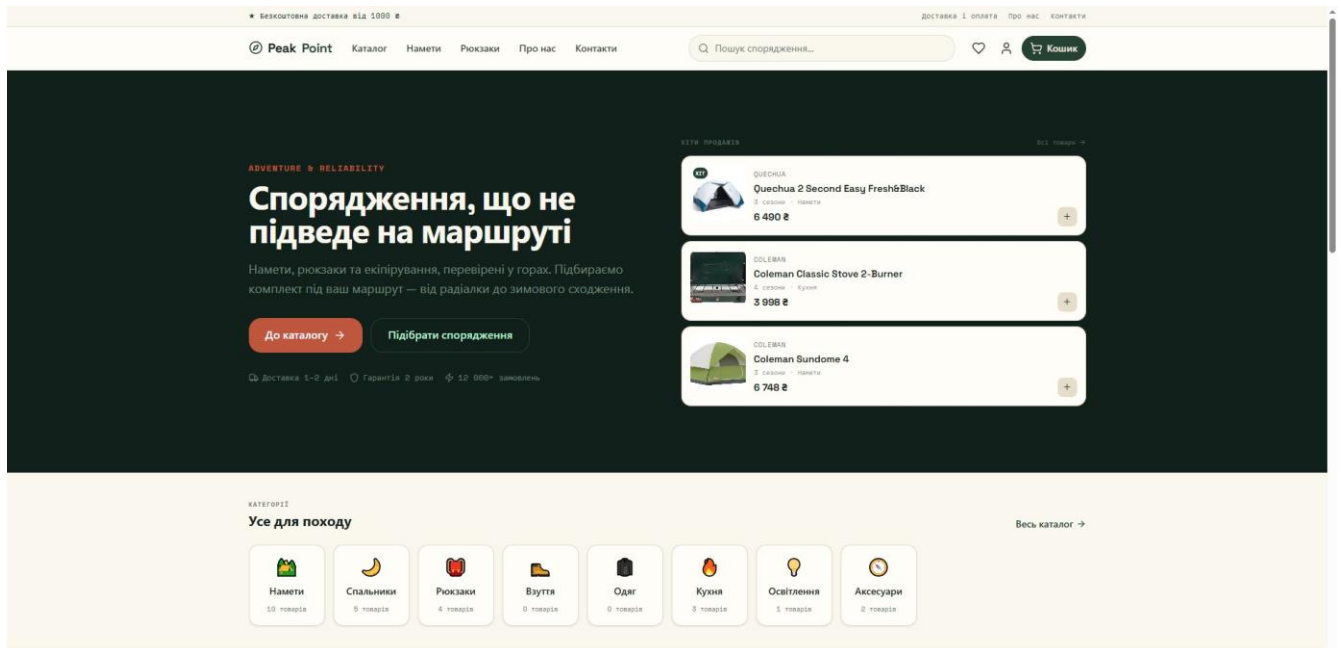


Рисунок 2.4 — Головна сторінка інформаційної системи «PeakPoint»

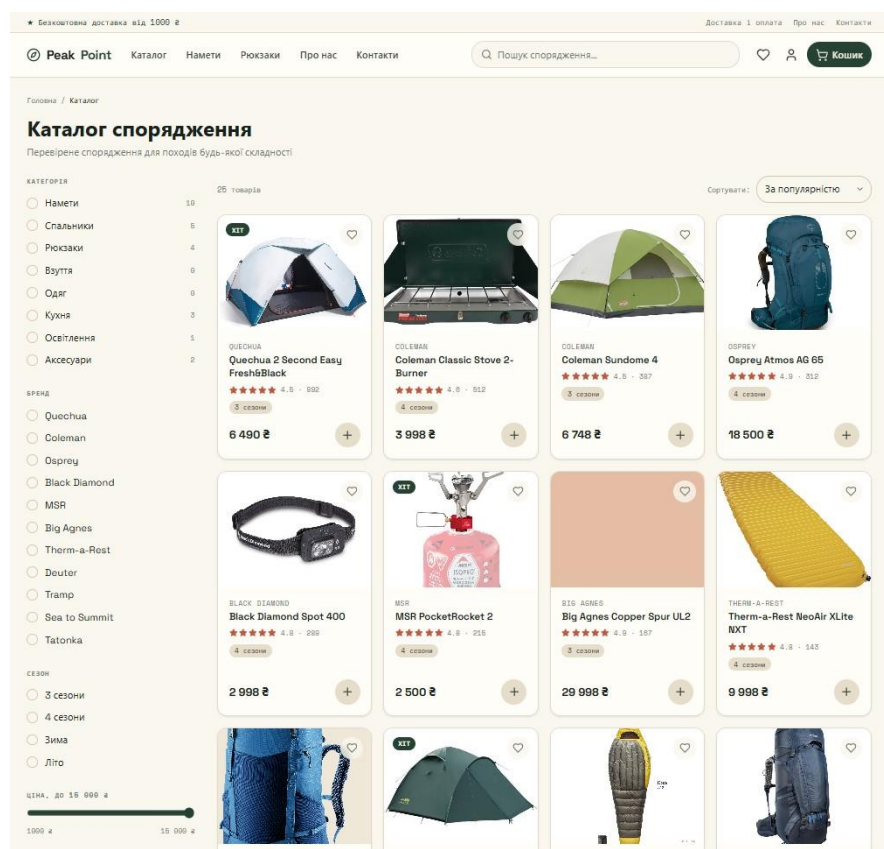


Рисунок 2.5 — Сторінка каталогу з активними фільтрами

Зовнішній вигляд картка товару з характеристиками та відгуками наведено на рисунку 2.6, сторінки блоку маршрутів – на рисунку 2.7.

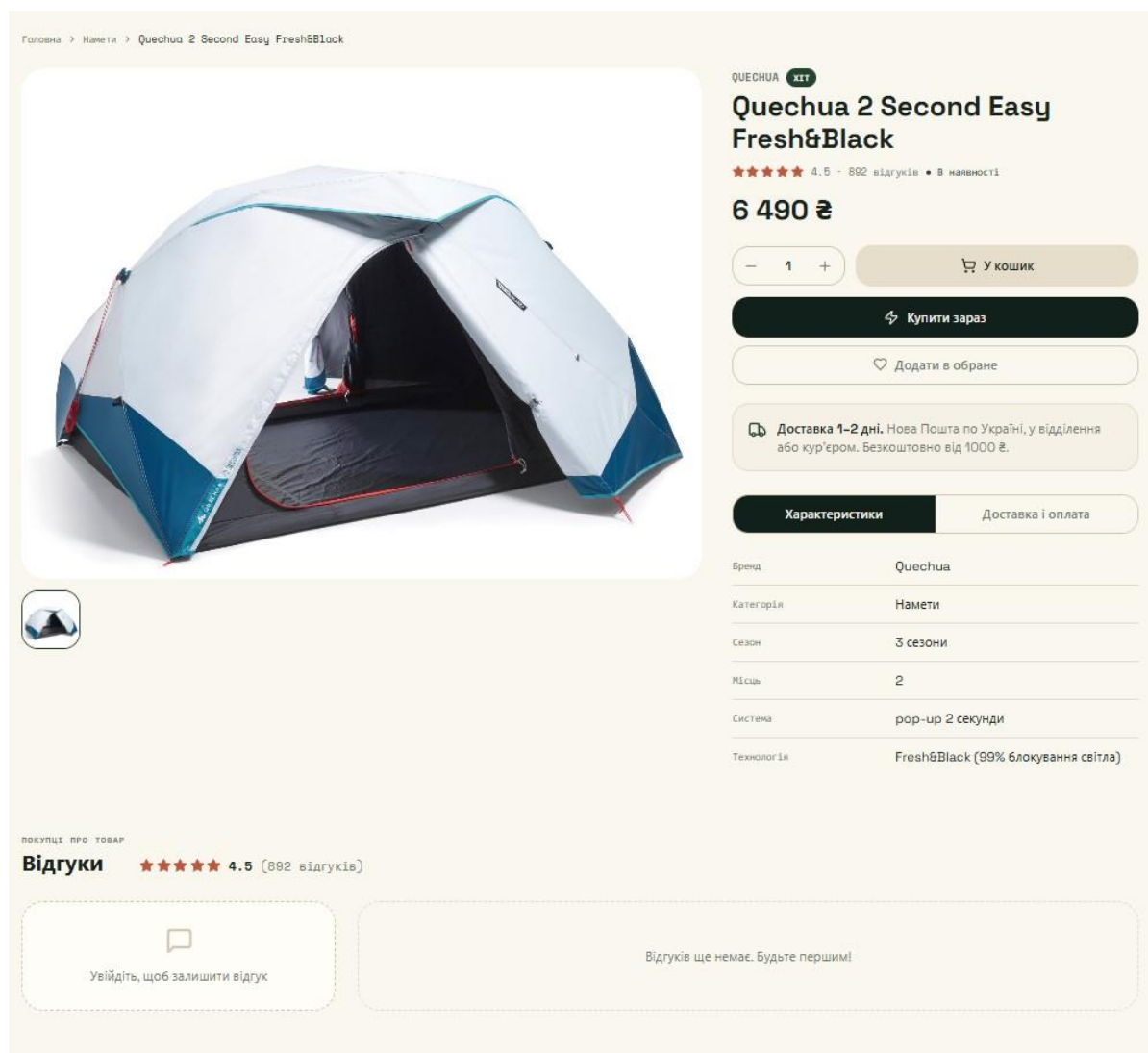


Рисунок 2.6 — Детальна картка товару з характеристиками та відгуками

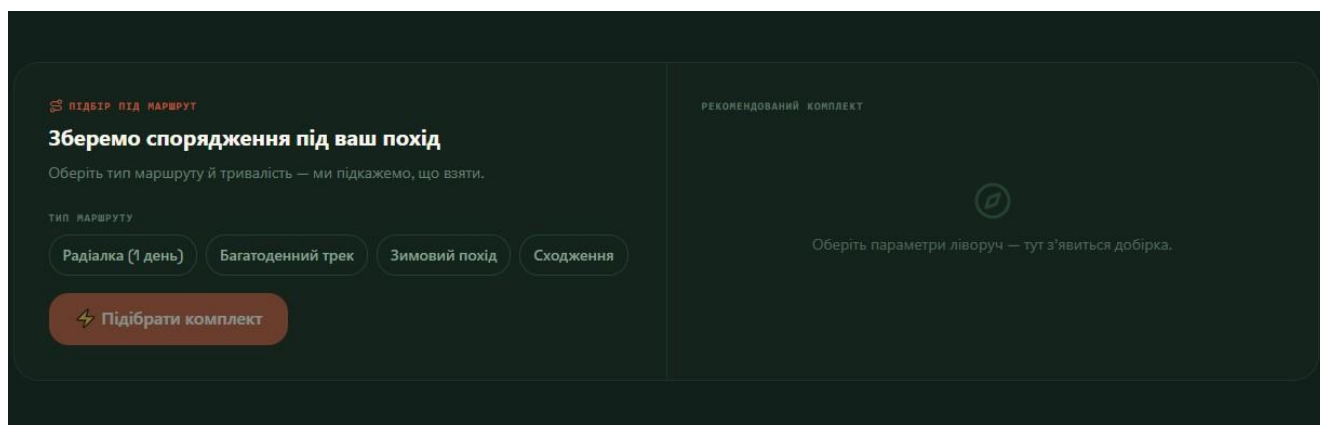


Рисунок 2.7 — Блок рекомендацій спорядження за типом маршруту

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

2.5.3 Тестування кошика та оформлення замовлення

Третій блок є ключовим для комерційної частини системи. Тестувався повний сценарій покупки від додавання товару до кошика до отримання підтвердження замовлення. Перевірялась коректність збереження стану кошика у localStorage між сесіями, логіка застосування промокоду та гостьове оформлення замовлення без реєстрації в системі. Окремо перевірялась серверна логіка обчислення загальної суми замовлення: система коректно ігнорує будь-які ціни, передані клієнтом, та самостійно зчитує актуальні значення з бази даних, що унеможлиблює підробку суми з боку покупця. Також було протестовано поведінку форми оформлення замовлення при введенні некоректних даних — система відображає повідомлення про помилки валідації без відправлення запиту на сервер. Після успішного оформлення перевірено отримання email-підтвердження замовлення та коректне відображення його деталей на сторінці підтвердження. Зовнішній вигляд сторінки кошика наведено на рисунку 2.8.

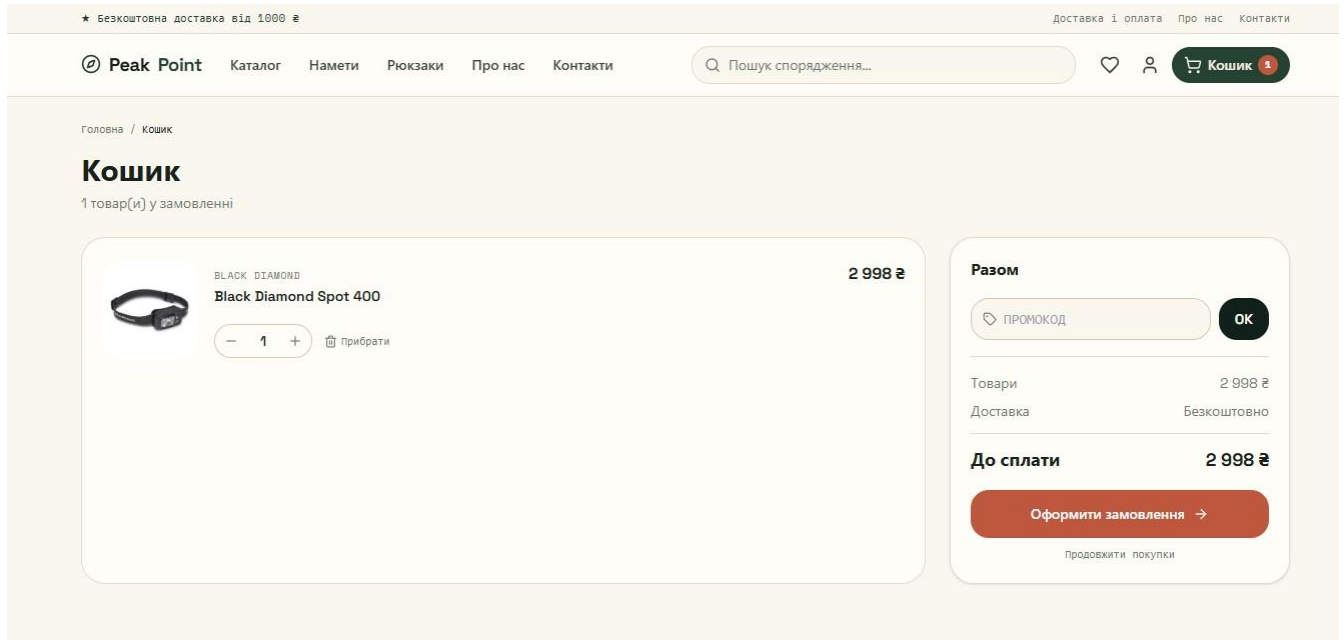


Рисунок 2.8 — Сторінка кошика з переліком товарів

Зовнішній вигляд сторінки форми оформлення замовлення — на рисунку 2.9, сторінки підтвердження замовлення — на рисунку 2.10.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

★ Безкоштовна доставка від 1000 ₴ Доставка і оплата Про нас Контакти

Peak Point [Каталог](#) [Намети](#) [Рюкзаки](#) [Про нас](#) [Контакти](#) [♡](#) [👤](#) [🛒 Кошик 1](#)

Головна / Кошик / Оформлення

Оформлення замовлення

1 Доставка — 2 Оплата

Контакт і отримувач

ІМ'Я ПРІЗВИЩЕ

ТЕЛЕФОН EMAIL

Спосіб доставки

Нова Пошта, відділення Безкоштовно
1-2 дні · по всій Україні

Кур'єр до дверей 120 ₴
1-3 дні · великі міста

МІСТО ВІДДІЛЕННЯ / ПОШТОМАТ

Далі: оплата →

Ваше замовлення

Black Diamond Spot 400 2 998 ₴
× 1

Доставка Безкоштовно

До сплати **2 998 ₴**

Рисунок 2.9 — Сторінка оформлення замовлення

★ Безкоштовна доставка від 1000 ₴ Доставка і оплата Про нас Контакти

Peak Point [Каталог](#) [Намети](#) [Рюкзаки](#) [Про нас](#) [Контакти](#) [♡](#) [👤](#) [🛒 Кошик 1](#)

Головна / Кошик / Оформлення

Оформлення замовлення

✓ Доставка — 2 Оплата

← Назад до доставки

Спосіб оплати

Картка онлайн
Visa · Mastercard · Apple Pay

При отриманні
накладений платіж

🛡 Підтвердити замовлення · 2 998 ₴

Натискаючи, ви погоджуєтесь з умовами PeakPoint

Ваше замовлення

Black Diamond Spot 400 2 998 ₴
× 1

Доставка Безкоштовно

До сплати **2 998 ₴**

Рисунок 2.10 — Сторінка підтвердження оформленого замовлення

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

2.5.4 Тестування адміністративної панелі

Четвертий блок охоплює функціонал, доступний користувачам з роллю ADMIN. Перевірялись CRUD-операції над товарами (включно з завантаженням зображень), зміна статусів замовлень з автоматичною email-нотифікацією, управління типами маршрутів та прив'язаним спорядженням, обробка звернень з форми зворотнього зв'язку та управління промокодами.

Зовнішній вигляд адміністративної панелі управління товарами наведено на рисунку 2.11.

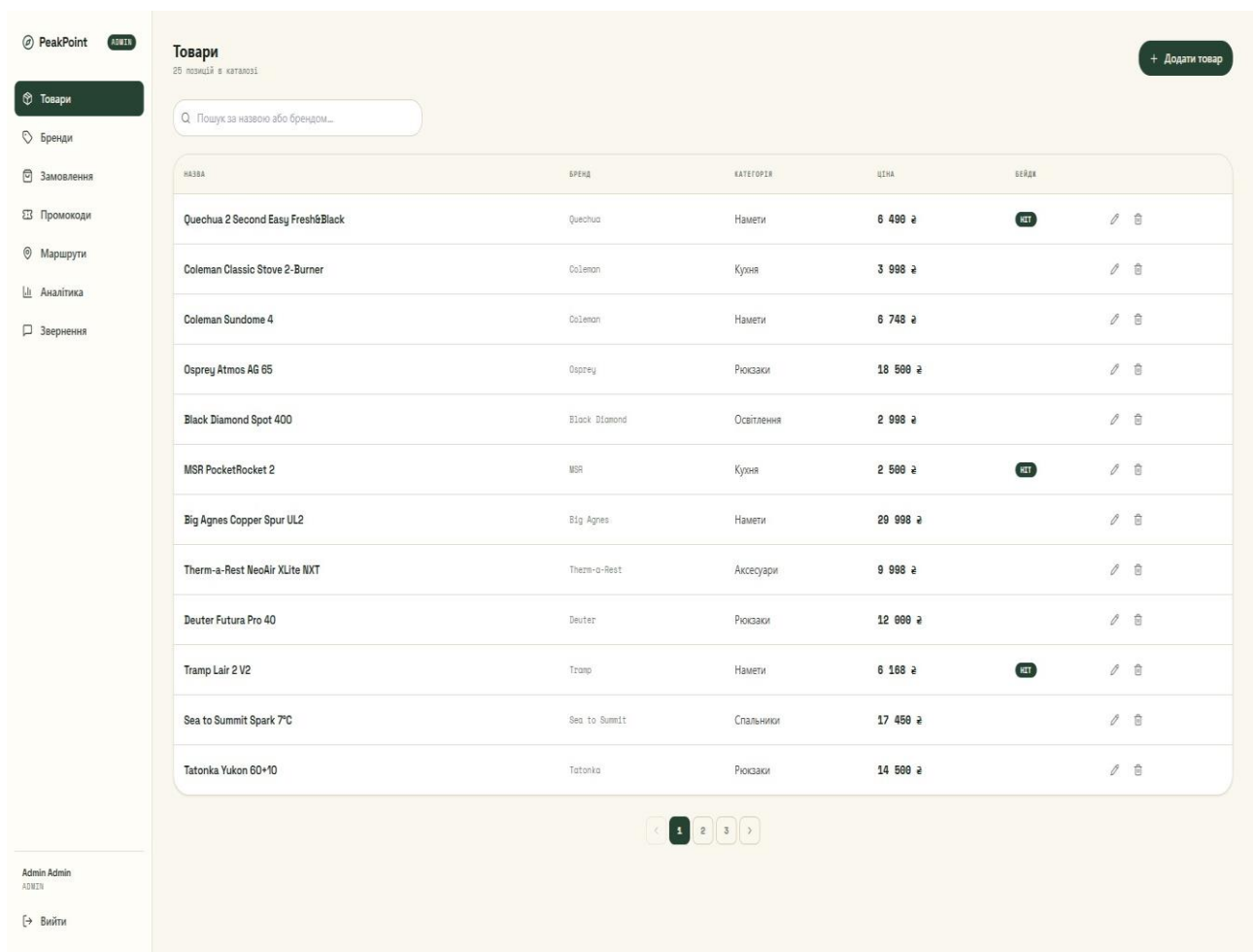


Рисунок 2.11 — Адміністративна панель управління товарами

Зовнішній вигляд адміністративної панелі управління замовленнями — на рисунку 2.12, панелі управління маршрутами — на рисунку 2.13.

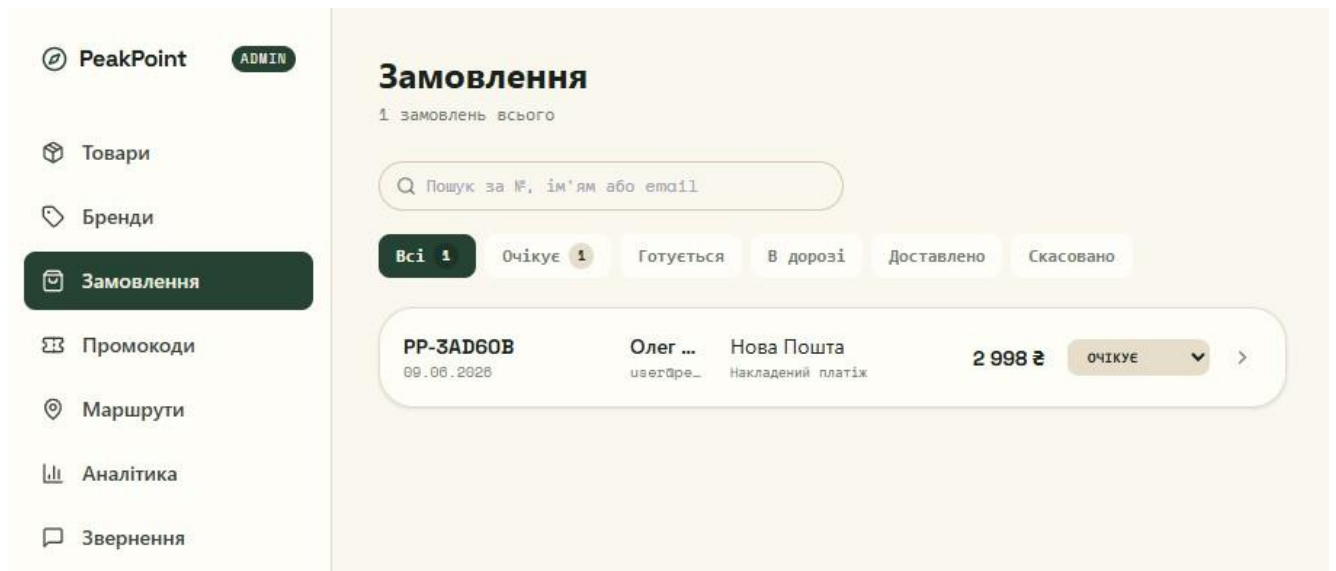


Рисунок 2.12 — Адміністративна панель управління замовленнями

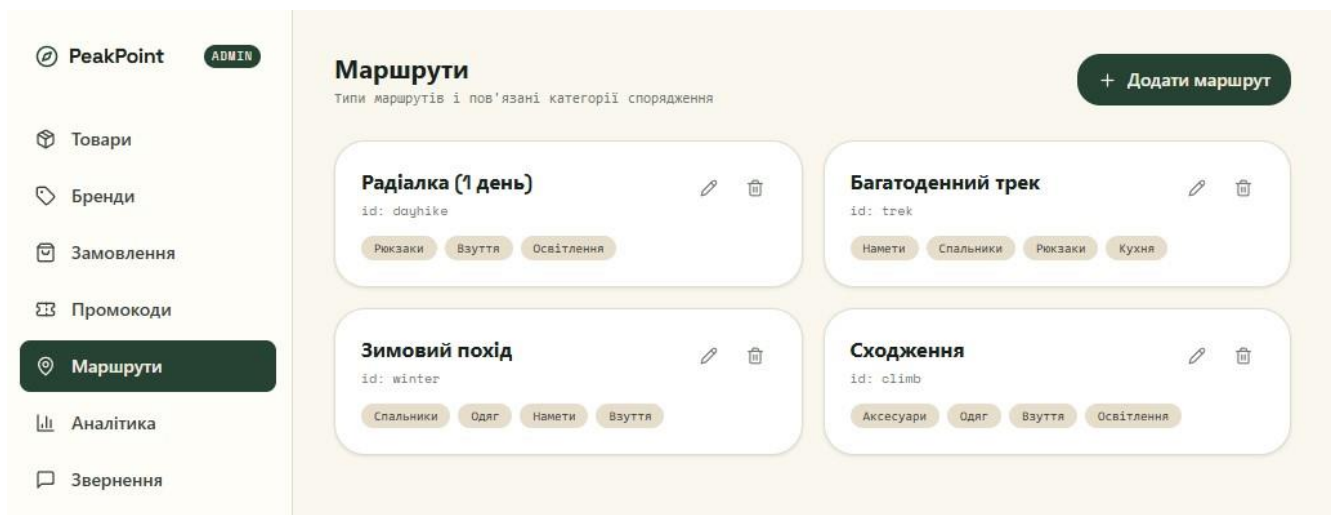


Рисунок 2.13 — Адміністративна панель управління маршрутами

2.5.5 Тестування списку бажань та відгуків та адаптивності

П'ятий блок об'єднує перевірку допоміжного функціоналу для авторизованих покупців — списку бажань та системи відгуків — а також крос-браузерне та адаптивне тестування. Адаптивність верстки перевірялась на трьох контрольних точках зламу: мобільній (375 px), планшетній (768 px) та десктопній (1440 px). На мобільному розширенні бічна панель фільтрів каталогу коректно трансформується у модальне вікно, а сітка карток переходить з триколонного в одноколонне відображення.

Зовнішній вигляд сторінки списку бажань наведено на рисунку 2.14, блоку відгуків на картці товару — на рисунку 2.15.

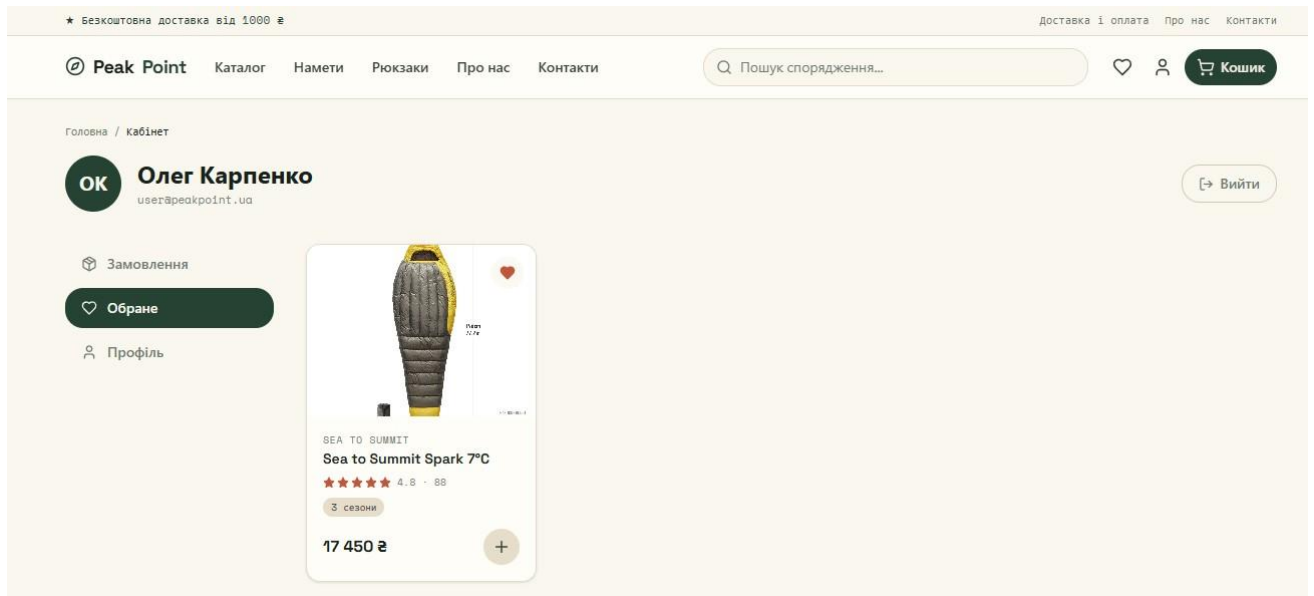


Рисунок 2.14 — Сторінка списку бажань покупця

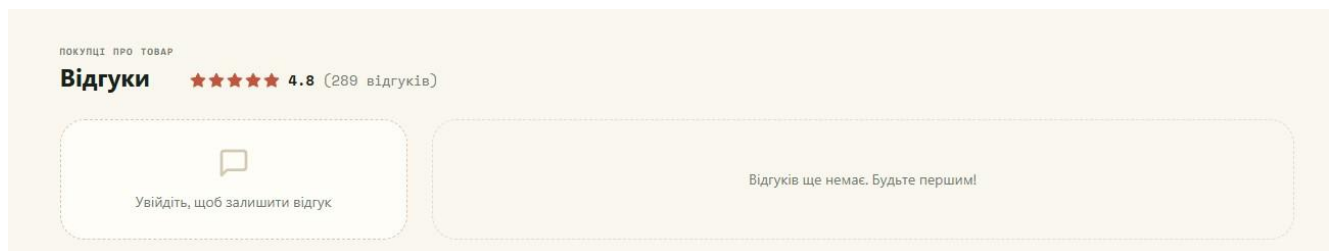


Рисунок 2.15 — Блок відгуків покупців на сторінці товару

За результатами мануального тестування інформаційна система коректно виконує всі визначені функціональні прецеденти для кожної рольової моделі: Гість, Покупець та Адміністратор. Захист маршрутів від несанкціонованого доступу функціонує відповідно до вимог рольової моделі. Клієнтська валідація форм запобігає відправці некоректних даних, а серверна валідація забезпечує другий рівень захисту. Адаптивна верстка коректно відображається у всіх перевірених браузерях і на всіх цільових розмірах екрану. Функціонал рекомендаційного модуля маршрутів та системи відгуків з автоматичним перерахунком рейтингу підтверджує свою відповідність технічному завданню.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розгортання інформаційної системи

Інструкція призначена для системних адміністраторів або розробників, які виконують первинне розгортання інформаційної системи «PeakPoint» на сервері. Розгортання охоплює три компоненти: серверну частину (Spring Boot JAR-файл), базу даних (PostgreSQL) та клієнтську частину (статична збірка React SPA). Усі команди наведено для операційної системи Ubuntu 24.04 LTS.

3.1.1 Вимоги до серверного середовища

Перед початком розгортання необхідно переконатися, що серверне середовище відповідає вимогам, наведеним у таблиці 3.1.

Таблиця 3.1 – Мінімальні та рекомендовані вимоги до серверного середовища

Компонент	Мінімальні вимоги	Рекомендовані вимоги
Процесор	2 ядра, 2.0 ГГц	4 ядра, 2.5 ГГц+
Оперативна пам'ять (RAM)	4 ГБ	8 ГБ
Дисковий простір (SSD)	10 ГБ вільного місця	20 ГБ вільного місця
Операційна система	Ubuntu 24.04 LTS / Debian 12	Ubuntu 24.04 LTS
Java (JDK)	JDK 17	JDK 21 LTS
СКБД PostgreSQL	Версія 14	Версія 16
Node.js (для збірки фронтенду)	v18 LTS	v20 LTS
Maven	3.9+	3.9+

Для доступу кінцевих користувачів до системи через мережу Інтернет сервер повинен мати статичну IP-адресу та зареєстроване доменне ім'я.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

Рекомендується також встановлення SSL/TLS-сертифіката (Let's Encrypt) для забезпечення захищеного HTTPS-з'єднання.

3.1.2 Підготовка серверного середовища

Підготовка середовища виконується у такій послідовності:

1. Оновлення системних пакетів та встановлення JDK 17:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y openjdk-17-jdk
java -version
```

2. Встановлення PostgreSQL:

```
sudo apt install -y postgresql postgresql-contrib
sudo systemctl start postgresql
sudo systemctl enable postgresql
```

3. Встановлення Node.js v20 LTS (необхідний лише для збірки фронтенду):

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
node -v && npm -v
```

4. Встановлення Maven для збірки серверної частини:

```
sudo apt install -y maven
mvn -version
```

3.1.3 Створення бази даних та користувача PostgreSQL

Для роботи системи необхідно створити окрему базу даних та окремого користувача PostgreSQL з обмеженими правами. Використання суперкористувача postgres для роботи застосунку є небезпечним і не рекомендується:

1. Перехід до консолі PostgreSQL від імені системного користувача postgres:

```
sudo -i -u postgres
```

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

```
psql
```

2. Створення користувача БД та бази даних:

```
CREATE USER peakpoint_user WITH PASSWORD 'StrongPassword123';  
CREATE DATABASE peakpoint OWNER peakpoint_user;  
GRANT ALL PRIVILEGES ON DATABASE peakpoint TO peakpoint_user;  
\q  
exit
```

3.1.4 Конфігурація та розгортання серверної частини

Серверна частина конфігурується через файл `application.yml`, розташований у директорії `src/main/resources/`. Перед збіркою необхідно налаштувати параметри підключення до бази даних та інші змінні середовища:

1. Клонування репозиторію серверної частини:

```
git clone https://github.com/sasha_gucal/PeakPoint_backend.git  
cd PeakPoint_backend
```

2. Налаштування файлу `application.yml`. Відкрити файл у текстовому редакторі та вказати параметри відповідно до налаштованого середовища. Ключові параметри, що потребують зміни:

```
spring:  
  datasource:  
    url: jdbc:postgresql://localhost:5432/peakpoint  
    username: peakpoint_user  
    password: StrongPassword123  
  
app:  
  jwt:  
    secret: <рядок-не-менше-256-біт-у-base64>  
    expiration: 900000 # 15 хвилин (мс)  
    refresh-expiration: 604800000 # 7 діб (мс)  
  upload:  
    base-url: https://peakpoint.com/uploads
```

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

```
dir: ./uploads/products
```

```
# SMTP (необов'язково; без нього email-нотифікації вимкнено)
```

```
spring:
```

```
mail:
```

```
host: smtp.gmail.com
```

```
port: 587
```

```
username: your@gmail.com
```

```
password: app-password
```

3. Збірка проєкту та створення виконуваного JAR-файлу:

```
mvn clean package -DskipTests
```

```
ls target/peakpoint-backend-*.jar
```

4. Запуск серверної частини. Рекомендується запуск як systemd-сервісу для автоматичного відновлення після перезавантаження сервера:

```
sudo nano /etc/systemd/system/peakpoint.service
```

Вміст файлу сервісу:

```
[Unit]
```

```
Description=PeakPoint Backend
```

```
After=network.target postgresql.service
```

```
[Service]
```

```
User=ubuntu
```

```
WorkingDirectory=/home/ubuntu/PeakPoint_backend
```

```
ExecStart=/usr/bin/java -jar target/peakpoint-backend-0.0.1-SNAPSHOT.jar
```

```
Restart=on-failure
```

```
RestartSec=10
```

```
[Install]
```

```
WantedBy=multi-user.target
```

5. Активація та запуск сервісу:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable peakpoint
```

```
sudo systemctl start peakpoint
```

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

```
sudo systemctl status peakpoint
```

При успішному старті у журналі сервісу з'явиться рядок «Started PeakPoint Application» та повідомлення DataInitializer про успішне заповнення бази даних початковими даними (якщо база порожня). Перевірити доступність API можна запитом до публічного ендпоінту:

```
curl http://localhost:8080/api/v1/products
```

3.1.5 Збірка та розгортання клієнтської частини

Клієнтська частина є статичним SPA-застосунком і після збірки являє собою набір HTML, CSS та JavaScript файлів, які можна роздавати будь-яким статичним веб-сервером. Рекомендується використання Nginx. Для збірки і налаштування клієнтської частини виконати наступні кроки:

1. Клонування репозиторію фронтенду та встановлення залежностей:

```
git clone https://github.com/sasha_gucal/PeakPoint_frontend.git
cd PeakPoint_frontend
npm install
```

2. Створення файлу змінних середовища для продуктивної збірки:

```
cp .env.example .env.production
nano .env.production
```

Вміст файлу .env.production:

```
VITE_API_BASE_URL=https://peakpoint.com/api/v1
VITE_APP_NAME=PeakPoint
```

3. Продуктивна збірка проєкту:

```
npm run build
ls dist/
```

4. Встановлення та налаштування Nginx для роздачі статичних файлів та проксування запитів до API:

```
sudo apt install -y nginx
sudo nano /etc/nginx/sites-available/peakpoint
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

Вміст конфігурації Nginx:

```
server {
    listen 80;
    server_name peakpoint.com;
    root /var/www/peakpoint;
    index index.html;

    # SPA fallback: усі маршрути повертають index.html
    location / {
        try_files $uri $uri/ /index.html;
    }

    # Проксування API-запитів до Spring Boot
    location /api/ {
        proxy_pass http://localhost:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    # Роздача завантажених зображень товарів
    location /uploads/ {
        alias /home/ubuntu/PeakPoint_backend/uploads/;
    }
}
```

5. Копіювання збірки у директорію Nginx та активація конфігурації:

```
sudo mkdir -p /var/www/peakpoint
sudo cp -r dist/* /var/www/peakpoint/
sudo ln -s /etc/nginx/sites-available/peakpoint /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```

Після виконання всіх кроків система доступна за адресою <http://peakpoint.com>.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

3.2 Інструкція з наповнення інформаційної системи

Інструкція призначена для контент-менеджерів та адміністраторів, відповідальних за підтримку актуального стану каталогу товарів, маршрутів та іншого контенту системи. Всі операції виконуються через захищену адміністративну панель, доступну за адресою /admin після входу з обліковим записом в ролі ADMIN.

Вхід до адміністративної панелі здійснюється через стандартну форму авторизації системи. Після успішного входу в меню навігації з'являються розділи «Товари», «Бренди», «Замовлення», «Промокоди», «Маршрути», «Аналітика» та «Звернення». Зовнішній вигляд адміністративної панелі наведено на рисунку 3.1.

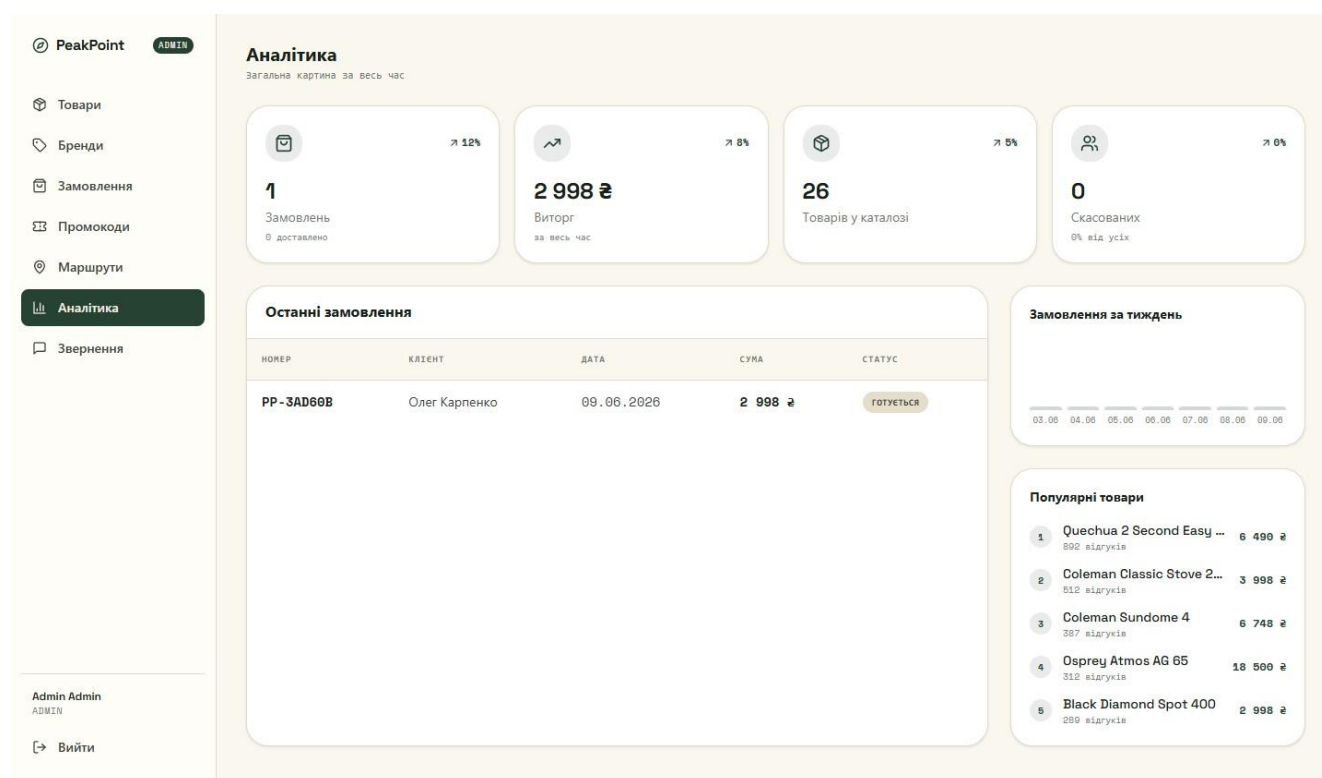


Рисунок 3.1 — Головна сторінка адміністративної панелі після входу

3.2.1 Управління брендами

Перед додаванням товарів рекомендується спочатку впевнитись у наявності необхідних категорій та брендів. Для додавання нового бренду необхідно перейти

до відповідного розділу адмін-панелі, натиснути кнопку «Додати бренд» та заповнити форму: назва бренду. Видалення бренду заблоковано системою, якщо до нього прив'язані товари — система відображає відповідне повідомлення про помилку. Зовнішній вигляд сторінки управління брендами наведено на рисунку 3.2, форми додавання бренду наведено на рисунку 3.3.

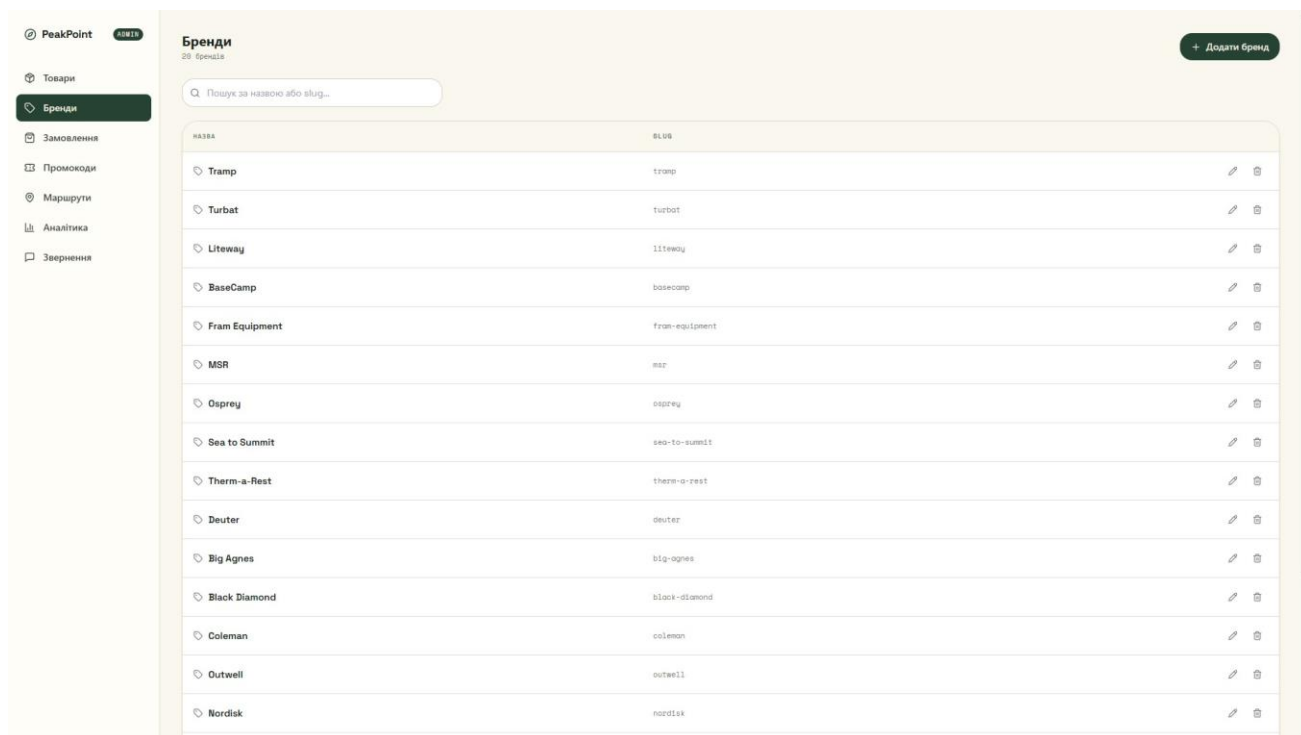


Рисунок 3.2 — Сторінка управління брендами в адміністративній панелі

Рисунок 3.3 — Форма додавання нового бренду

3.2.2 Додавання та редагування товарів

Розділ управління товарами є основним інструментом наповнення каталогу. Для переходу до нього слід обрати пункт «Товари» у бічному меню адміністративної панелі. Зовнішній вигляд сторінки управління товарами наведено на рисунку 3.4.

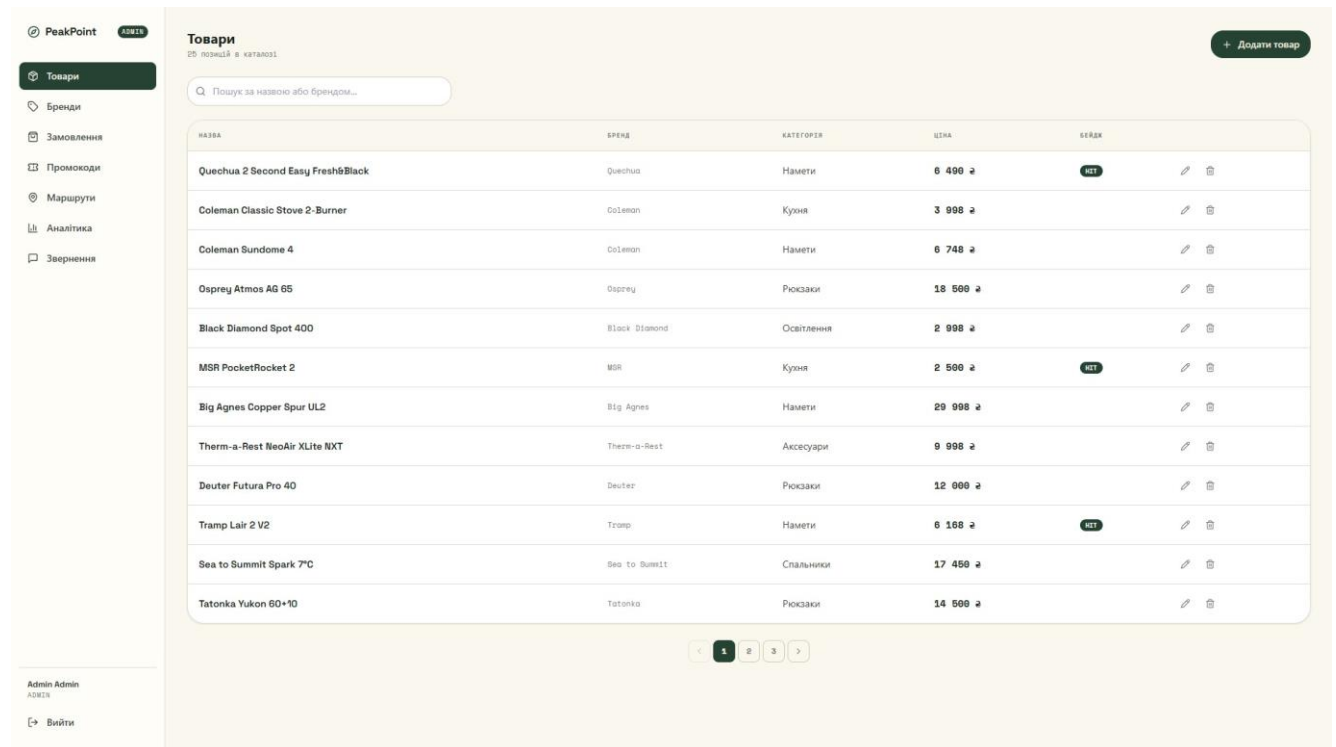


Рисунок 3.4 — Сторінка управління товарами в адміністративній панелі

Для додавання нового товару необхідно виконати такі дії:

1. Натиснути кнопку «Додати товар» у верхній частині сторінки.
2. Заповнити форму товару. Обов'язкові поля: назва (до 255 символів), короткий опис (до 500 символів, використовується у картці каталогу та пошуку), повний опис, ціна у гривнях (ціле число), залишок на складі, категорія, бренд. Необов'язкові поля: стара ціна (для відображення знижки), сезони придатності (можна обрати кілька), бейдж-мітка (SALE, NEW, BESTSELLER).
3. Додати технічні характеристики товару у блоці «Специфікації». Кожна характеристика складається з пари «назва — значення» (наприклад: «Вага — 2.3

кг», «Водостійкість — 3000 мм»). Для додавання нової характеристики слід натиснути кнопку «+».

4. Натиснути кнопку «Зберегти» для створення запису товару в базі даних.

5. Після збереження товару стає доступна функція завантаження зображень. Перейти до картки щойно створеного товару та натиснути «Завантажити зображення». Система приймає файли форматів JPEG та PNG. Перше завантажене зображення автоматично відображається як головне у каталозі. Рекомендований розмір зображень: не менше 800×800 пікселів.

Зовнішній вигляд форми додавання товару наведено на рисунку 3.5, форми завантаження зображень — на рисунку 3.6.

Новий товар ×

НАЗВА
Coleman Sundome 4

БРЕНД
Coleman

КАТЕГОРІЯ
Намети

ЦІНА (₴)
6748

СТАРА ЦІНА (₴)
7350

СЕЗОН
3 сезони

БЕЙДЖ
new

КОРОТКИЙ ОПИС
Намет Coleman Sundome 4-місний туристичний водонепроникний кемпінговий для відпочинку водостійкий.

ХАРАКТЕРИСТИКИ + Додати рядок

Базовий колір: зелений

Конструкція: двшарова

В НАЯВНОСТІ

Скасувати Далі: зображення →

Рисунок 3.5 — Форма додавання нового товару

Зм.	Арк.	№ докум.	Підпис	Дата

2026.КВР.122.421.07.00.00 ПЗ

Арк.

55



Рисунок 3.6 — Форма завантаження зображень товару

Для редагування існуючого товару слід знайти його у переліку та натиснути кнопку «Редагувати». Форма редагування аналогічна формі додавання; всі поля доступні для змін. Прапор «В наявності» дозволяє тимчасово приховати товар з публічного каталогу без його видалення з бази даних — зручно при відсутності товару на складі.

Видалення товару виконується через кнопку «Видалити» у картці товару. Операція є незворотною: система видаляє всі пов'язані зображення як з бази даних, так і з файлового сховища на диску.

3.2.3 Обробка замовлень

Розділ «Замовлення» відображає повний перелік усіх замовлень у системі з інформацією про покупця, загальну суму та поточний статус. Зовнішній вигляд сторінки управління замовленнями наведено на рисунку 3.7.

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

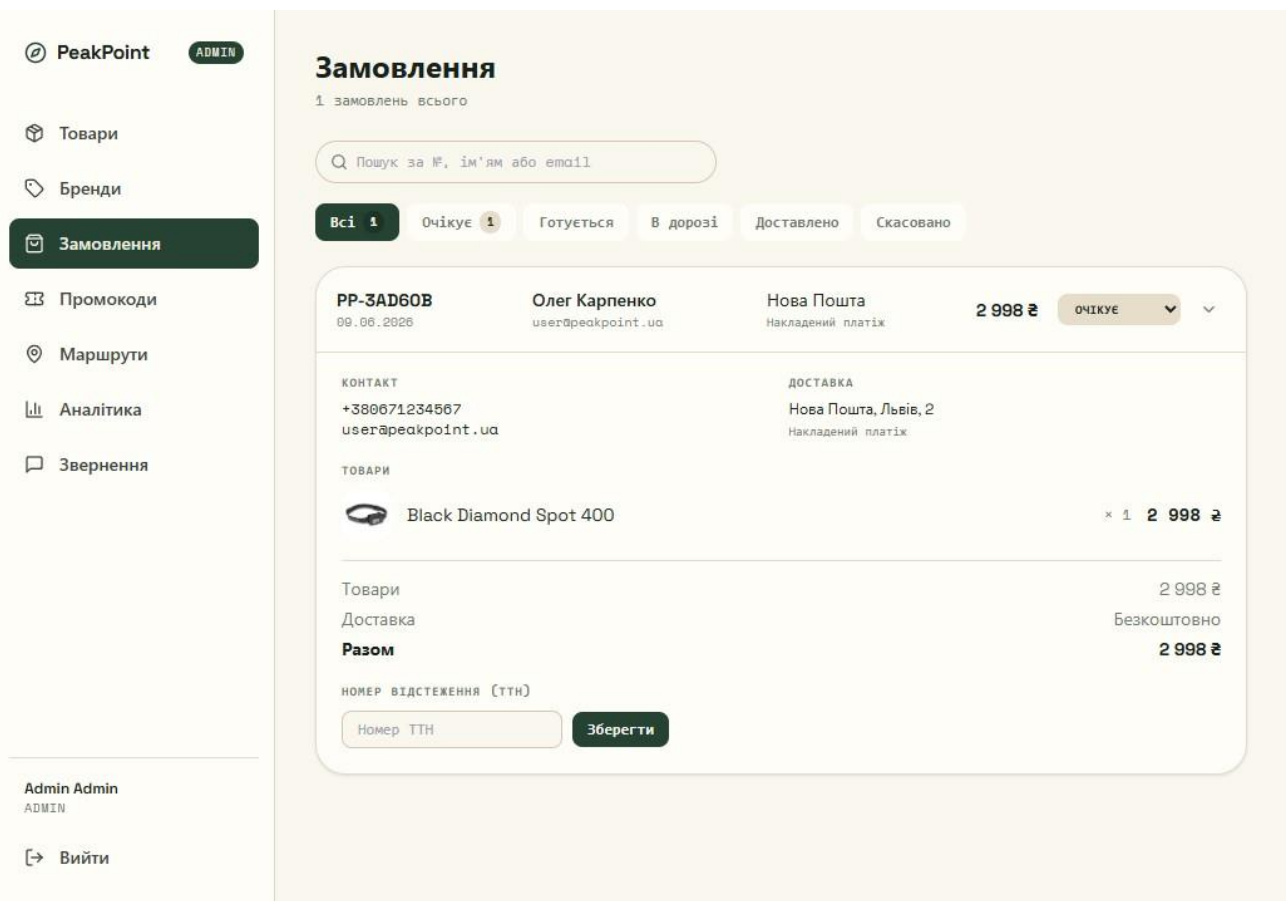


Рисунок 3.7 — Сторінка управління замовленнями в адміністративній панелі

Обробка замовлення виконується у такій послідовності. При надходженні нового замовлення воно отримує статус PENDING («Очікує»). Після комплектації та передачі посилки перевізнику менеджер змінює статус на SHIPPED («Відправлено»). Система автоматично надсилає email-нотифікацію покупцю при кожній зміні статусу. Після підтвердження отримання замовлення статус змінюється на DELIVERED («Доставлено»). У разі відмови покупця або відсутності товару на складі встановлюється статус CANCELLED («Скасовано»).

3.2.4 Управління туристичними маршрутами

Розділ «Маршрути» дозволяє керувати типами туристичних маршрутів, що відображаються на головній сторінці та слугують основою рекомендаційного модуля. Зовнішній вигляд сторінки управління маршрутами наведено на рисунку 3.8.

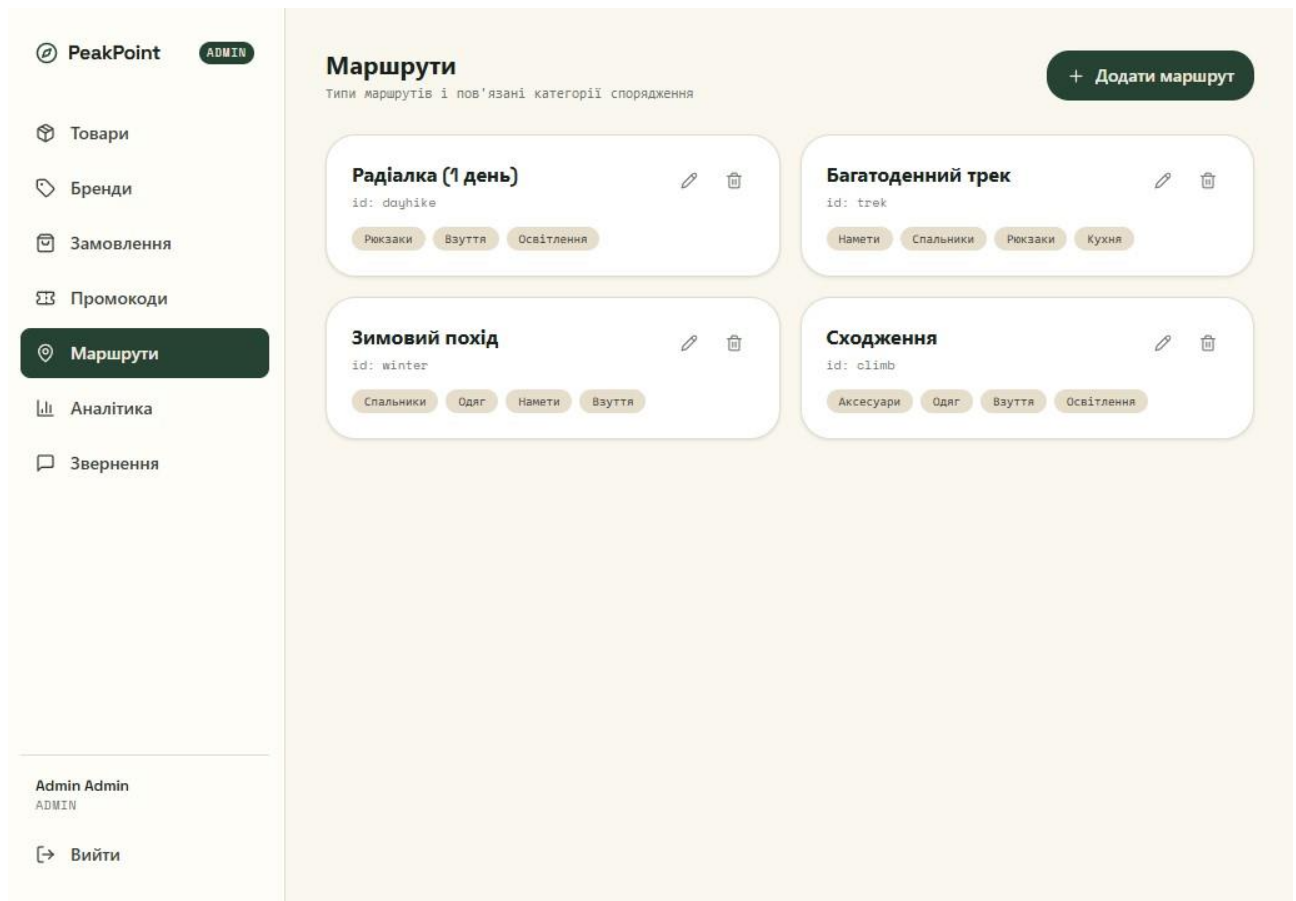


Рисунок 3.8 — Сторінка управління туристичними маршрутами

Для створення нового типу маршруту слід натиснути «Додати маршрут» та заповнити форму: назва (наприклад, «Гірський трекінг»), необхідні вибрати категорії спорядження. Система відобразить ці товари при виборі відповідного маршруту покупцем.

3.2.5 Управління промокодами

Розділ управління промокодами доступний лише адміністратору. Для створення нового промокоду слід заповнити форму: код (рядок у верхньому регістрі, наприклад SUMMER20), відсоток знижки (ціле число від 1 до 100), дата закінчення терміну дії та прапор активності. Деактивація промокоду виконується зміною прапора «Активний» без видалення запису — це зберігає можливість аудиту використаних кодів у вже оформлених замовленнях. Вигляд сторінки управління промокодами наведено на рисунку 3.9.

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

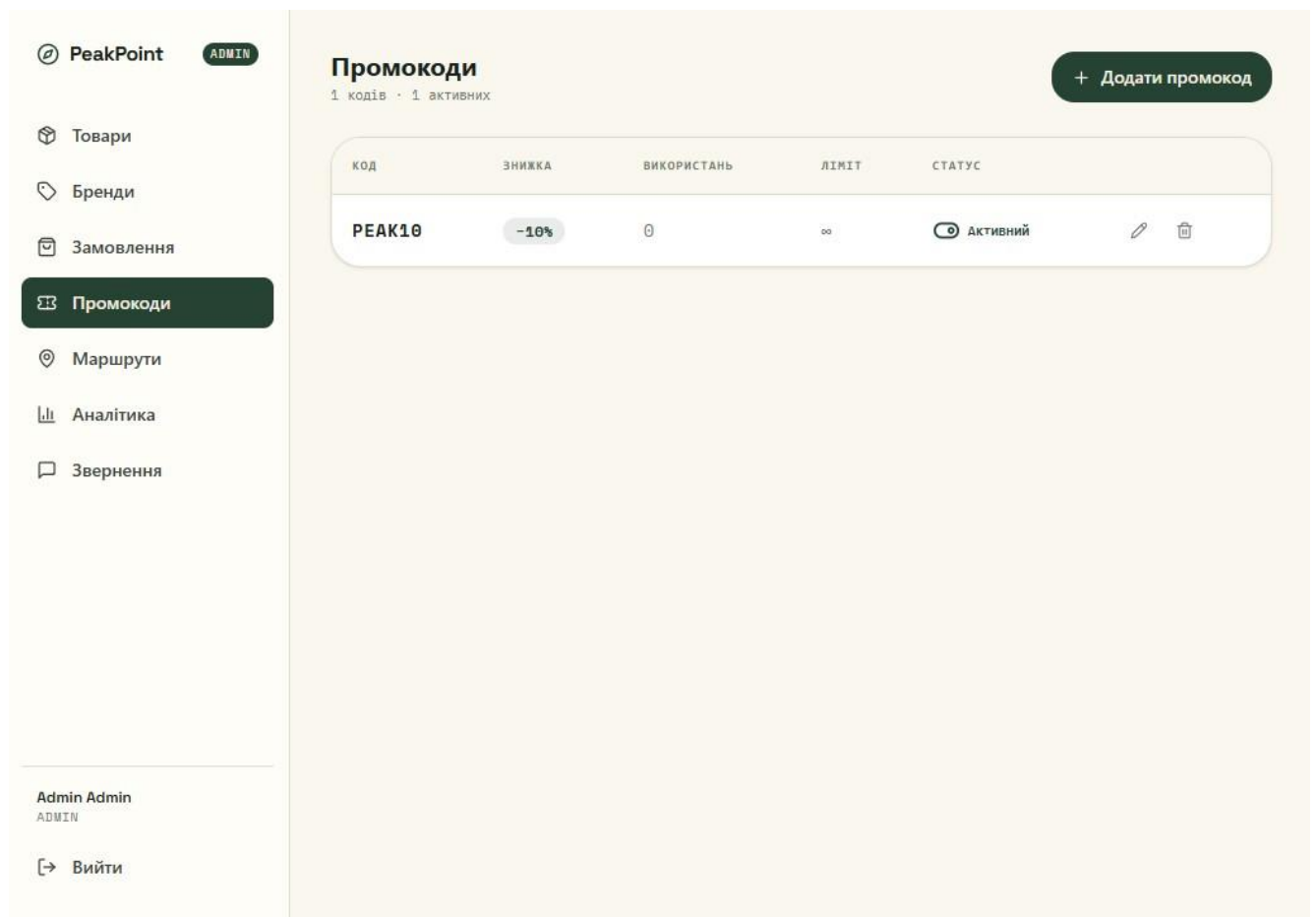


Рисунок 3.9 — Сторінка управління промокодами

3.3 Інструкція з популяризації та підтримки інформаційної системи

Пошукова оптимізація є основним безкоштовним каналом залучення нових відвідувачів через пошукові системи Google та Bing. Оскільки клієнтська частина «PeakPoint» реалізована як SPA, пошукові роботи можуть не індексувати динамічно згенерований контент. Для вирішення цієї проблеми рекомендується налаштувати серверний рендеринг (SSR) або використати інструмент попереднього рендерингу (pre-rendering), наприклад Prerender.io, у конфігурації Nginx.

На рівні контенту необхідно виконати такі заходи. Для кожної сторінки товару слід заповнити мета-теги title та description, що відображаються у результатах пошуку. Назви товарів та їх описи мають містити ключові слова, які потенційні покупці використовують при пошуку: конкретні назви моделей,

матеріали, характеристики (наприклад, «намет двомісний водонепроникний 3000 мм»). Зображення товарів повинні мати описові атрибути alt із назвою товару та брендом.

Технічні SEO-заходи включають: реєстрацію сайту в Google Search Console та Bing Webmaster Tools для моніторингу індексації та пошукових запитів; генерацію та подання файлу sitemap.xml, що містить усі URL товарів і категорій; налаштування файлу robots.txt для коректного направлення пошукових роботів; підключення Google Analytics або Plausible Analytics для відстеження трафіку та поведінки користувачів.

Для цільової аудиторії «PeakPoint» — туристів, альпіністів та прихильників кемпінгу — найбільш ефективними майданчиками є Instagram та TikTok. Рекомендується регулярна публікація контенту: фотографії та відео використання товарів у реальних умовах (походах, таборах), огляди новинок асортименту, корисні поради з підготовки до походів та вибору спорядження. Такий контент органічно поширюється в цільовій спільноті та формує довіру до бренду.

Усі пости в соціальних мережах мають містити посилання на відповідні сторінки сайту або на головну сторінку. У профілях соціальних мереж необхідно вказати URL платформи «PeakPoint». Для відстеження ефективності рекламних кампаній рекомендується використовувати UTM-мітки в посиланнях, що дозволить аналітиці Google Analytics чітко розмежувати трафік з різних джерел.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини кваліфікаційної роботи є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки інформаційної системи «PeakPoint», прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є інформаційна система електронної комерції для реалізації товарів для кемпінгу «PeakPoint».

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення розробки інформаційної системи

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки інтернет-магазину товарів для кемпінгу «PeakPoint». Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 4.1.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

Таблиця 4.1 - Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Планування та аналіз	Кер. проєкту (Pm)	4
		Інженер (11)	8
2	Розробка технічного завдання	Кер. проєкту (Pm)	4
		Інженер (11)	8
3	Дизайн інтерфейсу	Інженер (11)	16
		Інженер (12)	16
4	Розробка функціоналу	Інженер (11)	40
5	Тестування та відладка	Тестувальник	20
6	Документування	Інженер (11)	8
7	Розгортання та підтримка	Інженер (12)	8
8	Управління проєктом	Кер. проєкту (Pm)	8
Разом			140

Сумарний час виконання операцій технологічного процесу становить 140 годин.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки інтернет-магазину товарів для кемпінгу «PeakPoint».

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та діяльності підприємства.

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн.}} = T_c \cdot K_r, \quad (4.1)$$

де T_c – тарифна ставка, грн. (приймаємо для керівника проекту (Pm) – 450 грн./год, , інженера (I2) – 250 грн./год.), інженера (I1) – 150 грн./год., тестувальник – 120 грн./год.); K_r – кількість відпрацьованих годин.

Отже основна заробітна плата для:

- керівник проекту (Pm): $Z_{\text{осн1}} = 450 \cdot 16 = 7200$ грн.;
- інженера (I2): $Z_{\text{осн2}} = 250 \cdot 24 = 6000$ грн.;
- інженера (I1): $Z_{\text{осн3}} = 150 \cdot 80 = 12000$ грн.;
- тестувальника: $Z_{\text{осн4}} = 120 \cdot 20 = 2400$ грн.

Сумарна основна заробітна плата становить:

$$Z_{\text{осн}} = 7200 + 6000 + 12000 + 2400 = 27600 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} \cdot K_{\text{допл.}}, \quad (4.2)$$

де $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам (приймаємо 10%).

Отже додаткова заробітна плата по категоріях працівників становить:

- керівника проекту $Z_{\text{дод1}} = 7200 \cdot 0,1 = 720$ грн.;
- інженера (I2): $Z_{\text{осн2}} = 6000 \cdot 0,1 = 600$ грн.;
- інженера (I1): $Z_{\text{осн3}} = 12000 \cdot 0,1 = 1200$ грн.;
- тестувальника $Z_{\text{дод4}} = 2400 \cdot 0,1 = 240$ грн.

Загальна додаткова заробітна плата становить:

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

$$З_{\text{дод}} = 720 + 600 + 1200 + 240 = 2760 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($V_{\text{о.п.}}$) визначаються за формулою:

$$V_{\text{о.п.}} = Z_{\text{осн.}} + Z_{\text{дод.}} \quad (4.3)$$

$$V_{\text{о.п.}} = 27600 + 2760 = 30360 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$V_{\text{ЄСВ}} = V_{\text{о.п.}} \cdot 0,22 \quad (4.4)$$

$$V_{\text{ЄСВ.}} = 30360 \cdot 0,22 = 6679,20 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблицю 4.2.

Таблиця 4.2 - Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн.
		Тариф-на ставка, грн.	К-сть годин, год.	Фактично нарах. зарплати, грн.			
1	Кер. проекту (Pm)	450	16	7200	720	-	-
2	Інженера (I2)	250	24	6000	600	-	-
3	Інженера (I1)	150	80	12000	1200	-	-
4	Тестувальник	120	20	2400	240	-	-
Разом				27600	2760	6679,20	37039,20

Отже, загальні витрати на оплату праці становлять 37039,20 грн.

4.3 Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_{\text{в}} = W \cdot T \cdot S, \quad (4.5)$$

де W – необхідна потужність, кВт; T – кількість годин роботи обладнання; S – вартість кіловат-години електроенергії (приймаємо 15,94 грн).

При розробці використовувався 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні: комп'ютер – 0,5 кВт/год.

$$Z_{\text{в}} = 0,5 \cdot 140 \cdot 15,94 = 1115,80 \text{ грн.}$$

Витрати на електроенергію становлять 1115,80 грн.

4.4 Розрахунок суми амортизаційних відрахувань розробки інформаційної системи

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення. Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_B \cdot H_A}{100\%} \cdot T, \quad (4.6)$$

де A – амортизаційні відрахування за звітний період, грн.; B_B – балансова вартість групи основних фондів на початок звітного періоду, грн.; T - кількість годин роботи обладнання, год.; H_A – норма амортизації, 0,04%.

Оскільки для написання програми та її тестування використовується один ПК вартістю 42500 грн., тоді сума амортизаційних відрахувань становить:

$$A = \frac{42500 \cdot 0,04}{150} \cdot 140 = 1586,67 \text{ грн.}$$

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці. В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = B_{оп} \cdot 0,2 \dots 0,6, \quad (4.7)$$

де H_B – накладні витрати (прийmemo 40%).

$$H_B = 30360 \cdot 0,4 = 12144 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості розробки інформаційної системи

Для складання кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.3.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

Таблиця 4.3 - Кошторис витрат розробки інформаційної системи «PeakPoint»

№	Зміст витрат	Сума, грн.	В % до загальної суми
1	Витрати на оплату праці	37039,20	71,38
3	Витрати на електроенергію	1115,80	2,15
4	Амортизаційні відрахування	1586,67	3,06
5	Накладні витрати	12144,00	23,41
6	Собівартість	51885,67	100

Собівартість (C_B) НДР розраховуємо за формулою:

$$C_B = V_{O.P.} + Z_e + A + H_B \quad (4.8)$$

Отже, собівартість дорівнює:

$$C_B = 37039,20 + 1115,80 + 1586,67 + 12144 = 51885,67 \text{ грн.}$$

4.7 Розрахунок ціни розробки інформаційної системи

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка.

Ціну робіт можна визначити за формулою:

$$Ц = C_B \cdot (1 + P_{рен}) \cdot (1 + ПДВ), \quad (4.9)$$

де C_B – собівартість розробки; $P_{рен}$ – рівень рентабельності, (30 %); ПДВ – ставка

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

податку на додану вартість, (20 %).

$$Ц = 51885,67 \cdot (1 + 0,3) \cdot (1 + 0,2) = 80941,65 \text{ грн.}$$

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності ($T_{ок}$).

$$ЧТВ = -C_B + \sum_{i=1}^t \frac{\Gamma_{II}}{(1+i)^i} \quad (4.10)$$

де C_B – собівартість розробки; Γ_{II} – грошовий потік за t – ий рік; t – відповідний рік проєкту; i - величина дисконтної ставки (10%).

$$ЧТВ = -51885,67 + \frac{29055,98}{(1+0,1)} + \frac{29055,98}{(1+0,1)^2} + \frac{29055,98}{(1+0,1)^3} = 20372,24 \text{ грн.}$$

Якщо $ЧТВ \geq 0$, то проєкт може бути рекомендований до впровадження. Термін окупності визначається за формулою:

$$T_{ок} = T_{пв} + \frac{N_B}{\Gamma_{II}} \quad (4.11)$$

де $T_{пв}$ – період до повного відшкодування витрат, років; N_B – невідшкодовані витрати на початок року, грн.; Γ_{II} – грошовий потік на початок року, грн.

$$T_{ок} = 2 + 1457,94 / 29055,98 = 2,05.$$

Всі дані внесемо в зведену таблицю 4.4.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.4 - Техніко-економічні показники розробки інтернет-магазину товарів для кемпінгу «PeakPoint»

№ п/п	Показник	Значення
1.	Собівартість, грн.	51885,67
2.	Плановий прибуток, грн.	29055,98
3.	Ціна, грн.	80941,65
4.	Чиста теперішня вартість, грн.	20372,24
5.	Термін окупності, рік	2,05

Прибутковість проекту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення за два роки і один місяць. Отже, на основі отриманих показників можна зробити висновок, що розробка інформаційної системи для реалізації товарів для кемпінгу «PeakPoint» є доцільною з економічної точки зору.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

Розробка та експлуатація інформаційної системи «PeakPoint» здійснюється у офісних приміщеннях із застосуванням персональних комп'ютерів та іншої оргтехніки. Умови праці розробників і операторів системи регулюються законодавством України про охорону праці та вимогами пожежної безпеки, що становлять предмет розгляду цього розділу.

5.1 Гарантія прав на охорону праці

Основним законодавчим актом, що регулює відносини у сфері охорони праці між роботодавцем і працівником в Україні, є Закон України «Про охорону праці» [11]. Відповідно до статті 5 цього Закону, умови трудового договору не можуть містити положень, що суперечать законам та іншим нормативно-правовим актам з охорони праці.

Закон гарантує кожному працівникові такі права у сфері охорони праці. Право на безпечні та нешкідливі умови праці — роботодавець зобов'язаний створити умови праці на робочому місці відповідно до нормативно-правових актів, а також забезпечити дотримання вимог законодавства щодо прав працівників у галузі охорони праці. Право на відмову від виконання роботи — працівник має право відмовитись від дорученої роботи, якщо вона загрожує його життю або здоров'ю чи людей, котрі його оточують, або навколишньому середовищу і не передбачена трудовим договором [11].

Право на пільги та компенсації за важкі та шкідливі умови праці передбачає надання скороченого робочого часу, додаткових оплачуваних відпусток, пільгового пенсійного забезпечення та інших встановлених законодавством пільг і компенсацій за роботу в шкідливих умовах. Право на забезпечення засобами індивідуального захисту — роботодавець зобов'язаний за власні кошти забезпечувати засобами колективного та індивідуального захисту, змивальними засобами згідно з встановленими нормами [11].

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

Право на відшкодування шкоди у разі ушкодження здоров'я або у разі смерті працівника встановлює обов'язок роботодавця відшкодувати шкоду, заподіяну здоров'ю або майну відповідно до чинного законодавства.

Реалізація гарантованих прав забезпечується механізмом державного нагляду та громадського контролю у сфері охорони праці, право на здійснення якого надається уповноваженим представникам профспілок і трудового колективу [11].

Конституція України в статті 43 закріплює, що кожен має право на належні, безпечні і здорові умови праці. Використання примусової праці забороняється. Кожен має право на заробітну плату, не нижчу від визначеної законом. Громадянам гарантується захист від незаконного звільнення. [12]

Основним нормативно-правовим актом у сфері охорони праці є Закон України «Про охорону праці» від 14 жовтня 1992 року № 2694-XII (у редакції Закону № 229-IV від 21 листопада 2002 року, з подальшими змінами, остання редакція від 12 вересня 2025 року).

Цей Закон визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні. [11]

Відповідно до статті 1 Закону України «Про охорону праці», охорона праці — це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. [11]

Згідно зі статтею 5 Закону України «Про охорону праці», умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

повинні відповідати вимогам законодавства. [11]

Відповідно до статті 6 Закону України «Про охорону праці», працівник має право на:

- роботу на підприємстві, яке відповідає вимогам нормативно-правових актів з охорони праці;
- соціальне страхування від нещасних випадків на виробництві та професійних захворювань;
- одержання інформації від роботодавця та відповідних органів про умови та охорону праці на робочому місці, про наявність небезпечних і шкідливих виробничих факторів і можливі наслідки їх впливу на здоров'я та про заходи щодо захисту від впливу цих факторів;
- відмову від дорученої роботи, якщо створилась виробнича ситуація, небезпечна для його життя чи здоров'я або для людей, які його оточують, або для виробничого середовища чи довкілля;
- забезпечення засобами колективного та індивідуального захисту за рахунок роботодавця;
- звернення до органів державного нагляду за охороною праці та до органів прокуратури із скаргами на дії (бездіяльність) роботодавця щодо невиконання ним законодавчих та інших нормативних актів з охорони праці;
- участь у з'їздах, конференціях, нарадах, що проводяться з питань охорони праці.

Дія цього Закону поширюється на всіх юридичних та фізичних осіб, які відповідно до законодавства використовують найману працю, та на всіх працюючих. [11]

5.2 Засоби і заходи забезпечення пожежної безпеки виробничого об'єкта

Пожежна безпека виробничого об'єкта забезпечується системами запобігання пожежі, системами протипожежного захисту та організаційно-

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

технічними заходами відповідно до Кодексу цивільного захисту України та Правил пожежної безпеки в Україні . Основними напрямками забезпечення пожежної безпеки на підприємстві є обмеження кількості горючих речовин, попередження утворення горючого середовища, запобігання виникненню в горючому середовищі джерел запалювання, а також підтримання у справному стані засобів протипожежного захисту.

Основним законодавчим актом у сфері пожежної безпеки є Закон України «Про пожежну безпеку» від 17 грудня 1993 року № 3745-ХІІ. Закон визначає загальні правові, економічні та соціальні основи забезпечення пожежної безпеки на території України, регулює відносини державних органів, юридичних і фізичних осіб у цій галузі незалежно від виду їх діяльності та форм власності. [13]

Технічні вимоги щодо забезпечення пожежної безпеки на об'єктах різного призначення встановлює НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні», затверджені наказом Міністерства внутрішніх справ України від 30 грудня 2014 року № 1417 та зареєстровані в Міністерстві юстиції України 5 березня 2015 року за № 252/26697. Правила є обов'язковими для виконання всіма підприємствами, установами, організаціями незалежно від виду їх діяльності та форм власності, а також громадянами. [14]

Відповідно до вимог НАПБ А.01.001-2014, всі приміщення та території підприємств повинні бути забезпечені первинними засобами пожежогасіння — вогнегасниками та пожежним інвентарем. Норми оснащення приміщень вогнегасниками визначаються залежно від категорії пожежної небезпеки та площі приміщення. [14]

Офісні приміщення, де розміщено комп'ютерну техніку, відносяться до категорії В (пожежонебезпечні) за НАПБ Б.03.002-2007. Для таких приміщень рекомендується застосовувати вуглекислотні вогнегасники типу ВВК-2 або ВВК-5, які є безпечними для електронного обладнання та не залишають корозійних залишків після застосування. Порошкові вогнегасники також допустимі, однак при їх використанні поблизу комп'ютерної техніки можлива її пошкодження. [15]

Відповідно до додатку 1 НАПБ А.01.001-2014, норма оснащення

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

вуглекислотними вогнегасниками для офісних приміщень площею до 50 кв. м становить 1 вогнегасник ВВК-2 або ВВК-5. Для приміщень більшої площі встановлюється один вогнегасник на кожні 50 кв. м. Вогнегасники повинні розміщуватися у легкодоступних, добре видимих місцях, де виключено вплив прямих сонячних променів та безпосередній вплив опалювальних приладів. [14]

Огляд і перевірка вогнегасників здійснюються не рідше одного разу на рік. Вогнегасники з вичерпаним терміном придатності або перезарядженням підлягають заміні або перезарядці у спеціалізованих організаціях. Стан первинних засобів пожежогасіння відображається у журналі технічного обслуговування, який ведеться відповідальним за пожежну безпеку підприємства. [14]

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
						74
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проєктовано та розроблено інформаційну систему електронної комерції для реалізації товарів для кемпінгу «PeakPoint». Система є повнофункціональною вебплатформою, що об'єднує функціонал сучасного інтернет-магазину з унікальним рекомендаційним модулем підбору спорядження на основі типів туристичних маршрутів.

У ході роботи вирішено такі завдання.

Проведено аналітичний огляд чотирьох існуючих вебплатформ у сфері туристичного спорядження на ринку України: «Gorgany», «Alp», «Tramp» та «Перевал». Аналіз виявив суттєву технологічну та функціональну прогалину: жодна з досліджених платформ не реалізує динамічний SPA-інтерфейс та не пропонує інструментів автоматичної рекомендації спорядження під конкретний маршрут. Ці висновки стали основою для формування технічного завдання та визначення конкурентних переваг системи «PeakPoint».

Сформовано технічне завдання, що визначає рольову модель системи (Гість, Покупець, Адміністратор), функціональні вимоги до кожної ролі, вимоги до надійності та безпеки, часові характеристики, а також технічні умови експлуатації.

Спроектовано реляційну базу даних PostgreSQL, що складається з 15 таблиць, об'єднаних у чотири функціональні домени: управління користувачами і безпекою, товарний каталог, обробка замовлень та рекомендаційний модуль маршрутів. Розроблено ER-діаграму, що відображає всі сутності та зв'язки між ними, включно з унікальним зв'язком «багато до багатьох» між типами маршрутів і товарами.

Розроблено серверну частину системи мовою Java 17 на базі фреймворку Spring Boot 3.3.4. Реалізовано REST API, що складається з 9 контролерів та охоплює 43 ендпоінти. Автентифікація побудована на стандарті JWT з механізмом refresh token rotation. Захист даних забезпечено хешуванням паролів алгоритмом BCrypt та дворівневою валідацією вхідних даних. API

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

задокументовано у стандарті OpenAPI 3.0 та доступне через інтерфейс Swagger UI.

Розроблено клієнтську частину системи як Single Page Application на базі React 18 з використанням React Router DOM v6, Tailwind CSS та Axios. Реалізовано адаптивний інтерфейс, коректний на мобільних (375 px), планшетних (768 px) та десктопних (1440 px) екранах. Архітектуру компонентів побудовано за принципом атомарного дизайну з централізованим управлінням станом через React Context API.

Реалізовано унікальний рекомендаційний модуль: адміністратор прив'язує перелік рекомендованого спорядження до кожного типу маршруту, а система автоматично формує персоналізовану підбірку товарів для покупця на основі обраного маршруту та його тривалості. Цей функціонал є ключовою конкурентною перевагою «PeakPoint» над аналізованими аналогами.

Проведено комплексне мануальне тестування клієнтської частини, що охопило 43 тест-кейси у п'яти функціональних блоках. Тестування виконувалось у браузерях Google Chrome, Mozilla Firefox та Apple Safari на трьох контрольних розширеннях екрану. Усі тест-кейси отримали статус «Пройдено», що підтверджує відповідність системи вимогам технічного завдання.

Складено інструкцію з розгортання системи на сервері Ubuntu 24.04 LTS із використанням systemd та Nginx, інструкцію з наповнення та адміністрування контенту, а також рекомендації щодо пошукової оптимізації, просування у соціальних мережах, моніторингу та резервного копіювання даних.

Практичне значення виконаної роботи полягає у створенні готового до розгортання та експлуатації вебзастосунку «PeakPoint». Система автоматизує роботу контент-менеджерів через адміністративну панель, забезпечує клієнтам зручний процес вибору та придбання товарів, знижує операційні витрати бізнесу та формує додаткову цінність для покупця через рекомендаційний інструментарій.

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		76

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Gorgany: магазин туристичного спорядження, одягу і взуття : вебсайт. URL: <https://gorgany.com> (дата звернення: 15.05.2026).
- 2) Інтернет-магазин ALP: спорядження для туризму та кемпінгу : вебсайт. URL: <https://alp.com.ua> (дата звернення: 15.05.2026).
- 3) Tramp: фірмовий магазин виробника туристичного спорядження в Україні : вебсайт. URL: <https://tramp.in.ua> (дата звернення: 15.05.2026).
- 4) Перевал: магазин туристичного спорядження : вебсайт. URL: <https://pereval.org.ua> (дата звернення: 15.05.2026).
- 5) Методичні вказівки до виконання кваліфікаційної роботи для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки» / уклад.: Марціяш Г. Я., Слободян Р. О. Тернопіль : ВСП «ТФК ТНТУ ім. І. Пулюя», 2026. 48 с.
- 6) React Documentation. React Team, 2024. URL: <https://react.dev> (дата звернення: 10.06.2026).
- 7) React Router Documentation. Remix Software, 2024. URL: <https://reactrouter.com/en/main> (дата звернення: 10.06.2026).
- 8) Axios Documentation. Matt Zabriskie, 2024. URL: <https://axios-http.com/docs/intro> (дата звернення: 10.06.2026).
- 9) Spring Data JPA Reference Documentation. VMware, Inc., 2024. URL: <https://docs.spring.io/spring-data/jpa/reference/index.html> (дата звернення: 10.06.2026).
- 10) Spring Security Reference Documentation. VMware, Inc., 2024. URL: <https://docs.spring.io/spring-security/reference/index.html> (дата звернення: 10.06.2026).
- 11) Закон України «Про охорону праці» від 14 жовтня 1992 р. № 2694-ХІІ. Відомості Верховної Ради України. 1992. № 49. Ст. 668. Редакція від 12.09.2025. URL: <https://zakon.rada.gov.ua/go/2694-12> (дата звернення: 14.06.2026).
- 12) Конституція України від 28 червня 1996 р. № 254к/96-ВР. Відомості

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		77

Верховної Ради України. 1996. № 30. Ст. 141. URL: <https://zakon.rada.gov.ua/laws/show/254к/96-вр> (дата звернення: 14.06.2026).

13) Закон України «Про пожежну безпеку» від 17 грудня 1993 р. № 3745-ХІІ. Відомості Верховної Ради України. 1994. № 5. Ст. 21. URL: <https://zakon.rada.gov.ua/laws/show/3745-12> (дата звернення: 14.06.2026).

14) Правила пожежної безпеки в Україні: НАПБ А.01.001-2014, затверджено наказом Міністерства внутрішніх справ України від 30 грудня 2014 р. № 1417. Зареєстровано в Міністерстві юстиції України 05 березня 2015 р. за № 252/26697. URL: <https://zakon.rada.gov.ua/go/z0252-15> (дата звернення: 14.06.2026).

15) Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою: НАПБ Б.03.002-2007, затверджено наказом Міністерства України з питань надзвичайних ситуацій від 03 грудня 2007 р. № 833. URL: <https://zakon.rada.gov.ua/laws/show/z1410-07> (дата звернення: 14.06.2026).

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		78

ДОДАТКИ

Додаток А. Лістинг файлу «RouteGuards.jsx»

```
import { Navigate, Outlet, useLocation } from 'react-router-dom';
import { useAuth } from '../hooks/useAuth';
import { hasRole, ROUTES } from '../utils/constants';

function SessionSpinner() {
  return (
    <div className="flex min-h-screen items-center justify-center bg-
[#faf7ef]">
      <div className="flex flex-col items-center gap-4">
        <div className="h-10 w-10 animate-spin rounded-full border-4
border-[#e3dcccb] border-t-[#244233]" />
        <p className="font-mono text-xs uppercase tracking-widest text-
[#717a70]">
          Завантаження...
        </p>
      </div>
    </div>
  );
}

export function ProtectedRoute() {
  const { isAuthenticated, isLoading } = useAuth();
  const location = useLocation();

  if (isLoading) return <SessionSpinner />;

  if (!isAuthenticated) {
    return (
      <Navigate
        to={ROUTES.LOGIN}

```

					2026.КВР.122.421.07.00.00 ПЗ	Арк.
						79
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        state={{ from: location }}
        replace
      />
    );
  }

  return <Outlet />;
}

export function RoleProtectedRoute({ minRole }) {
  const { isAuthenticated, isLoading, role } = useAuth();
  const location = useLocation();

  if (isLoading) return <SessionSpinner />;

  if (!isAuthenticated) {
    return <Navigate to={ROUTES.LOGIN} state={{ from: location }} replace
/>;
  }

  if (!hasRole(role, minRole)) {
    // Authenticated but insufficient privilege
    return <Navigate to={ROUTES.HOME} replace />;
  }

  return <Outlet />;
}

export function PublicOnlyRoute() {
  const { isAuthenticated, isLoading, role } = useAuth();

  if (isLoading) return <SessionSpinner />;

  if (isAuthenticated) {

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		80

```
const dest =
  role === 'ROLE_ADMIN' || role === 'ROLE_MANAGER'
    ? ROUTES.ADMIN_PRODUCTS
    : ROUTES.PROFILE;
return <Navigate to={dest} replace />;
}

return <Outlet />;
}
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

Додаток Б. Лістинг файлу «api.js»

```
import axios from 'axios';
import { API_BASE_URL, STORAGE_KEYS } from '../utils/constants';

const api = axios.create({
  baseURL: API_BASE_URL,
  timeout: 15_000,
  headers: {
    'Content-Type': 'application/json',
    'Accept': 'application/json',
  },
});

export const setAuthToken = (token) => {
  if (token) {
    localStorage.setItem(STORAGE_KEYS.AUTH_TOKEN, token);
    api.defaults.headers.common['Authorization'] = `Bearer ${token}`;
  }
};

export const clearAuthToken = () => {
  localStorage.removeItem(STORAGE_KEYS.AUTH_TOKEN);
  delete api.defaults.headers.common['Authorization'];
};

// Seed from storage on module load (page refresh persistence)
const storedToken = localStorage.getItem(STORAGE_KEYS.AUTH_TOKEN);
if (storedToken) setAuthToken(storedToken);

api.interceptors.request.use(
  (config) => {
    // Always read from storage so token updates are reflected immediately
    const token = localStorage.getItem(STORAGE_KEYS.AUTH_TOKEN);
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82

```

    if (token) {
        config.headers['Authorization'] = `Bearer ${token}`;
    }
    return config;
},
(error) => Promise.reject(error),
);

let isRefreshing = false;
let failedQueue = []; // queued requests while refreshing

const processQueue = (error, token = null) => {
    failedQueue.forEach(({ resolve, reject }) =>
        error ? reject(error) : resolve(token),
    );
    failedQueue = [];
};

api.interceptors.response.use(
    (response) => response,
    async (error) => {
        const originalRequest = error.config;

        if (error.response?.status === 401 && !originalRequest._retried) {
            originalRequest._retried = true;

            if (isRefreshing) {
                // Queue this request until refresh resolves
                return new Promise((resolve, reject) => {
                    failedQueue.push({ resolve, reject });
                })
                    .then((token) => {
                        originalRequest.headers['Authorization'] = `Bearer ${token}`;
                        return api(originalRequest);
                    });
            }
        }
    }
);

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

```

    })
    .catch((err) => Promise.reject(err));
}

isRefreshing = true;
const refreshToken =
localStorage.getItem(STORAGE_KEYS.REFRESH_TOKEN);

if (!refreshToken) {
    // No refresh token – force logout
    isRefreshing = false;
    window.dispatchEvent(new CustomEvent('auth:unauthorized'));
    return Promise.reject(error);
}

try {
    const { data } = await axios.post(
        `${API_BASE_URL}/auth/refresh`,
        { refreshToken },
        { headers: { 'Content-Type': 'application/json' } },
    );

    const newToken = data.accessToken;
    setAuthToken(newToken);
    if (data.refreshToken) {
        localStorage.setItem(STORAGE_KEYS.REFRESH_TOKEN,
data.refreshToken);
    }

    processQueue(null, newToken);
    originalRequest.headers['Authorization'] = `Bearer ${newToken}`;
    return api(originalRequest);
} catch (refreshError) {
    processQueue(refreshError, null);
}

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

```
clearAuthToken();
localStorage.removeItem(STORAGE_KEYS.REFRESH_TOKEN);
window.dispatchEvent(new CustomEvent('auth:unauthorized'));
return Promise.reject(refreshError);
} finally {
  isRefreshing = false;
}
}

if (error.response?.status === 403) {
  window.dispatchEvent(new CustomEvent('auth:forbidden'));
}

return Promise.reject(error);
},
);

export default api;
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

Додаток В. Лістинг файлу «AuthContext.jsx»

```
import {
  createContext,
  useCallback,
  useContext,
  useEffect,
  useMemo,
  useReducer,
} from 'react';
import { useNavigate } from 'react-router-dom';

import {
  clearAuthToken,
  setAuthToken,
} from '../services/api';
import {
  getMeRequest,
  loginRequest,
  logoutRequest,
} from '../services/authService';
import { ROUTES, STORAGE_KEYS } from '../utils/constants';

const INITIAL_STATE = {
  user:          null,
  isAuthenticated: false,
  isLoading:     true,    // true on first render while we check stored
token
  error:         null,
};

const AUTH_ACTIONS = /** @type {const} */ ({
  SET_LOADING: 'SET_LOADING',
  SET_USER:   'SET_USER',
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		86

```

SET_ERROR:    'SET_ERROR',
LOGOUT:       'LOGOUT',
});

function authReducer(state, action) {
  switch (action.type) {
    case AUTH_ACTIONS.SET_LOADING:
      return { ...state, isLoading: action.payload };

    case AUTH_ACTIONS.SET_USER:
      return {
        ...state,
        user:          action.payload,
        isAuthenticated: Boolean(action.payload),
        isLoading:    false,
        error:         null,
      };

    case AUTH_ACTIONS.SET_ERROR:
      return { ...state, error: action.payload, isLoading: false };

    case AUTH_ACTIONS.LOGOUT:
      return { ...INITIAL_STATE, isLoading: false };

    default:
      return state;
  }
}

const AuthContext = createContext(null);

export function AuthProvider({ children }) {
  const [state, dispatch] = useReducer(authReducer, INITIAL_STATE);
  const navigate = useNavigate();

```

					<i>2026.KBP.122.421.07.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		87

```

useEffect(() => {
  const token = localStorage.getItem(STORAGE_KEYS.AUTH_TOKEN);

  if (!token) {
    dispatch({ type: AUTH_ACTIONS.SET_LOADING, payload: false });
    return;
  }

  setAuthToken(token);
  getMeRequest()
    .then((user) => dispatch({ type: AUTH_ACTIONS.SET_USER, payload: user
}))
    .catch(() => {
      // Token expired or invalid – clear silently
      clearAuthToken();
      localStorage.removeItem(STORAGE_KEYS.REFRESH_TOKEN);
      dispatch({ type: AUTH_ACTIONS.LOGOUT });
    });
}, []);

useEffect(() => {
  const handleUnauthorized = () => {
    dispatch({ type: AUTH_ACTIONS.LOGOUT });
    navigate(ROUTES.LOGIN, { replace: true });
  };

  window.addEventListener('auth:unauthorized', handleUnauthorized);
  return () => window.removeEventListener('auth:unauthorized',
handleUnauthorized);
}, [navigate]);

const login = useCallback(async (email, password) => {
  dispatch({ type: AUTH_ACTIONS.SET_LOADING, payload: true });

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		88

```

dispatch({ type: AUTH_ACTIONS.SET_ERROR,   payload: null });

try {
  const { accessToken, refreshToken, user } = await loginRequest(email,
password);

  setAuthToken(accessToken);
  if (refreshToken) {
    localStorage.setItem(STORAGE_KEYS.REFRESH_TOKEN, refreshToken);
  }

  dispatch({ type: AUTH_ACTIONS.SET_USER, payload: user });

  // Redirect based on role
  if (user.role === 'ROLE_ADMIN' || user.role === 'ROLE_MANAGER') {
    navigate(ROUTES.ADMIN_PRODUCTS, { replace: true });
  } else {
    navigate(ROUTES.PROFILE, { replace: true });
  }
} catch (err) {
  const message =
    err.response?.data?.message ??
    'Невірний email або пароль. Спробуйте ще раз.';
  dispatch({ type: AUTH_ACTIONS.SET_ERROR, payload: message });
  throw err; // re-throw so form can catch
}
}, [navigate]);

const updateUser = useCallback((user) => {
  dispatch({ type: AUTH_ACTIONS.SET_USER, payload: user });
}, []);

const logout = useCallback(async () => {
  const refreshToken = localStorage.getItem(STORAGE_KEYS.REFRESH_TOKEN);

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		89

```

    try {
      await logoutRequest(refreshToken);
    } catch {
    } finally {
      clearAuthToken();
      localStorage.removeItem(STORAGE_KEYS.REFRESH_TOKEN);
      dispatch({ type: AUTH_ACTIONS.LOGOUT });
      navigate(ROUTES.HOME, { replace: true });
    }
  }, [navigate]);

const value = useMemo(
  () => ({
    user: state.user,
    isAuthenticated: state.isAuthenticated,
    isLoading: state.isLoading,
    role: state.user?.role ?? null,
    error: state.error,
    login,
    logout,
    updateUser,
  }),
  [state, login, logout, updateUser],
);
return <AuthContext.Provider
value={value}>{children}</AuthContext.Provider>;
}

export function useAuth() {
  const ctx = useContext(AuthContext);
  if (!ctx) throw new Error('useAuth must be used inside <AuthProvider>');
  return ctx;
}

export default AuthContext;

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		90

Додаток Г. Лістинг файлу «CartContext.jsx»

```
import {
  createContext,
  useCallback,
  useContext,
  useEffect,
  useMemo,
  useReducer,
  useState,
} from 'react';

import { STORAGE_KEYS } from '../utils/constants';

const loadFromStorage = () => {
  try {
    const raw = localStorage.getItem(STORAGE_KEYS.CART);
    return raw ? JSON.parse(raw) : [];
  } catch {
    return [];
  }
};

/** @param {import('../types').CartItem[]} items */
const persistToStorage = (items) => {
  try {
    localStorage.setItem(STORAGE_KEYS.CART, JSON.stringify(items));
  } catch {
    // storage full or private-mode denial – fail silently
  }
};

const CART_ACTIONS = /** @type {const} */ ({
  ADD: 'ADD',
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		91

```

REMOVE:      'REMOVE',
UPDATE_QTY:  'UPDATE_QTY',
CLEAR:       'CLEAR',
HYDRATE:     'HYDRATE',
});

/**
 * @param {import('../types').CartItem[]} state
 * @param {{ type: string, payload?: any }} action
 * @returns {import('../types').CartItem[]}
 */
function cartReducer(state, action) {
  switch (action.type) {
    case CART_ACTIONS.HYDRATE:
      return action.payload;

    case CART_ACTIONS.ADD: {
      const { id, qty = 1 } = action.payload;
      const existing = state.find((item) => item.id === id);
      if (existing) {
        return state.map((item) =>
          item.id === id ? { ...item, qty: item.qty + qty } : item,
        );
      }
      return [...state, { id, qty }];
    }

    case CART_ACTIONS.REMOVE:
      return state.filter((item) => item.id !== action.payload);

    case CART_ACTIONS.UPDATE_QTY: {
      const { id, qty } = action.payload;
      if (qty <= 0) return state.filter((item) => item.id !== id);
      return state.map((item) =>

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		92

```

        item.id === id ? { ...item, qty } : item,
    );
}

case CART_ACTIONS.CLEAR:
    return [];

default:
    return state;
}
}

export function CartProvider({ children, productPriceMap = {} }) {
    const [items, dispatch] = useReducer(cartReducer, [], loadFromStorage);
    const [promoCode, setPromoCode] = useState('');
    const [promoPercent, setPromoPercent] = useState(0);

    // Hydrate once on mount (handles SSR-style scenarios / hot reload)
    useEffect(() => {
        const stored = loadFromStorage();
        dispatch({ type: CART_ACTIONS.HYDRATE, payload: stored });
    }, []);

    // Persist every change
    useEffect(() => {
        persistToStorage(items);
    }, [items]);

    const removeItem = useCallback((productId) => {
        dispatch({ type: CART_ACTIONS.REMOVE, payload: productId });
    }, []);

    const updateQty = useCallback((productId, qty) => {

```

					<i>2026.KBP.122.421.07.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		93

```

    dispatch({ type: CART_ACTIONS.UPDATE_QTY, payload: { id: productId, qty
} });
}, []);

/** Clear the entire cart and promo. */
const clearCart = useCallback(() => {
    dispatch({ type: CART_ACTIONS.CLEAR });
    setPromoCode('');
    setPromoPercent(0);
}, []);

const applyPromo = useCallback((code, percent) => {
    setPromoCode(code);
    setPromoPercent(percent);
}, []);

const clearPromo = useCallback(() => {
    setPromoCode('');
    setPromoPercent(0);
}, []);

const totalCount = useMemo(
    () => items.reduce((sum, item) => sum + item.qty, 0),
    [items],
);

const totalPrice = useMemo(
    () =>
        items.reduce((sum, item) => {
            const price = productPriceMap[item.id] ?? 0;
            return sum + price * item.qty;
        }, 0),
    [items, productPriceMap],
);

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		94

```

/** @param {string} productId */
const isInCart = useCallback(
  (productId) => items.some((i) => i.id === productId),
  [items],
);

/** @param {string} productId */
const getQty = useCallback(
  (productId) => items.find((i) => i.id === productId)?.qty ?? 0,
  [items],
);

const value = useMemo(
  () => ({
    items,
    totalCount,
    totalPrice,
    addItem,
    removeItem,
    updateQty,
    clearCart,
    isInCart,
    getQty,
    promoCode,
    promoPercent,
    applyPromo,
    clearPromo,
  }),
  [items, totalCount, totalPrice, addItem, removeItem, updateQty,
clearCart,
  isInCart, getQty, promoCode, promoPercent, applyPromo, clearPromo],
);

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		95

```

return <CartContext.Provider
value={value}>{children}</CartContext.Provider>;
}

export function useCart() {
  const ctx = useContext(CartContext);
  if (!ctx) throw new Error('useCart must be used inside <CartProvider>');
  return ctx;
}

export default CartContext;

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		96

Додаток Д. Лістинг файлу «JwtAuthFilter.java»

```
package ua.peakpoint.security;

import jakarta.servlet.FilterChain;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import lombok.RequiredArgsConstructor;
import
org.springframework.security.authentication.UsernamePasswordAuthenticationT
oken;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import
org.springframework.security.web.authentication.WebAuthenticationDetailsSou
rce;
import org.springframework.stereotype.Component;
import org.springframework.web.filter.OncePerRequestFilter;

import java.io.IOException;

@Component
@RequiredArgsConstructor
public class JwtAuthFilter extends OncePerRequestFilter {

    private final JwtTokenProvider jwtTokenProvider;
    private final UserDetailsService userDetailsService;

    @Override
    protected void doFilterInternal(HttpServletRequest request,
                                    HttpServletResponse response,
                                    FilterChain filterChain) throws
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		97

```

ServletException, IOException {
    String authHeader = request.getHeader("Authorization");
    if (authHeader == null || !authHeader.startsWith("Bearer ")) {
        filterChain.doFilter(request, response);
        return;
    }

    String token = authHeader.substring(7);
    try {
        String email = jwtTokenProvider.extractUsername(token);
        if (email != null &&
SecurityContextHolder.getContext().getAuthentication() == null) {
            UserDetails userDetails =
userDetailsService.loadUserByUsername(email);
            if (jwtTokenProvider.isTokenValid(token, userDetails)) {
                UsernamePasswordAuthenticationToken authToken =
                new
UsernamePasswordAuthenticationToken(userDetails, null,
userDetails.getAuthorities());
                authToken.setDetails(new
WebAuthenticationDetailsSource().buildDetails(request));

SecurityContextHolder.getContext().setAuthentication(authToken);
            }
        }
    } catch (Exception ignored) {
        // невалідний токен – просто продовжуємо без аутентифікації
    }

    filterChain.doFilter(request, response);
}
}

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		98

Додаток Е. Лістинг файлу «ProductSpec.java»

```
package ua.peakpoint.entity;

import jakarta.persistence.*;
import lombok.*;

@Entity
@Table(name = "product_specs")
@Getter @Setter @NoArgsConstructor @AllArgsConstructor @Builder
public class ProductSpec {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "product_id", nullable = false)
    private Product product;

    @Column(name = "spec_key", nullable = false)
    private String key;

    @Column(name = "spec_value", nullable = false)
    private String value;

    @Column(nullable = false)
    @Builder.Default
    private Integer sortOrder = 0;
}
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		99

Додаток Ж. Лістинг файлу «OrderService.java»

```
package ua.peakpoint.service;

import lombok.RequiredArgsConstructor;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import ua.peakpoint.dto.request.PlaceOrderRequest;
import ua.peakpoint.dto.response.OrderItemResponse;
import ua.peakpoint.dto.response.OrderResponse;
import ua.peakpoint.dto.response.PlaceOrderResponse;
import ua.peakpoint.entity.Order;
import ua.peakpoint.entity.OrderItem;
import ua.peakpoint.entity.Product;
import ua.peakpoint.entity.PromoCode;
import ua.peakpoint.entity.User;
import ua.peakpoint.enums.DeliveryMethod;
import ua.peakpoint.enums.OrderStatus;
import ua.peakpoint.exception.ResourceNotFoundException;
import ua.peakpoint.repository.OrderRepository;
import ua.peakpoint.repository.ProductRepository;
import ua.peakpoint.repository.PromoCodeRepository;

import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
import java.util.stream.Collectors;

@Service
@RequiredArgsConstructor
@Transactional
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		100

```

public class OrderService {

    private final OrderRepository orderRepository;
    private final ProductRepository productRepository;
    private final PromoCodeRepository promoCodeRepository;
    private final EmailService emailService;

    @Value("${app.free-shipping-threshold}")
    private int freeShippingThreshold;

    @Value("${app.shipping-cost}")
    private int shippingCost;

    public PlaceOrderResponse placeOrder(PlaceOrderRequest request, User
user) {
        // Промокод
        int discountPercent = 0;
        if (request.promoCode() != null && !request.promoCode().isBlank())
        {
            PromoCode promo = promoCodeRepository
                .findByCodeIgnoreCaseAndIsActiveTrue(request.promoCode())
                .orElse(null);
            if (promo != null && (promo.getMaxUsage() == null ||
promo.getUsageCount() < promo.getMaxUsage())) {
                discountPercent = promo.getDiscountPercent();
                promo.setUsageCount(promo.getUsageCount() + 1);
                promoCodeRepository.save(promo);
            }
        }

        // Формуємо позиції та рахуємо суму
        int subtotal = 0;
        List<OrderItem> orderItems = new ArrayList<>();
    }
}

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		101

```

        for (var itemReq : request.items()) {
            Product product =
productRepository.findById(itemReq.productId())
                .orElseThrow(() -> new ResourceNotFoundException("Товар не
знайдений: " + itemReq.productId()));
            int lineTotal = product.getPrice() * itemReq.qty();
            subtotal += lineTotal;
            orderItems.add(OrderItem.builder()
                .product(product)
                .productName(product.getName())
                .productPrice(product.getPrice())
                .qty(itemReq.qty())
                .lineTotal(lineTotal)
                .build());
        }

        int discount = discountPercent > 0 ? Math.round(subtotal *
discountPercent / 100f) : 0;
        int shipping = (request.delivery() == DeliveryMethod.PICKUP ||
subtotal >= freeShippingThreshold)
            ? 0 : shippingCost;
        int total = subtotal - discount + shipping;

        String orderNo = "PP-" + UUID.randomUUID().toString().replace("-",
"" ).substring(0, 6).toUpperCase();

        Order order = Order.builder()
            .orderNo(orderNo)
            .user(user)
            .status(OrderStatus.PENDING)
            .firstName(request.firstName())
            .lastName(request.lastName())
            .phone(request.phone())
            .email(request.email())

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		102

```

        .delivery(request.delivery())
        .city(request.city())
        .address(request.address())
        .payment(request.payment())
        .subtotal(subtotal)
        .shippingCost(shipping)
        .discountAmount(discount)
        .total(total)
        .promoCode(request.promoCode())
        .build();

orderItems.forEach(item -> item.setOrder(order));
order.setItems(orderItems);

Order saved = orderRepository.save(order);
emailService.sendOrderConfirmation(saved);
return new PlaceOrderResponse(saved.getId(), saved.getOrderNo());
}

@Transactional(readOnly = true)
public List<OrderResponse> getMyOrders(User user) {
    return orderRepository.findByUserOrderByCreatedAtDesc(user)
        .stream().map(this::toResponse).collect(Collectors.toList());
}

@Transactional(readOnly = true)
public OrderResponse getOrderById(Long id, User user) {
    Order order = orderRepository.findById(id)
        .orElseThrow(() -> new ResourceNotFoundException("Замовлення не
знайдено: " + id));

    boolean isOwner = order.getUser() != null &&
order.getUser().getId().equals(user.getId());
    boolean isManager = user.getRole().name().equals("ROLE_MANAGER")

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		103

```

        || user.getRole().name().equals("ROLE_ADMIN");

        if (!isOwner && !isManager) {
            throw new ResourceNotFoundException("Замовлення не знайдено: "
+ id);
        }
        return toResponse(order);
    }

    @Transactional(readOnly = true)
    public Page<OrderResponse> getAllOrders(Pageable pageable, String
status) {
        if (status != null && !status.isBlank()) {
            OrderStatus s = OrderStatus.valueOf(status.toUpperCase());
            return orderRepository.findByStatusOrderByCreatedAtDesc(s,
pageable).map(this::toResponse);
        }
        return
orderRepository.findAllByOrderByCreatedAtDesc(pageable).map(this::toRespons
e);
    }

    public OrderResponse updateStatus(Long id, OrderStatus status) {
        Order order = orderRepository.findById(id)
            .orElseThrow(() -> new ResourceNotFoundException("Замовлення не
знайдено: " + id));
        order.setStatus(status);
        Order saved = orderRepository.save(order);
        emailService.sendOrderStatusUpdate(saved);
        return toResponse(saved);
    }

    public OrderResponse updateTracking(Long id, String trackingNo) {
        Order order = orderRepository.findById(id)
            .orElseThrow(() -> new ResourceNotFoundException("Замовлення не
знайдено: " + id));

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		104

```

        order.setTrackingNo(trackingNo);
        return toResponse(orderRepository.save(order));
    }

    private OrderResponse toResponse(Order o) {
        List<OrderItemResponse> items = o.getItems().stream()
            .map(i -> {
                String imageUrl = i.getProduct().getImages().isEmpty()
                    ? null
                    : i.getProduct().getImages().get(0).getUrl();
                return new OrderItemResponse(
                    i.getId(), i.getProduct().getId(),
                    i.getProductName(), imageUrl,
                    i.getProductPrice(), i.getQty(), i.getLineTotal()
                );
            })
            .collect(Collectors.toList());

        return new OrderResponse(
            o.getId(), o.getOrderNo(),
            o.getStatus().name().toLowerCase(),
            o.getFirstName(), o.getLastName(),
            o.getPhone(), o.getEmail(),
            o.getDelivery().name().toLowerCase(),
            o.getCity(), o.getAddress(),
            o.getPayment().name().toLowerCase(),
            o.getSubtotal(), o.getShippingCost(),
            o.getDiscountAmount(), o.getTotal(),
            o.getPromoCode(), o.getTrackingNo(),
            items,
            o.getCreatedAt() != null ? o.getCreatedAt().toString() : null
        );
    }
}

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		105

Додаток II. Лістинг файлу «GlobalExceptionHandler.java»

```
package ua.peakpoint.exception;
import jakarta.servlet.http.HttpServletRequest;
import org.springframework.http.ResponseEntity;
import org.springframework.security.access.AccessDeniedException;
import org.springframework.security.authentication.BadCredentialsException;
import org.springframework.web.bind.MethodArgumentNotValidException;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.RestControllerAdvice;
import ua.peakpoint.dto.response.ErrorResponse;
import java.time.LocalDateTime;
import java.util.stream.Collectors;

@RestControllerAdvice
public class GlobalExceptionHandler {

    @ExceptionHandler(ResourceNotFoundException.class)
    public ResponseEntity<ErrorResponse>
handleNotFound(ResourceNotFoundException ex, HttpServletRequest req) {
        return ResponseEntity.status(404).body(error(404, "Not Found",
ex.getMessage(), req));
    }

    @ExceptionHandler(BadRequestException.class)
    public ResponseEntity<ErrorResponse>
handleBadRequest(BadRequestException ex, HttpServletRequest req) {
        return ResponseEntity.status(400).body(error(400, "Bad Request",
ex.getMessage(), req));
    }

    @ExceptionHandler(BadCredentialsException.class)
    public ResponseEntity<ErrorResponse>
```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		106

```

handleBadCredentials(BadCredentialsException ex, HttpServletRequest req) {
    return ResponseEntity.status(401).body(error(401, "Unauthorized",
"Невірний email або пароль", req));
}

ExceptionHandler(MethodArgumentNotValidException.class)
public ResponseEntity<ErrorResponse>
handleValidation(MethodArgumentNotValidException ex, HttpServletRequest
req) {
    String message = ex.getBindingResult().getFieldErrors().stream()
        .map(e -> e.getField() + ": " + e.getDefaultMessage())
        .collect(Collectors.joining(", "));
    return ResponseEntity.status(400).body(error(400, "Validation
Error", message, req));
}

ExceptionHandler(AccessDeniedException.class)
public ResponseEntity<ErrorResponse>
handleForbidden(AccessDeniedException ex, HttpServletRequest req) {
    return ResponseEntity.status(403).body(error(403, "Forbidden",
"Доступ заборонено", req));
}

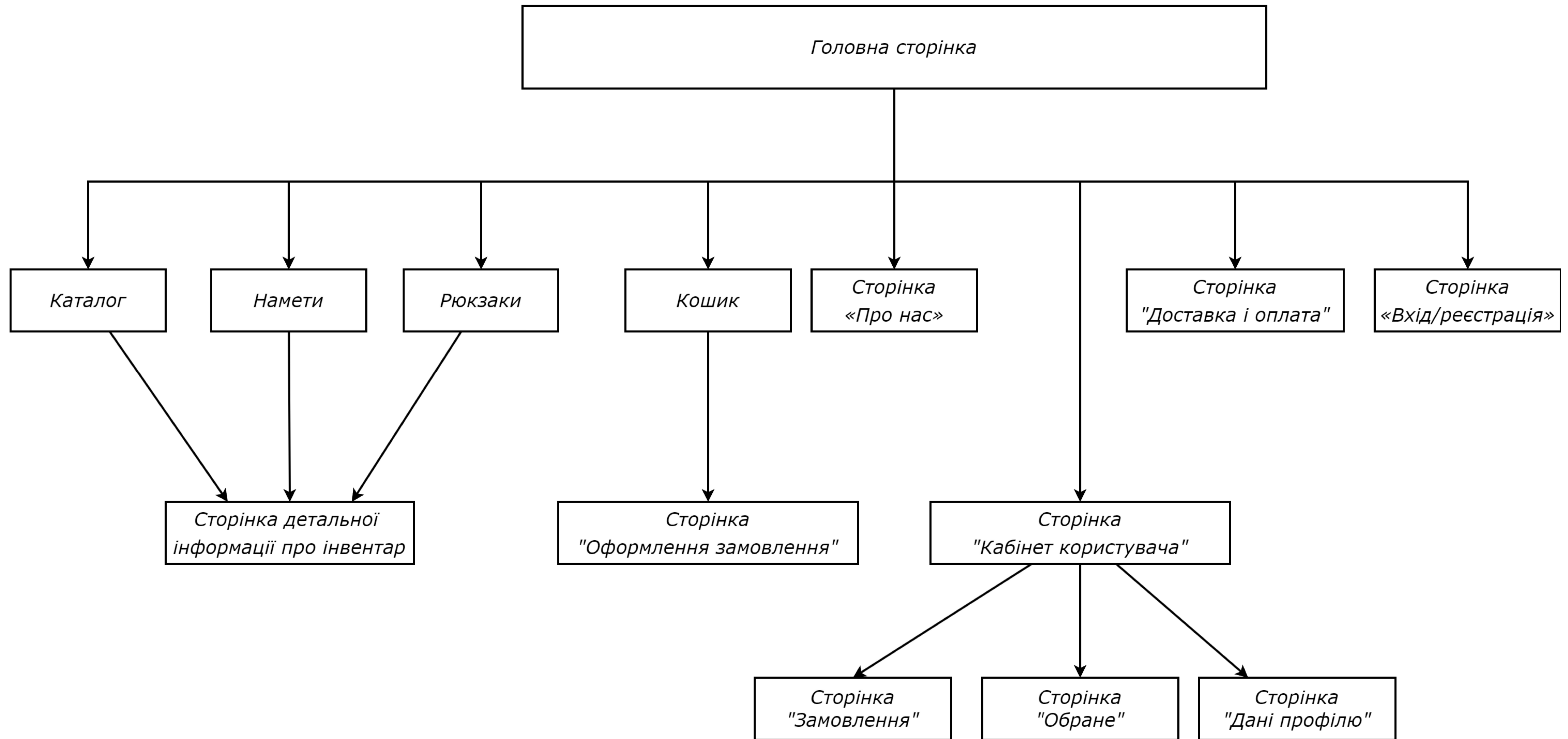
ExceptionHandler(Exception.class)
public ResponseEntity<ErrorResponse> handleGeneral(Exception ex,
HttpServletRequest req) {
    return ResponseEntity.status(500).body(error(500, "Internal Server
Error", "Внутрішня помилка сервера", req));
}

private ErrorResponse error(int status, String error, String message,
HttpServletRequest req) {
    return new ErrorResponse(LocalDate.now(), status, error,
message, req.getRequestURI());
}
}

```

					2026.KBP.122.421.07.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		107

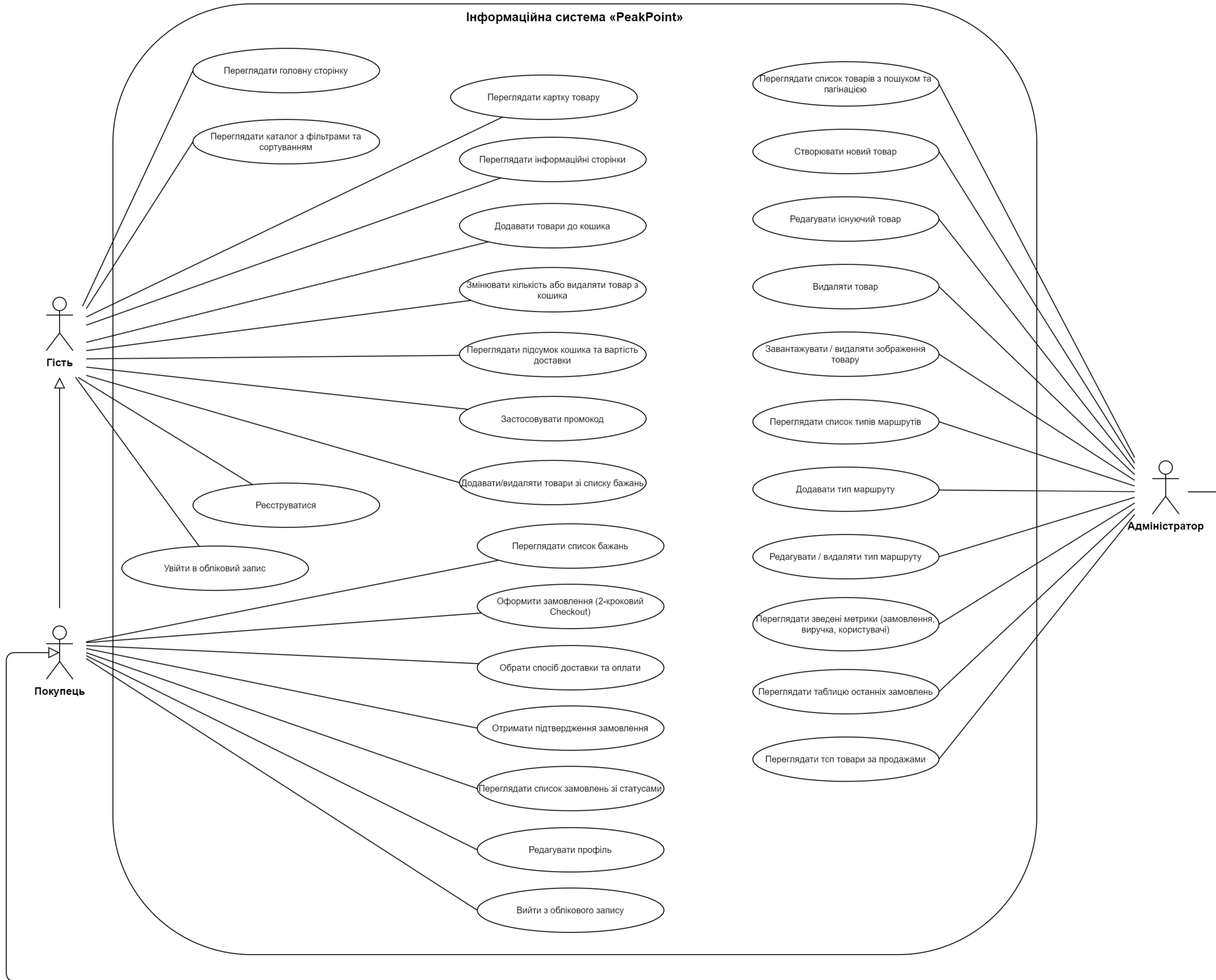
Схема структурна інформаційної системи



					2026.KBP.122.42107.00.00 CC			
Зм	Арк	№ докум	Підп	Дата	Розробка інформаційної системи електронної комерції для реалізації тободарів для кемпінгу «PeakPoint»	Лист	Маса	Масштаб
Розроб		Гуден О.А.			Схема структурної інформаційної системи			
Перев		Радчук Г.І.				Аркци	Архци	1
Т.контр						ВП ТФК ТНТУ КН-4.21		
Рецензент						м. Тернопіль		
Н.контр		Приймак В.А.						
Затв								

Перш. запис. Сторінка №. Підп. і дата. Зам. № і арк. №. Підп. і дата. Під № арк. №.

UML-діаграма варіантів використання

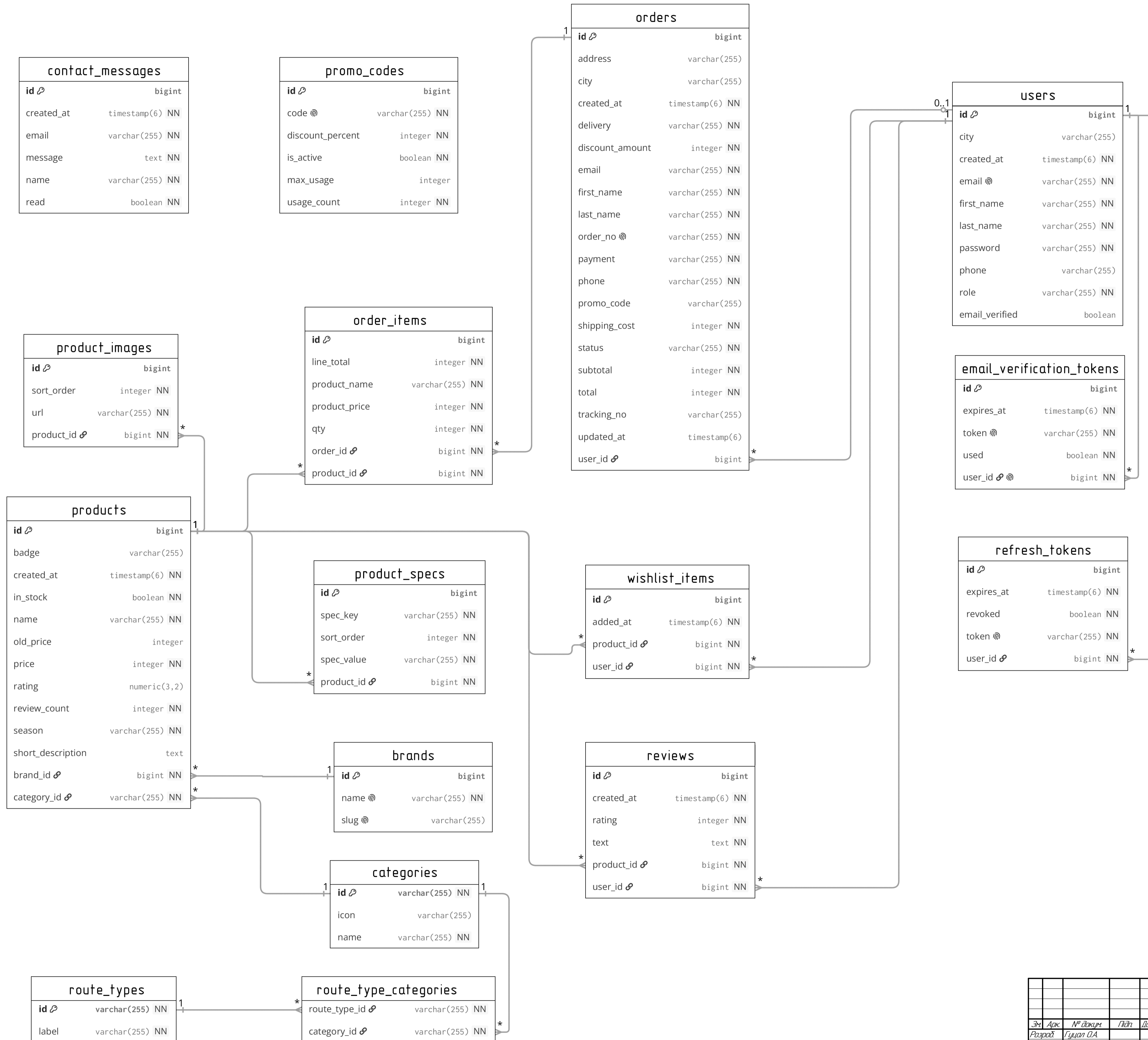


2026.KBP.122.4.2107.00.00 ДВ

Перш. запис.
Стор. №
Ліст. №
Зем. №
Ліст. №

					2026.KBP.122.4.2107.00.00 ДВ		
Зм.	Арк.	№ докум.	Підп.	Дата	Розробка інформаційної системи електронної комерції для реалізації товарів для кампанії «PeakPoint» UML-діаграма варіантів використання		
Розроб.	Гуден О.А.				Лист	Маса	Масштаб
Перев.	Радчук Г.І.				Архіви	Архіви	1
Текст.					ВП ТФК ТНТУ КН-4.21		
Рецензент					м. Тернопіль		
Начальник	Примак В.А.						
Затв.					Копія		
					Формат А1		

ER-діаграма бази даних інформаційної системи



Лист № 1
Лист № 2
Лист № 3
Лист № 4
Лист № 5
Лист № 6
Лист № 7
Лист № 8
Лист № 9
Лист № 10
Лист № 11
Лист № 12
Лист № 13
Лист № 14
Лист № 15
Лист № 16
Лист № 17
Лист № 18
Лист № 19
Лист № 20
Лист № 21
Лист № 22
Лист № 23
Лист № 24
Лист № 25
Лист № 26
Лист № 27
Лист № 28
Лист № 29
Лист № 30
Лист № 31
Лист № 32
Лист № 33
Лист № 34
Лист № 35
Лист № 36
Лист № 37
Лист № 38
Лист № 39
Лист № 40
Лист № 41
Лист № 42
Лист № 43
Лист № 44
Лист № 45
Лист № 46
Лист № 47
Лист № 48
Лист № 49
Лист № 50
Лист № 51
Лист № 52
Лист № 53
Лист № 54
Лист № 55
Лист № 56
Лист № 57
Лист № 58
Лист № 59
Лист № 60
Лист № 61
Лист № 62
Лист № 63
Лист № 64
Лист № 65
Лист № 66
Лист № 67
Лист № 68
Лист № 69
Лист № 70
Лист № 71
Лист № 72
Лист № 73
Лист № 74
Лист № 75
Лист № 76
Лист № 77
Лист № 78
Лист № 79
Лист № 80
Лист № 81
Лист № 82
Лист № 83
Лист № 84
Лист № 85
Лист № 86
Лист № 87
Лист № 88
Лист № 89
Лист № 90
Лист № 91
Лист № 92
Лист № 93
Лист № 94
Лист № 95
Лист № 96
Лист № 97
Лист № 98
Лист № 99
Лист № 100

2026.KBP.122.4.21.07.00.00 БД					Лист	Маса	Масштаб
Змі	Арх	№ докум	Підп	Дата	Розробка інформаційної системи електронної комерції для реалізації мобард для кемпінгу «РезортПіт»		
Розроб	Гуцалю О.А.	Радчук Г.Л.			ER-діаграма бази даних		
Перев					Архив	Архив	1
Т.контр					ВП ТФК ТНТУ КН-421		
Рецензент					м. Тернопіль		
Н.контр	Прішмак В.А.						
Затв							
Копія					Формат А1		

Таблиця техніко-економічних показників

№ п/п	Показник	Одиниці вимірювання	Значення
1	Мови програмування	–	Java, JavaScript
2	Технології клієнтської частини	–	React 18.3, Vite 5.4, TailwindCSS 3.4, Axios 1.7
3	Технології серверної частини	–	Spring Boot 3.3.4, Spring Security 6
4	База даних	–	PostgreSQL 15
5	Середовище розробки	–	IntelliJ IDEA 2026.1.3, WebStorm 2025.4
6	Захист	–	JWT, Spring Security
7	Содівартість	грн.	51 885,67
8	Плановий прибуток	грн.	29 055,98
9	Ціна	грн.	80 941,65
10	Чиста теперішня вартість	грн.	20 372,24
11	Термін окупності	рік	2,1

				2026.КВР.122.42107.00.00 ТБ				
Зм.	Арк.	№ докум.	Підп.	Дата	Розробка інформаційної системи електронної комерції для реалізації товарів для кемпінгу «ReactPoint»	Лист	Маса	Масштаб
Розроб.		Гуден О.А.			Таблиця техніко-економічних показників			
Перев.		Радчук Г.І.				Аркци	Аркци	1
Т.контр.						ВСП ТФК ТНТУ м. Тернопіль КН-4.21		
Рецензент								
Н.контр.		Примак В.А.						
Затв.								