

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 вебсайту зоомагазину «Лапка»**

Виконав: студент IV курсу, групи КН-421
спеціальності: 122 Комп'ютерні науки

Владислав ЧОКАН

Керівник Галина МАРЦІЯШ

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 МАРЦЯШ

« 02 » березня 2026 року

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

_____ Чокану Владиславу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебсайту зоомагазину "Лапка"

керівник роботи _____ Марціяш Галина Ярославівна _____,
(прізвище, ім'я, по батькові)

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

2. Строк подання студенткою роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мови програмування: JavaScript; середовище виконання Node.js; система керування базами даних MongoDB Atlas; хмарний сервіс Cloudinary; фреймворки: Express.js; бібліотека React, Vite 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 вебзастосування
 - 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 сайту і вебсторінок

2.2 Розробка структури бази даних сайту

2.3 Створення та верстка сторінок сайту

2.4 Програмування сайту

2.4.1 Написання клієнтської частини

2.4.2 Написання серверної частини

3 Спеціальний розділ

3.1 Інструкція з розміщення сайту в Інтернеті

3.2 Інструкція з обслуговування та наповнення сайту

3.3 Інструкція з популяризації та підтримки сайту

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 вебсайту зоомагазину

«ЛАПКА»

4.5 Обчислення накладних витрат

4.6 Складання кошторису витрат та визначення собівартості вебсайту

зоомагазину«ЛАПКА»

4.7 Розрахунок ціни вебсайту зоомагазину «ЛАПКА»

4.8 Визначення економічної ефективності і терміну окупності капітальних

вкладень

5 Охорона праці, техніка безпеки та екологічні вимоги

5.1 Інструктажі з охорони праці, їх поділ

5.2 Системи засобів і заходів забезпечення електробезпеки

6 Висновки

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – вебсайту.

5. Перелік графічного матеріалу:

1. UML-діаграма варіантів використання програми
2. Структурна схема головної сторінки сайту
3. ER-діаграма бази даних вебсайту
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	Геннадій 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 проєкту	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.

Студент

_____ (підпис)

Владислав ЧОКАН

Керівник роботи

_____ (підпис)

Галина МАРЦЯШ

ЗМІСТ

АНОТАЦІЯ	7
ANNOTATION	8
ВСТУП	9
1 ЗАГАЛЬНИЙ РОЗДІЛ	9
1.1 Аналітичний огляд існуючих рішень	10
1.2 Технічне завдання.....	12
1.2.1 Найменування та область застосування.....	12
1.2.2 Призначення розробки	13
1.2.3 Вимоги до функціоналу web-сайту.....	14
1.2.4 Вимоги до програмної документації	15
1.2.5 Техніко-економічні показники	16
1.2.6 Стадії та етапи розробки	17
1.2.7 Порядок тестування та прийому	19
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ	21
2.1 Розробка структури сайту і web-сторінок	21
2.2 Створення та верстка сторінок сайту	23
2.3 Розробка структури бази даних сайту	26
2.4 Програмування сайту	29
2.4.1 Написання клієнтської частини.....	32
2.4.2 Написання admin частини	33
2.5 Тестування сайту.....	34
3 СПЕЦІАЛЬНИЙ РОЗДІЛ	42
3.1 Інструкція з розміщення сайту в Інтернеті.....	42
3.2 Інструкція з обслуговування та наповнення сайту.....	44
3.3 Інструкція з популяризації та підтримки сайту	46

					2026.КВР.122.421.21.00.00 ПЗ			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розробка вебсайту зоомагазину “ЛАПКА” Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Розроб.</i>		<i>Чокан.В.В.</i>					5	
<i>Перевір.</i>		<i>Марціяш Г. Я.</i>						
<i>Реценз.</i>								
<i>Н. Контр.</i>		<i>Приймак В. А.</i>						
<i>Затверд.</i>								ВСП ТФК ТНТУ КН-421 м. Тернопіль

4. ЕКОНОМІЧНИЙ РОЗДІЛ	49
4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР	49
4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи	50
4.3. Розрахунок витрат на електроенергію	52
4.4. Розрахунок суми амортизаційних відрахувань вебсайту зоомагазину «ЛАПКА»	53
4.5. Обчислення накладних витрат	54
4.6. Складання кошторису витрат та визначення собівартості вебсайту зоомагазину«ЛАПКА»	54
4.7. Розрахунок ціни вебсайту зоомагазину «ЛАПКА»	55
4.8. Визначення економічної ефективності і терміну окупності капітальних вкладень	55
5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ	5.1
Інструкції з охорони праці, їх поділ.....	57
5.2 Системи засобів і заходів забезпечення електробезпеки	58
ВИСНОВКИ.....	60
ПЕРЕЛІК ПОСИЛАНЬ	61
ДОДАТОК А. Лістинг файлу «App.jsx»	63
ДОДАТОК Б. Лістинг файлу «main.jsx».....	64
ДОДАТОК В. Лістинг файлу «HomePage.jsx»	65
ДОДАТОК Г. Лістинг файлу «CatalogPage.jsx»	69
ДОДАТОК Д. Лістинг файлу «ProductDetailsPage .jsx».....	77
ДОДАТОК Е. Лістинг файлу «CartPage.jsx»	82
ДОДАТОК К. Лістинг файлу «server.js»	99
ДОДАТОК И. Лістинг файлу «admin.js».....	107

АНОТАЦІЯ

Тема кваліфікаційної роботи: Розробка вебсайту зоомагазину «ЛАПКА».

Метою даного проекту є розробка зручного та функціонального вебсайту, призначеного для перегляду, пошуку та онлайн-замовлення товарів для домашніх тварин.

Пояснювальна записка складається з п'яти розділів.

У загальній частині описуються аналітичний огляд існуючих рішень та аналіз технічного завдання.

У другому розділі технічного завдання представлено процес створення програмного продукту, опис та обґрунтування вибору мов програмування і технологій розробки вебсайту, опис бази даних проекту, зовнішнє проєктування програми, тестування та налагодження системи.

У спеціальній частині описані процес інсталяції програмного продукту, інструкція з використання тестових даних та інструкція користувача вебсайту зоомагазину «ЛАПКА»

Розрахунок вартості розробки та економічної ефективності вебсайту наведено в економічній частині, що знаходиться у четвертому розділі.

Основні питання охорони праці та техніки безпеки розглянуто у п'ятому розділі.

Обсяг пояснювальної записки становить 109 сторінок. До складу кваліфікаційної роботи входить графічна частина, яка складається зі структурної схеми програми, ER-діаграми бази даних, техніко-економічних показників, UML-діаграми використання вебсайту, виконаних на окремих аркушах формату А1.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

ANNOTATION

Qualification work topic: Development of the "LAPKA" Pet Shop Website.

The purpose of this project is to develop a convenient and functional website intended for browsing, searching, and online ordering of pet products.

The explanatory note consists of five chapters.

The general chapter contains an analytical review of existing solutions and an analysis of the technical specification.

The second chapter describes the software development process, provides a justification for the selected programming languages and development technologies, presents the project database structure, external software design, testing, and debugging procedures.

The special chapter describes the software installation process, instructions for using test data, and the user manual for the "LAPKA" pet shop website.

The calculation of development costs and the economic efficiency of the website is presented in the economic chapter, which is the fourth chapter of the qualification work.

The main issues of occupational safety and labor protection are considered in the fifth chapter.

The explanatory note consists of 109 pages and has been prepared in accordance with the methodological guidelines [1].

The qualification work also includes a graphical part consisting of a program structure diagram, an ER database diagram, technical and economic indicators, and a UML use case diagram of the website, presented on separate A1 format sheets.

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

У сучасному суспільстві сфера електронної комерції активно розвивається, тому бізнеси потребують сучасних цифрових інструментів для роботи з клієнтами. Для ефективної роботи вебсайту зоомагазину «ЛАПКА» необхідна надійна онлайн-платформа, яка надає можливість зручно переглядати товари, швидко оформлювати замовлення та керувати контентом сайту. Традиційні способи продажу мають певні недоліки, серед яких обмежений доступ до інформації, складність обробки замовлень і недостатня автоматизація основних процесів, що може негативно впливати на ефективність роботи магазину та кількість клієнтів.

Сучасні вебтехнології відіграють важливу роль у розвитку онлайн-бізнесу та покращенні обслуговування користувачів. Добре спроектований вебсайт із продуманим UX/UI-дизайном допомагає централізовано керувати товарами, швидко отримувати доступ до каталогу продукції та зручно користуватися функціями сайту. Завдяки цьому покупці можуть легко знаходити потрібні товари, використовувати пошук і фільтрацію, оформлювати замовлення та переглядати інформацію про продукцію.

Метою даної кваліфікаційної роботи є розробка структури, дизайну та функціоналу вебсайту зоомагазину «ЛАПКА» для автоматизації процесу продажу товарів і покращення взаємодії з користувачами. У вебсайті передбачається реалізувати адаптивний інтерфейс, систему керування товарами, кошик покупок, авторизацію користувачів та адміністративну панель.

Використання такого рішення допоможе оптимізувати процес продажу товарів, покращити обслуговування клієнтів і спростити адміністрування інтернет-магазину. Крім того, вебсайт забезпечить швидкий доступ до інформації про продукцію, підвищить зручність оформлення замовлень та сприятиме розширенню клієнтської бази. Реалізація даного проєкту дозволить підвищити конкурентоспроможність зоомагазину «ЛАПКА», автоматизувати основні бізнес-процеси та покращити взаємодію між продавцем і покупцем.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

У сфері електронної комерції сучасний ринок програмного забезпечення пропонує велику кількість рішень для організації онлайн-продажів, управління товарами та взаємодії з клієнтами. Основні підходи до створення інтернет-магазинів з урахуванням потреб користувачів включають готові SaaS-платформи, універсальні системи управління контентом (CMS), індивідуальну розробку на замовлення та створення повнофункціональних вебзастосунків на базі сучасного Full-Stack стеку технологій. Кожен із цих підходів має свої переваги та недоліки, які необхідно враховувати для забезпечення зручності використання, адаптивності, швидкодії та оптимізації витрат.

Готові SaaS-платформи для створення вебсайтів електронної комерції, такі як Shopify або BigCommerce, є популярними рішеннями для організації онлайн-продажів. Вони надають користувачам готовий функціонал для керування товарами, приймання замовлень, інтеграції платіжних систем та адміністрування магазину. Під час аналізу вимог визначали відповідність структури майбутнього вебсайту технічному завданню та перелік основних сценаріїв роботи користувача із системою.

Проте подібні системи часто мають обмеження щодо гнучкого налаштування дизайну та внутрішньої логіки роботи вебсайту. Крім цього, більшість сервісів працює за моделлю щомісячної підписки, що створює додаткові фінансові витрати для власника магазину.

Універсальні системи управління контентом, такі як WordPress у поєднанні з WooCommerce або OpenCart, дозволяють більш детально контролювати структуру вебресурсу та реалізовувати додатковий функціонал. Такі рішення є проміжним варіантом між готовими платформами та повністю індивідуальною розробкою. Вони забезпечують можливість створення каталогу товарів, системи фільтрації, пошуку, кошика покупок та адміністрування замовлень. Проте

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

використання великої кількості сторонніх модулів може негативно впливати на продуктивність системи, безпеку та складність підтримки вебсайту.

Індивідуальна розробка вебсайту на базі сучасних технологій Full-Stack розробки дозволяє реалізувати систему, максимально адаптовану до потреб конкретного бізнесу. У межах даного проєкту для створення клієнтської частини буде використовуватися React, серверна частина реалізується за допомогою Node.js та Express, а для зберігання даних використовується MongoDB. Такий підхід допомагає підвищити продуктивність системи, спрощує налаштування функціоналу та підтримує інтеграцію сучасних сервісів, зокрема Cloudinary для роботи із зображеннями товарів [13].

У результаті аналізу було встановлено, що найбільш доцільним рішенням для реалізації кваліфікаційної роботи є створення власного вебсайту зоомагазину «ЛАПКА». Такий підхід дає можливість реалізувати необхідний функціонал, створити адаптивний інтерфейс, організувати зручну роботу з товарами та забезпечити ефективну взаємодію користувачів із системою.

Для керування товарами, категоріями та замовленнями у вебсайту буде передбачатись адміністративну панель. Через неї адміністратор може додавати нові товари, редагувати інформацію про продукцію, завантажувати фотографії та переглядати оформлені замовлення користувачів.

Індивідуальні рішення на замовлення, створені спеціалізованими вебстудіями, дозволяють максимально адаптувати вебсайт до потреб конкретного бізнесу. Такі системи будуються з урахуванням індивідуальних вимог до UX/UI-дизайну, структури навігації, інтеграції платіжних сервісів та систем керування товарами. Однак основним недоліком такого підходу є висока вартість розробки, тестування та подальшої підтримки проєкту. Будь-які зміни або масштабування системи потребують додаткових фінансових витрат і залучення технічних спеціалістів, що ускладнює подальший розвиток вебсайту.

Сучасний Full-Stack стек технологій є найбільш доцільним підходом для розробки вебсайту зоомагазину «ЛАПКА». У межах цього проєкту він забезпечує

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

максимальну гнучкість архітектури, повний контроль над зберіганням даних та незалежність від сторонніх платформ і платних ліцензій. Реалізація компонентного підходу на фронтенді з використанням React та Vite, а також розробка власного REST API на Node.js і Express дозволяють створити швидкий, адаптивний та оптимізований вебсайт.

Основними перевагами використання цього підходу є можливість реалізації власної логіки взаємодії користувача із системою, захищена авторизація користувачів на базі JWT-токенів [12], централізоване зберігання фотографій товарів у Cloudinary та висока масштабованість бази даних MongoDB. Такий підхід потребує детального проєктування архітектури системи та структури бази даних, однак дозволяє створити надійний, швидкодіючий і повністю контрольований вебсайт, який відповідає всім вимогам сучасного інтернет-магазину.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

Тема кваліфікаційної роботи – «Розробка вебсайту зоомагазину «ЛАПКА». Вебсайт призначений для автоматизації процесу продажу товарів для домашніх тварин, організації взаємодії між покупцями та адміністрацією магазину, а також централізованого керування каталогом продукції. Сайт забезпечує користувачам зручний доступ до товарів, швидке оформлення замовлень та можливість ефективного керування контентом через адміністративну панель.

Для користувачів передбачено можливість реєстрації власного облікового запису і переглядати інформацію про замовлення та історію покупок, де відображаються оформлені замовлення, інформація про товари та історія покупок. Крім цього, вебсайт повинен підтримувати функціонал кошика покупок, пошуку та фільтрації товарів за категоріями, а також взаємодію з REST API для отримання та оновлення даних у режимі реального часу.

Адміністративна панель повинна надавати можливість керувати товарами,

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

категоріями та фотографіями продукції, що дає змогу додавати, редагувати та видаляти товари, категорії та фотографії продукції. Для оптимізації роботи із зображеннями передбачається використання централізоване хмарне сховище Cloudinary, яка повинна надавати можливість зменшити навантаження на сервер та забезпечити швидке завантаження медіафайлів. Адміністративна панель також повинна надавати можливість переглядати та обробляти замовлення користувачів.

Також у вебсайту планується реалізувати адаптивну навігаційну структуру та сучасний інтерфейс користувача, завдяки чому сайт коректно відображається на різних типах пристроїв.

Завдяки впровадженню цієї системи покращується зручність користування інтернет-магазином, оптимізується процес адміністрування товарів та підвищується ефективність взаємодії з користувачами.

1.2.2 Призначення розробки

Метою розробки вебсайту зоомагазину «ЛАПКА» є створення сучасної онлайн-платформи для автоматизації процесу продажу товарів, організації взаємодії з користувачами та централізованого керування контентом.

Майбутній вебсайт зоомагазину «ЛАПКА» повинен забезпечити вирішення таких завдань:

- централізоване управління товарами та замовленнями: адміністратор зможе зручно керувати каталогом продукції, категоріями товарів, фотографіями та замовленнями користувачів;

- організація взаємодії з користувачами: користувачі матимуть можливість реєстрації, авторизації, перегляду товарів, використання пошуку та фільтрації, а також оформлення замовлень через кошик покупок;

- зручність використання та адміністрування: створення інтуїтивно зрозумілого інтерфейсу для користувачів і функціональної адміністративної панелі для керування товарами, фотографіями та замовленнями, що мінімізує помилки під час роботи із системою.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

Розробка вебсайту зоомагазину «ЛАПКА» спрямована на підвищення ефективності роботи магазину, спрощення процесу адміністрування, автоматизацію обробки замовлень та покращення взаємодії з користувачами шляхом створення сучасного й адаптивного вебресурсу.

1.2.3 Вимоги до функціоналу web-сайту

Для ефективної роботи з вебзастосунком «Розробка вебсайту зоомагазину «ЛАПКА» важливо забезпечити інтуїтивно зрозумілий інтерфейс і широкий функціонал для повноцінної взаємодії користувачів із системою та керування контентом. Основна мета цього вебсайту – спростити процес пошуку товарів, оформлення замовлень і забезпечити зручний інструмент для адміністрування інтернет-магазину.

Однією з ключових функцій вебсайту має бути централізоване зберігання та відображення інформації про товари, категорії продукції та фотографії. Для цього передбачається створення відповідної структури бази даних, яка забезпечуватиме впорядковане зберігання та обробку інформації.

- для товарів — назву, опис, категорію, вартість, кількість на складі та характеристики продукції;
- для категорій — назву, опис та перелік доступних товарів;
- для фотографій — назву та унікальну URL-адресу із хмарного сховища Cloudinary, винесену в окрему колекцію для уникнення дублювання файлів.

Завдяки цьому інформація про товари буде впорядкованою та зручною для перегляду. Для швидкого пошуку товарів у системі планується реалізувати категорії, фільтрацію та пошук.

В особистому кабінеті користувач може авторизованим користувачам швидко отримувати доступ до персоналізованої інформації. Це скоротить час на перегляд оформлених замовлень, відобразить історію покупок та забезпечить можливість взаємодії з кошиком товарів. Користувачі зможуть додавати товари до кошика, змінювати їх кількість та оформлювати замовлення через зручний

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

інтерфейс.

Додавання нових товарів, реалізація повного циклу CRUD-операцій для наявних записів, керування категоріями та фотографіями продукції, а також обробка замовлень належать до основних можливостей системи. Адміністратор може швидко оновлювати інформацію про товари, додавати нові фотографії, змінювати категорії та переглядати оформлені замовлення через спеціалізовану адміністративну панель.

Важливою складовою функціоналу вебсайту має бути захист даних користувачів і розмежування прав доступу. Для цього передбачається використання механізмів автентифікації та авторизації, які забезпечуватимуть доступ до функцій сайту відповідно до ролі користувача.

У системі передбачено розмежування прав доступу за ролями користувачів (user та admin). Адміністратор матиме повний доступ до функцій керування товарами, фотографіями та замовленнями, тоді як авторизовані користувачі зможуть взаємодіяти лише з власним профілем, кошиком і історією покупок. Такий підхід дозволить захистити персональні дані та забезпечити надійний контроль за роботою системи.

Завдяки впровадженню цієї функціональності вебсайт забезпечить автоматизацію процесу продажу товарів, спростить взаємодію користувачів із системою та оптимізує адміністрування інтернет-магазину. Це сприятиме підвищенню якості обслуговування клієнтів і забезпечить зручне цифрове середовище для здійснення онлайн-покупок.

1.2.4 Вимоги до програмної документації

Для забезпечення якісної роботи вебсайту зоомагазину «ЛАПКА» необхідно мати зрозумілу програмну документацію, яка міститиме опис структури системи, її функціональних можливостей, інтерфейсних рішень та правил використання. Документація містить інформацію для встановлення, налаштування та подальшої підтримки роботи вебсайту.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

До складу програмної документації повинні входити такі основні компоненти:

- опис архітектури вебсайту, який включає структуру клієнтської та серверної частин, REST API, модель бази даних MongoDB та механізм взаємодії між компонентами системи;
- інструкція з інсталяції та запуску програмного забезпечення, що містить вимоги до програмного середовища, порядок налаштування сервера, підключення бази даних і запуск frontend та backend частин проєкту;
- посібник користувача з описом функціоналу системи, зокрема роботи з каталогом товарів, використання пошуку та фільтрів, оформлення замовлень, перегляду кошика та особистого кабінету;
- інструкція адміністратора для роботи з адміністративною панеллю, яка містить опис функцій додавання, редагування та видалення товарів, категорій, фотографій і перегляду замовлень;
- технічна документація розробника, у якій описано структуру вихідного коду, особливості реалізації React-компонентів, серверної логіки на Node.js та Express, а також принципи роботи з базою даних;
- опис засобів безпеки системи, що включає механізми авторизації та автентифікації користувачів, розмежування прав доступу, захист персональних даних та безпечну роботу адміністративної частини;
- журнал змін і оновлень вебсайту, де фіксуються вдосконалення функціоналу, виправлення помилок та подальші зміни системи.

Повна технічна та користувацька документація спрощує використання вебсайту та його подальшу підтримку. Це повино надавати можливість швидко адаптувати систему до нових вимог і підтримувати стабільну роботу інтернет-магазину.

1.2.5 Техніко-економічні показники

Для забезпечення ефективного функціонування та подальшого розвитку

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

вебсайту «Розробка вебсайту зоомагазину «ЛАПКА»» важливо визначити основні техніко-економічні показники системи. Вони дозволяють оцінити часові та фінансові витрати на етапах проектування інтерфейсу, розробки клієнтської та серверної частин, створення бази даних, а також проаналізувати ефективність експлуатації вебсайту.

Аналіз техніко-економічних показників допомагає оцінити витрати на підтримку та розвиток системи. Крім цього, оцінка техніко-економічних характеристик повинен надавати можливість визначити ефективність автоматизації процесу продажу товарів і взаємодії з користувачами через вебсайт.

Детальний аналіз і повний перелік техніко-економічних показників розробленої системи наведено у відповідній таблиці графічної частини кваліфікаційної роботи. Отже, визначення техніко-економічних показників дає змогу оцінити доцільність розробки вебсайту зоомагазину «ЛАПКА», спрогнозувати витрати на його створення та подальшу підтримку. Отримані результати використовуються для розрахунку собівартості, ціни програмного продукту та оцінки економічної ефективності його впровадження.

1.2.6 Стадії та етапи розробки

Стадії та етапи розробки вебсайту зоомагазину «ЛАПКА» охоплюють послідовні кроки створення та реалізації програмного продукту. Послідовне виконання цих етапів допомагає створити стабільну систему, стабільну роботу вебсайту та відповідність функціональним вимогам.

На початковому етапі проводиться аналіз предметної області та визначається концепція вебсайту. Формується основна мета проєкту, визначаються ключові функції сайту та цільова аудиторія користувачів. Також здійснюється аналіз аналогічних вебресурсів, їхніх переваг і недоліків, що допомагає обрати оптимальні рішення для створення інтерфейсу, структури каталогу товарів і навігації сайту. Паралельно розробляється технічне завдання, яке містить опис функціональних вимог, структури вебсайту, вимог до безпеки та продуктивності.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

Після аналізу вимог виконується проєктування архітектури вебсайту та його основних модулів. Визначається структура системи, склад її компонентів і порядок їх взаємодії.

Визначається структура клієнтської частини на React і Vite, серверної частини на Node.js та Express, а також логіка взаємодії компонентів через REST API. Також було створено структуру бази даних MongoDB, яка містить інформацію про товари, категорії, користувачів, замовлення та фотографії продукції.

Після створення структури системи розпочинається програмна реалізація вебсайту. На цьому етапі розробляється серверна частина, реалізується бізнес-логіка, налаштовується база даних, REST API та механізми авторизації користувачів за допомогою JWT-токенів. Для зберігання фотографій товарів передбачається використання сервісу Cloudinary. Паралельно розробляється клієнтська частина вебсайту, де створюється інтерфейс користувача, налаштовується маршрутизація сторінок, реалізується взаємодія з API через Axios та адаптивний дизайн вебресурсу.

Після завершення етапу розробки проводиться тестування вебсайту. Під час тестування перевіряється робота основних функцій, зокрема реєстрації та авторизації користувачів, каталогу товарів, кошика покупок, оформлення замовлень і адміністративної панелі.

Також проводиться тестування продуктивності, адаптивності дизайну та коректного відображення вебсайту на різних типах пристроїв і браузерів.

Після завершення тестування систему було розгорнуто у робочому середовищі. На цьому етапі проводилося налаштування хмарних сервісів Vercel і Render [3,5], підключення бази даних MongoDB Atlas та конфігурація серверної частини системи. Також було підготовлено технічну документацію та інструкції для користувачів і адміністратора системи.

Після впровадження вебсайт може оновлюватися та доповнюватися новими функціями. Для підтримки стабільної роботи системи виконуються оновлення залежностей, виправлення помилок, оптимізація функціоналу та покращення

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

продуктивності. Окремо проводиться контроль безпеки, резервне копіювання даних і подальше вдосконалення вебсайту відповідно до нових вимог користувачів.

1.2.7 Порядок тестування та прийому

Під час розробки вебсайту зоомагазину «ЛАПКА» важливо встановити чіткий порядок контролю та прийому системи для забезпечення відповідності програмного продукту функціональним, технічним і експлуатаційним вимогам. Процес контролю та прийому складається з кількох послідовних етапів, кожен із яких має важливе значення для перевірки якості вебсайту.

Перевірка якості повинна проводитися на всіх етапах розробки вебсайту для своєчасного виявлення та усунення можливих помилок. Під час аналізу вимог перевіряли відповідність структури вебсайту технічному завданню та коректність реалізації основних сценаріїв роботи користувача із системою. Етап проєктування передбачає перевірку архітектури вебсайту, структури бази даних MongoDB, організації REST API та взаємодії між клієнтською і серверною частинами системи. Проміжне тестування спрямоване на перевірку окремих функціональних модулів, зокрема роботи авторизації, каталогу товарів, кошика покупок та адміністративної панелі.

Фінальне тестування повинно проводитися в умовах, максимально наближених до реальної експлуатації у середовищах Vercel і Render. На цьому етапі проводиться комплексне тестування інтерфейсу, перевіряється адаптивність дизайну, коректність відображення сторінок на мобільних і десктопних пристроях, а також швидкість завантаження даних і медіафайлів із Cloudinary. Окремо перевіряється робота авторизації та захисту даних користувачів, а також розмежування прав доступу між адміністратором і звичайними користувачами системи.

Після усунення виявлених помилок виконується приймання вебсайту. На цьому етапі перевіряється відповідність системи функціональним вимогам, стабільність роботи, зручність інтерфейсу та готовність програмного продукту до

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

подальшого використання.

Особливу увагу під час тестування необхідно приділити перевірці коректності роботи всіх функціональних модулів вебсайту, а також стабільності його роботи за різних умов використання. Результати тестування повинні підтвердити правильність реалізації функціональних вимог і готовність вебресурсу до використання кінцевими користувачами.

Після завершення всіх перевірок та усунення виявлених недоліків вебсайт може бути рекомендований до впровадження. Дотримання встановленого порядку контролю та прийому сприятиме підвищенню якості програмного продукту, його надійності та зручності використання.

Також виконується перевірка коректності роботи вебсайту в різних браузерах і на різних пристроях. Тестування проводиться в актуальних версіях Google Chrome, Mozilla Firefox, Microsoft Edge та інших популярних браузерів. Особлива увага приділяється адаптивності інтерфейсу та правильному відображенню всіх елементів на екранах різного розміру.

Позитивні результати проведених перевірок свідчать про готовність вебсайту до експлуатації та підтверджують можливість його використання для автоматизації діяльності зоомагазину «ЛАПКА». Впровадження розробленого програмного продукту сприятиме покращенню якості обслуговування клієнтів, підвищенню ефективності роботи персоналу та розвитку онлайн-продажів.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

2 РОЗРОБКА ТЕХНІЧОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури сайту і web-сторінок

Під час розробки вебсайту зоомагазину «ЛАПКА» було важливо спроектувати його загальну структуру, логіку навігації та взаємозв'язки між основними сторінками системи. Від правильності побудови структури вебсайту залежать зручність використання сайту, швидкість пошуку необхідних товарів і ефективність взаємодії користувача з функціональними можливостями вебсайту.

Розроблений вебсайт призначений для перегляду товарів для домашніх тварин, оформлення онлайн-замовлень, пошуку продукції за категоріями та взаємодії користувача із системою через особистий кабінет. Крім публічної частини, у проєкті реалізовано функціонал кошика покупок, систему авторизації користувачів та адміністративну панель для керування товарами, категоріями й замовленнями. Вебсайт використовується не лише для перегляду інформації про товари, а й для оформлення онлайн-замовлень, забезпечуючи взаємодію між користувачем, каталогом товарів та адміністративною частиною системи.

Під час створення структури вебсайту основну увагу приділено тому, щоб його структура була максимально зрозумілою для користувача та забезпечувала швидкий доступ до основних функцій сайту. Для користувача та забезпечувати швидкий доступ до основних функцій інтернет-магазину. Тому сторінки системи поділено на публічні та захищені. Публічні сторінки доступні всім відвідувачам сайту без авторизації та містять інформацію про товари, категорії, акційні пропозиції й контакти магазину. Захищені сторінки відкриваються лише після авторизації користувача та забезпечують доступ до особистого кабінету, історії замовлень і адміністративного функціоналу.

До основних публічних сторінок вебсайту зоомагазину «ЛАПКА» належать головна сторінка, каталог товарів, сторінка окремого товару, сторінка категорій, сторінка кошика покупок, сторінка оформлення замовлення, сторінка контактів, а також сторінки авторизації, реєстрації та відновлення пароля. На цих сторінках

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

користувачі можуть швидко переглядати інформацію про товари та користуватися основними функціями вебсайту. Додатково реалізовано сторінку помилки 404, яка відкривається у разі переходу на неіснуючий маршрут.

Головна сторінка використовується для ознайомлення користувача з асортиментом товарів та основними можливостями інтернет-магазину. Структура головної сторінки створена так, щоб користувач міг швидко ознайомитися з асортиментом товарів, актуальними пропозиціями та перейти до потрібного розділу. На головній сторінці розміщено головний банер, популярні категорії товарів, новинки, акційні товари, переваги магазину, відгуки покупців та інформаційні блоки про доставку й оплату.

У каталозі відображаються всі доступні товари інтернет-магазину. Користувач може здійснювати пошук товарів, використовувати фільтрацію та сортування за категоріями. Для кожного товару створено окрему сторінку, на якій міститься детальна інформація про товар, його характеристики, фотографії, ціна та можливість додавання до кошика.

Сторінка кошика дозволяє переглядати додані товари та оформлювати замовлення. У кошику користувач може змінювати кількість товарів, видаляти позиції та переглядати загальну суму замовлення. Після підтвердження вибору користувач переходить на сторінку оформлення замовлення, де вводяться контактні дані та інформація для доставки.

Контактна сторінка містить основну інформацію для зв'язку з магазином, контактні телефони, адресу електронної пошти та форму зворотного зв'язку. Сторінки авторизації та реєстрації забезпечують створення облікового запису користувача та доступ до персоналізованих функцій системи.

Захищена частина вебсайту представлена особистим кабінетом користувача. Після авторизації користувач отримує можливість переглядати власні замовлення, список обраних товарів, адреси доставки та персональні дані. Також у системі реалізовано адміністративну панель, яка дозволяє адміністратору керувати товарами, категоріями, замовленнями та користувачами.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

Головне меню сайту містить такі розділи «Головна», «Каталог», «Категорії», «Кошик», «Контакти» та «Кабінет користувача». Така структура меню допомагає швидко переходити між основними розділами сайту та робить інтерфейс більш зрозумілим для користувача. Додатково у нижній частині сторінок розміщено дублюючі навігаційні посилання для підвищення зручності користування вебресурсом.

У межах вебсайту зоомагазину «ЛАПКА» передбачено кілька основних сценаріїв взаємодії користувача із системою. Перший сценарій полягає в ознайомленні користувача з асортиментом товарів через головну сторінку, каталог та сторінки категорій. Другий сценарій передбачає перегляд детальної сторінки товару, ознайомлення з його характеристиками та додавання товару до кошика покупок. Третій сценарій пов'язаний із оформленням замовлення, введенням контактних даних та підтвердженням покупки. Четвертий сценарій охоплює роботу користувача з особистим кабінетом після авторизації. Окремий сценарій передбачено для адміністратора, який здійснює керування товарами, категоріями, замовленнями та користувачами через адміністративну панель.

Таким чином, розроблена структура вебсайту зоомагазину «ЛАПКА» забезпечує логічну організацію сторінок, зручну навігацію та швидкий доступ до основних функцій ресурсу. Це створює комфортні умови для пошуку товарів, оформлення замовлень і взаємодії користувачів із вебсайтом.

2.2 Створення та верстка сторінок сайту

Після розробки структури вебсайту було виконано створення та верстку його сторінок. На цьому етапі сформовано зовнішній вигляд основних сторінок ресурсу, визначено розташування елементів інтерфейсу та реалізовано адаптивну структуру вебсайту.

Верстка сторінок виконувалася з використанням HTML5, CSS3 та можливостей Tailwind CSS [14]. Під час розробки особлива увага приділялася адаптивності інтерфейсу, щоб вебсайт коректно відображався на персональних

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

комп'ютерах, планшетах і мобільних пристроях.

Для створення єдиного стилю оформлення використано спільну кольорову палітру, систему відступів, типографіку та набір повторно використовуваних елементів інтерфейсу. Основними складовими сторінок є навігаційне меню, інформаційні блоки, картки товарів, форми введення даних та службові елементи керування.

У результаті виконано верстку головної сторінки, каталогу товарів, сторінки окремого товару, кошика покупок, сторінок авторизації та реєстрації, особистого кабінету користувача, а також адміністративної частини вебсайту.

Загальна структура сторінок формується за допомогою компонента Layout. Він об'єднує верхню навігаційну панель, основний зміст сторінки та нижній блок сайту. Navbar містить основне меню навігації сайту, Footer — за нижню частину сторінки, а центральна область використовується для відображення основного контенту. Анімації переходів між елементами реалізовано за допомогою бібліотеки Framer Motion. Структуру загального макета сторінки подано на рисунку 2.1.

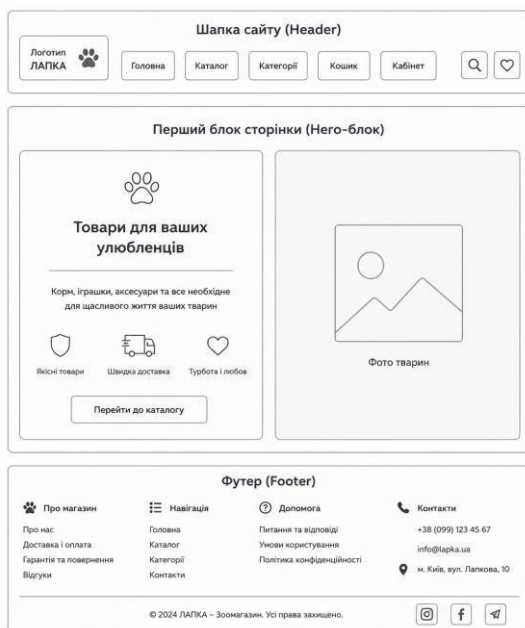


Рисунок 2.1 – Загальна структура макета сторінки

У процесі верстки вебсайту зоомагазину «ЛАПКА» було використано

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

компонентний підхід. Повторювані елементи інтерфейсу винесено в окремі компоненти, це спрощує редагування інтерфейсу та допомагає підтримувати єдиний стиль сторінок. До таких елементів належать кнопки, картки товарів, поля форм, блоки зображень, повідомлення про помилки, стани завантаження та елементи порожнього результату. Тому зміни в інтерфейсі можна вносити швидше без необхідності редагування кожної сторінки окремо

Основні компоненти клієнтської частини наведено в таблиці 2.1.

Таблиця 2.1 – Основні компоненти клієнтської частини вебсайту

Компонент	Призначення
Layout	Формує загальну структуру сторінки
Navbar	Відповідає за верхню навігацію сайту
Footer	Відображає нижній інформаційний блок
PageHero	Формує вступний блок сторінки
Button	Забезпечує єдиний вигляд кнопок
FormField	Використовується для полів форм
CartItem	Відображає товар у кошику покупок
LazyImage	Оптимізує відображення зображень товарів
LoadingState	Відображає стан завантаження даних
Skeleton	Показує тимчасовий макет під час завантаження контенту
ThemeToggle	Забезпечує перемикання теми оформлення

Дизайн вебсайту побудований у сучасному стилі з використанням спокійної та зрозумілої кольорової гами. Основні кольори інтерфейсу підбрані так, щоб сторінки виглядали охайно, не перевантажували користувача та відповідали тематиці інтернет-магазину товарів для домашніх тварин.

Акцентні кольори використовуються для виділення кнопок, посилань, цін і

важливих елементів, пов'язаних із додаванням товару до кошика або оформленням замовлення.

Для заголовків використано виразні шрифти, які добре сприймаються на екрані, а основний текст оформлено сучасними системними шрифтами. Завдяки цьому текст зручно читати на різних пристроях описів товарів, характеристик, назв категорій і службової інформації на різних пристроях.

Вебсайт має адаптивну структуру, тому сторінки коректно відображаються на мобільних телефонах, планшетах, ноутбуках і великих моніторах. Для цього використано breakpoint-систему Tailwind CSS. Сітки сторінок будуються за допомогою адаптивних класів, які автоматично змінюють кількість колонок, розміри відступів і розташування блоків залежно від ширини екрана.

На мобільних пристроях головна навігація замінюється на burger-menu, що дозволяє зручно користуватися вебзастосунком навіть на невеликих екранах. На великих екранах відображається повна горизонтальна навігація з доступом до основних розділів: головної сторінки, каталогу, категорій, кошика, контактів і особистого кабінету користувача.

Для покращення взаємодії з користувачем використовуються повідомлення react-hot-toast. Вони інформують користувача про результат виконання дій, наприклад успішне додавання товару до кошика, помилку авторизації, некоректно заповнені поля форми або успішне оформлення замовлення. У випадках, коли дані завантажуються з сервера, відображаються loading states або skeleton-елементи. Якщо товари відсутні або виникає помилка під час завантаження, користувач бачить відповідне повідомлення, а не порожню сторінку.

Під час створення та верстки сторінок вебсайту «ЛАПКА» було враховано адаптивність інтерфейсу, зручну навігацію, повторне використання компонентів та цілісний зовнішній вигляд сторінок інтернет-магазину.

2.3 Розробка структури бази даних сайту

Для забезпечення роботи вебсайту зоомагазину «ЛАПКА» було розроблено

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

структуру бази даних, яка забезпечує зберігання, обробку та пошук інформації про товари, користувачів, категорії товарів, замовлення та фотографії.

Під час проектування бази даних було визначено основні об'єкти предметної області та їх властивості. На основі аналізу функціональних вимог сформовано набір колекцій, необхідних для реалізації функціоналу вебсайту.

До основних колекцій бази даних належать Users, Products, Categories, Orders та Photos. Колекція Users використовується для зберігання інформації про користувачів системи. Колекція Products містить дані про товари, їх характеристики, ціни та залишки. Колекція Categories використовується для групування товарів за категоріями. Колекція Orders забезпечує зберігання інформації про замовлення користувачів, а Photos містить інформацію про фотографії товарів.

Під час розробки структури бази даних було визначено необхідні поля для кожної колекції, типи даних та логічні зв'язки між об'єктами. Такий підхід забезпечує цілісність даних, спрощує їх обробку та дозволяє розширювати функціональні можливості вебсайту в майбутньому.

Особливу увагу під час проектування бази даних приділено забезпеченню цілісності та узгодженості інформації між окремими колекціями. Для цього кожна колекція виконує власну функцію та використовується лише для зберігання визначеного набору даних.

Колекції взаємодіють між собою під час виконання основних операцій вебсайту. Наприклад, під час оформлення замовлення використовуються дані про користувача, обрані товари та інформація про категорії продукції. Це дає змогу отримувати необхідні відомості без дублювання даних і забезпечує ефективну роботу системи.

Розроблена структура бази даних також враховує можливість подальшого розвитку вебсайту. За необхідності до системи можуть бути додані нові колекції та поля без суттєвих змін існуючої структури даних.

Основні колекції бази даних та їх призначення наведено в таблиці 2.2.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Таблиця 2.2 – Структура колекцій бази даних та їх призначення

Колекція	Призначення
Users	Зберігання інформації про користувачів системи, їх облікові записи, ролі та персональні дані
Products	Зберігання інформації про товари, їх характеристики, ціни, залишки та належність до категорій
Categories	Зберігання категорій товарів для групування продукції та організації навігації каталогом
Orders	Зберігання інформації про замовлення користувачів, склад замовлення, статус та дату створення
Photos	Зберігання інформації про фотографії товарів та посилань на зображення у хмарному сховищі

Наведені колекції формують основу структури бази даних вебсайту зоомагазину «ЛАПКА». Кожна з них відповідає за зберігання окремого типу інформації та використовується під час виконання відповідних функцій системи.

Колекція Users забезпечує роботу механізмів реєстрації, авторизації та керування правами доступу користувачів. Колекції Products і Categories використовуються для формування каталогу товарів та організації зручної навігації вебсайтом. Колекція Orders призначена для обробки та зберігання інформації про замовлення покупців, а колекція Photos забезпечує роботу із зображеннями товарів.

Використання окремих колекцій для різних типів даних дозволяє уникнути дублювання інформації, підвищити продуктивність системи та спростити її

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

подальше супроводження. Крім того, така структура забезпечує можливість розширення функціоналу вебсайту шляхом додавання нових сутностей без суттєвої зміни вже існуючої архітектури бази даних.

Отже, розроблена структура бази даних повністю відповідає функціональним вимогам вебсайту зоомагазину «ЛАПКА» та забезпечує надійне зберігання й обробку інформації, необхідної для роботи всіх основних модулів системи.

2.4 Програмування сайту

Програмування вебсайту зоомагазину «ЛАПКА» виконувалося із застосуванням сучасних web-технологій та Full-Stack підходу. Такий підхід передбачає розділення системи на клієнтську та серверну частини, які взаємодіють між собою через REST API.

Написання програмного коду виконувалося у середовищі Visual Studio Code. Для контролю версій використовувалася система Git, а зберігання вихідного коду здійснювалося за допомогою репозиторію GitHub. Для створення клієнтської частини застосовано React та Vite, а серверна частина реалізована на платформі Node.js із використанням фреймворку Express.js.

Клієнтська частина відповідає за відображення інтерфейсу користувача, навігацію між сторінками, роботу з формами та обмін даними із сервером. Серверна частина забезпечує обробку запитів користувачів, взаємодію з базою даних, авторизацію, керування товарами, категоріями та замовленнями.

Під час програмування було реалізовано модульну структуру проєкту [4], яка забезпечує розподіл функціоналу між окремими компонентами системи. Такий підхід спрощує супровід програмного коду, подальше розширення можливостей вебсайту та внесення змін до його функціоналу.

Основні програмні модулі вебсайту охоплюють роботу каталогу товарів, кошика покупок, системи авторизації користувачів, оформлення замовлень та адміністративної панелі. Взаємодія між окремими модулями забезпечує коректне

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

функціонування вебсайту та обробку даних користувачів.

Основні API-маршрути наведено в таблиці 2.3.

Таблиця 2.3 – Основні API-маршрути web-застосунку

API-маршрут	Призначення
/api/users	Реєстрація, авторизація, вихід, профіль користувача
/api/products	Отримання товарів та інформації про них
/api/categories	Робота з категоріями товарів
/api/cart	Робота з кошиком покупок
/api/orders	Створення та перегляд замовлень
/api/payments	Робота з оплатою замовлень
/api/reviews	Робота з відгуками користувачів
/api/admin	Адміністративне керування
/api/media	Службові маршрути для медіафайлів
/api/health	Перевірка працездатності backend

Авторизація користувачів реалізована за допомогою JWT. Після успішного входу або реєстрації користувач отримує токен, який зберігається у localStorage. Під час виконання захищених запитів токен передається у заголовок Authorization. На сервері middleware requireAuth перевіряє наявність і правильність токена, а middleware requireAdmin додатково перевіряє роль користувача для доступу до адміністративних функцій.

Під час реєстрації користувач вводить необхідні облікові дані, які передаються на сервер. Backend перевіряє отриману інформацію, створює нового користувача, хешує пароль за допомогою bcryptjs та повертає токен авторизації. Під час входу система перевіряє email і пароль, після чого оновлює стан авторизації на клієнтській частині web-застосунку.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

Сценарій оформлення замовлення передбачає перегляд товару, додавання його до кошика та створення замовлення. Якщо користувач не авторизований, система перенаправляє його на сторінку входу. Після авторизації користувач може оформити замовлення, вказати контактні дані та підтвердити покупку. Після успішного створення замовлення інформація зберігається у базі даних, а користувач отримує повідомлення про результат виконання операції. Схему оформлення замовлення подано на рисунку 2.2.

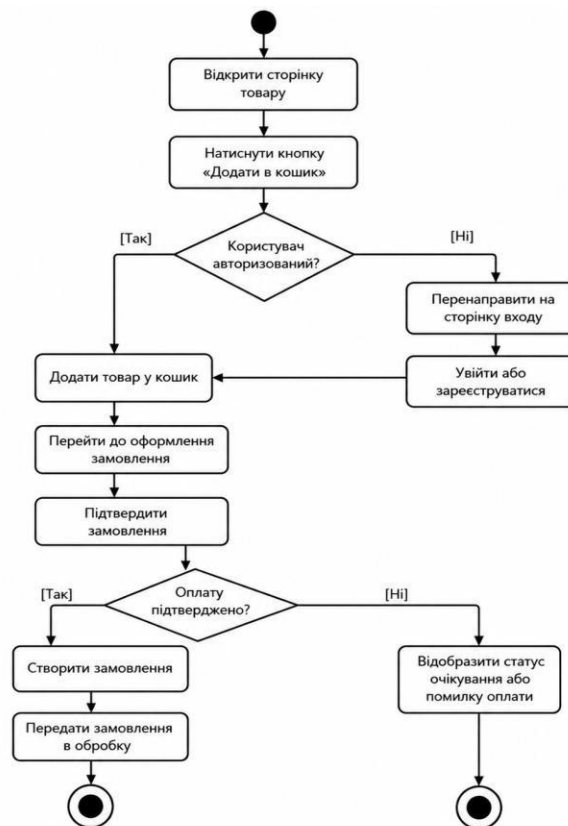


Рисунок 2.2 – Схема оформлення замовлення користувачем

Сценарій оформлення замовлення реалізовано через сторінку кошика та маршрут оформлення покупки. Під час переходу до оформлення замовлення backend перевіряє, чи авторизований користувач та чи містить кошик товари. Якщо перевірка успішна, система формує замовлення та передає дані для подальшої обробки. Користувач може переглядати список товарів, змінювати їх кількість або видаляти окремі позиції із кошика перед підтвердженням покупки.

Сценарій оформлення покупки залежить від обраного способу оплати. Після підтвердження замовлення система створює запис у колекції orders та зберігає інформацію про товари, суму замовлення, користувача та статус оплати. У разі успішної оплати замовлення отримує відповідний статус, а користувач отримує повідомлення про успішне оформлення покупки.

Для роботи із зображеннями товарів використовується сервіс Cloudinary. Адміністратор завантажує фотографії через адміністративну панель, після чого сервер передає файл до Cloudinary, отримує URL-адресу зображення та зберігає її у базі даних. Надалі ці зображення використовуються у каталозі товарів, картках товарів та адміністративній панелі.

Надійність роботи серверної частини забезпечується хешуванням паролів, JWT-авторизацією, перевіркою ролей користувачів, налаштуванням CORS, використанням helmet, обмеженням частоти запитів через express-rate-limit та централізованою обробкою помилок. Такий підхід дозволяє зменшити ризик несанкціонованого доступу до даних, підвищити безпеку web-застосунку та забезпечити стабільну роботу системи.

2.4.1 Написання клієнтської частини

Клієнтська частина вебсайту зоомагазину «ЛАПКА» реалізована з використанням бібліотеки React та інструмента Vite. Такий підхід дозволяє створити швидкий та зручний користувацький інтерфейс, який забезпечує комфортну взаємодію користувача із системою.

Структура клієнтської частини побудована за компонентним принципом. Основні елементи інтерфейсу реалізовано у вигляді окремих компонентів, серед яких Layout, Navbar, Footer, ProductCard, CategoryCard, CartItem, FormField та інші. Повторне використання компонентів спрощує супровід програмного коду та забезпечує єдиний стиль оформлення сторінок.

Для організації навігації між сторінками використовується React Router DOM [15]. За допомогою системи маршрутів реалізовано переходи між головною

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

сторінкою, каталогом товарів, сторінкою товару, кошиком покупок, сторінками авторизації та особистим кабінетом користувача.

Обмін даними між клієнтською та серверною частинами здійснюється за допомогою бібліотеки Axios. Клієнтська частина надсилає HTTP-запити до REST API та отримує необхідну інформацію про товари, категорії, замовлення та користувачів. Для керування станом авторизації використовується компонент AuthContext, який забезпечує доступ до інформації про поточного користувача в різних частинах вебзастосунку.

Для покращення взаємодії з користувачем використовуються повідомлення react-hot-toast, а також анімації Framer Motion. Завдяки цьому користувач отримує повідомлення про успішне виконання операцій або виникнення помилок під час роботи із системою.

Таким чином, клієнтська частина забезпечує відображення інтерфейсу, навігацію між сторінками, взаємодію з сервером та зручне використання основних функціональних можливостей вебсайту.

2.4.2 Написання admin частини

Серверна частина вебсайту зоомагазину «ЛАПКА» реалізована на платформі Node.js із використанням фреймворку Express.js. Backend-частина забезпечує обробку HTTP-запитів, взаємодію з базою даних MongoDB Atlas, авторизацію користувачів, роботу з товарами, категоріями, замовленнями та адміністративними функціями системи.

Структура серверної частини побудована за модульним принципом. Основні маршрути розміщені у папці routes, контролери — у controllers, моделі бази даних — у models, middleware-компоненти — у middleware, а допоміжні функції — у utils. Такий підхід спрощує супровід програмного коду та забезпечує можливість подальшого розширення функціоналу вебсайту.

Для взаємодії з базою даних використовується бібліотека Mongoose. Вона дозволяє описувати структуру документів, виконувати перевірку даних та

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

працювати з колекціями MongoDB за допомогою об'єктної моделі.

Авторизація користувачів реалізована за допомогою JWT-токенів. Після успішного входу до системи користувач отримує токен доступу, який використовується для виконання захищених запитів. Для перевірки прав доступу застосовуються middleware-компоненти `requireAuth` та `requireAdmin`, які контролюють доступ до службових і адміністративних функцій.

Для захисту облікових записів паролі користувачів зберігаються у вигляді хешів, створених за допомогою бібліотеки `bcryptjs`. Це підвищує безпеку системи та зменшує ризик несанкціонованого доступу до персональних даних користувачів.

Завантаження та зберігання фотографій товарів реалізовано за допомогою сервісу `Cloudinary`. Після завантаження файлу сервер отримує URL-адресу зображення та зберігає її у базі даних. Це дозволяє зменшити навантаження на сервер та прискорити завантаження сторінок вебсайту.

Для підвищення безпеки web-застосунку використовуються middleware-компоненти `helmet`, `cors` та `express-rate-limit`. Вони забезпечують захист від поширених мережових атак, контролюють доступ до ресурсів системи та обмежують частоту виконання запитів.

Таким чином, серверна частина забезпечує стабільну роботу вебсайту, обробку запитів користувачів, взаємодію з базою даних, виконання бізнес-логіки та підтримку основних функціональних можливостей інтернет-магазину «ЛІАПКА».

2.5 Тестування сайту

Тестування вебсайту зоомагазину «ЛІАПКА» виконувалося з метою перевірки коректності роботи основних функцій системи, стабільності клієнтської та серверної частин, правильності роботи форм, авторизації, кошика покупок, оформлення замовлень, адміністративної панелі та адаптивності інтерфейсу.

Під час перевірки використовувалися функціональне тестування, тестування інтерфейсу користувача, тестування навігації, тестування авторизації, адаптивне

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

тестування та кросбраузерне тестування. Перевірка виконувалася як для клієнтської, так і для серверної частин вебзастосунку.

Функціональне тестування передбачало перевірку основних сценаріїв роботи користувача. Було перевірено перегляд каталогу товарів, відкриття сторінок товарів, додавання товарів до кошика, оформлення замовлень, реєстрацію та авторизацію користувачів, а також роботу особистого кабінету та адміністративної панелі.

Для перевірки працездатності вебсайту було сформовано контрольний список тестування (checklist), який містить основні функціональні можливості системи та результати їх перевірки. Контрольний список наведено в таблиці 2.4.

Таблиця 2.4 – Контрольний список тестування вебсайту

№	Перевірка	Результат
1	Відкриття головної сторінки	Успішно
2	Перегляд каталогу товарів	Успішно
3	Відкриття сторінки товару	Успішно
4	Пошук товарів	Успішно
5	Додавання товару до кошика	Успішно
6	Видалення товару з кошика	Успішно
7	Авторизація користувача	Успішно
8	Оформлення замовлення	Успішно
9	Робота адміністративної панелі	Успішно
10	Додавання нового товару	Успішно
11	Редагування товару	Успішно
12	Видалення товару	Успішно

Після формування контрольного списку було проведено практичне тестування

основних сторінок вебсайту та його функціональних модулів. Під час перевірки аналізувалася коректність відображення інтерфейсу, робота форм введення даних, маршрутизація між сторінками, авторизація користувачів та функціонування адміністративної панелі.

Результати тестування основних сторінок вебсайту наведено рисунку 2.3

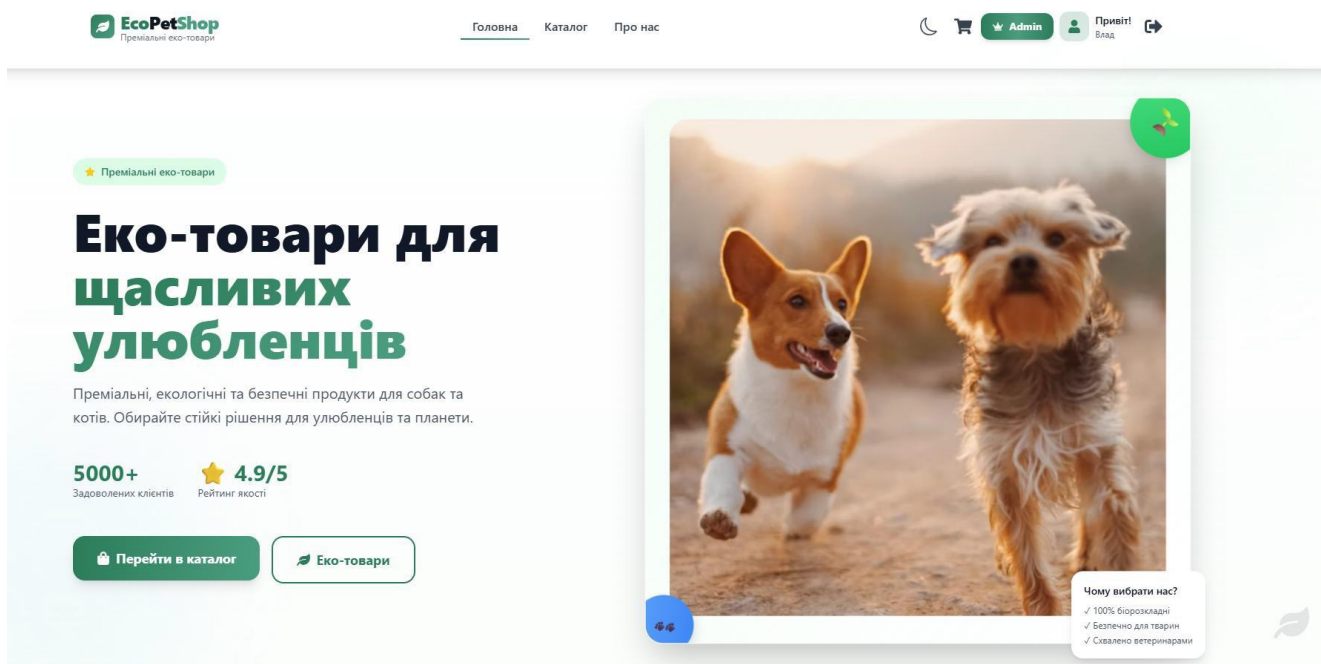


Рисунок 2.3 – Головна сторінка вебсайту

На рисунку 2.3 наведено головну сторінку вебсайту зоомагазину «ЛАПКА». Під час тестування було перевірено коректність відображення основних елементів інтерфейсу, навігаційного меню, інформаційних блоків та кнопок переходу до каталогу товарів. Результати перевірки показали правильну роботу всіх елементів сторінки та їх коректне відображення на екрані користувача.

Крім того, було перевірено адаптивність головної сторінки під час відображення на пристроях із різними розмірами екранів. Особливу увагу приділено коректності відображення банерів, інформаційних блоків, кнопок навігації та елементів меню. Під час тестування не виявлено помилок у роботі інтерфейсу, а всі елементи сторінки залишалися доступними та зручними для використання. Отримані результати підтвердили відповідність головної сторінки

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

встановленим функціональним та ергономічним вимогам.

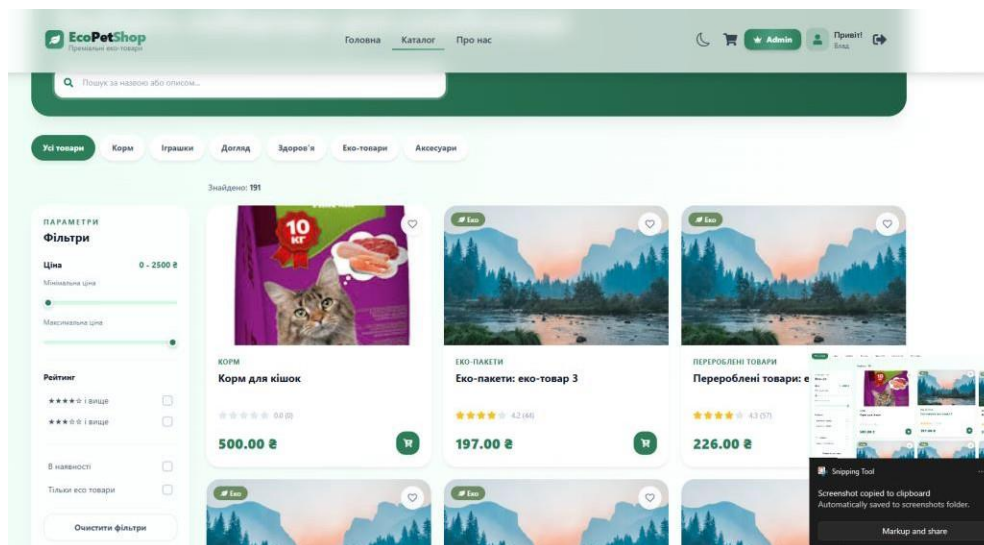


Рисунок 2.4 – Каталог товарів

Під час тестування каталогу товарів було перевірено коректність завантаження інформації з бази даних, відображення карток товарів, роботу пошуку та навігації між сторінками вебсайту. Також перевірялося коректне відображення назв товарів, цін, фотографій та категорій.

У результаті тестування встановлено, що каталог товарів функціонує коректно, а інформація про товари відображається без помилок. Користувач може швидко знайти необхідний товар та перейти до перегляду його детальної інформації. Результати перевірки сторінки окремого товару наведено на рисунку 2.5.



Рисунок 2.5 – Сторінка товару

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

Тестування сторінки товару виконувалося для перевірки коректності відображення детальної інформації про товар. Під час перевірки контролювалася робота завантаження фотографій, відображення назви товару, опису, ціни, характеристик та кнопки додавання товару до кошика.

Також перевірялося коректне завантаження сторінки після прямого переходу за посиланням та правильність роботи маршрутизації. У результаті тестування встановлено, що сторінка товару функціонує без помилок і надає користувачеві повну інформацію про обраний товар.

Наступним етапом була перевірка роботи кошика покупок та процесу оформлення замовлення. Результати тестування кошика покупок наведено на рисунку 2.6.

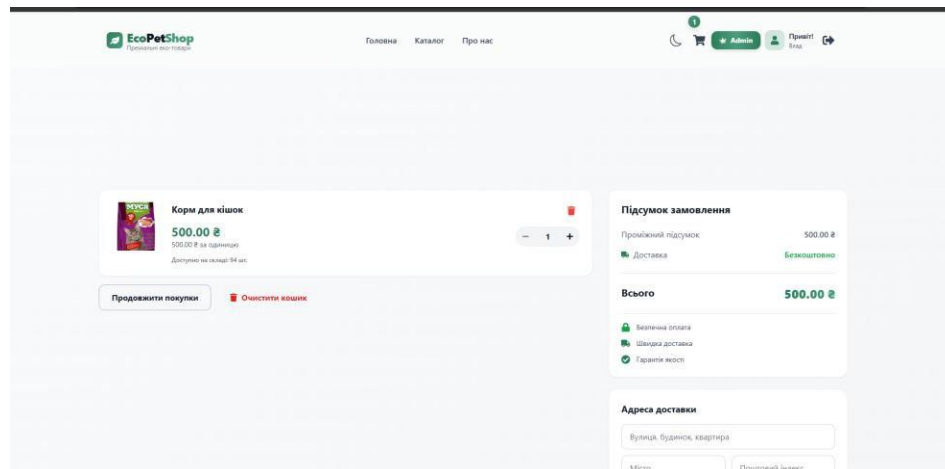


Рисунок 2.6 – Кошик покупок

Під час тестування кошика покупок було перевірено додавання товарів, зміну їх кількості, видалення окремих позицій та автоматичний перерахунок загальної вартості замовлення. Також перевірялася коректність збереження даних кошика після оновлення сторінки та повторної авторизації користувача.

У результаті перевірки встановлено, що кошик коректно обробляє дії користувача та забезпечує правильне формування замовлення. Всі зміни

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

відображаються миттєво, а інформація про товари синхронізується із серверною частиною вебзастосунку.

Після формування кошика було проведено тестування процесу оформлення замовлення. Перевірялося введення контактних даних, створення замовлення та збереження інформації у базі даних. Результати перевірки сторінки оформлення замовлення наведено на рисунку 2.7.

The screenshot displays a checkout interface with the following elements:

- Всього** (Total): 500.00 ₴
- Service icons: Безпечна оплата (Secure payment), Швидка доставка (Fast delivery), and Гарантія якості (Quality guarantee).
- Адреса доставки** (Delivery address) section with input fields for:
 - Item: леся курбаса
 - City: Тернопіль
 - Postal code: 45667
 - Country: Україна
 - Phone number: 0684364446
- Спосіб оплати** (Payment method) section with radio buttons for:
 - visa Кредитна картка
 - Mastercard
 - PayPal
 - Оплата при отриманні (Checked)
- A green button at the bottom: **→ Оформити замовлення**

Рисунок 2.7 – Оформлення замовлення

Тестування процесу оформлення замовлення виконувалося з метою перевірки правильності створення нових замовлень та передачі даних між клієнтською і серверною частинами системи. Під час перевірки контролювалася коректність заповнення форми, збереження контактних даних користувача та створення запису про замовлення у базі даних.

У результаті тестування встановлено, що процес оформлення замовлення працює коректно. Після підтвердження покупки система успішно створює

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

замовлення та відображає повідомлення про результат виконання операції.

Наступним етапом було тестування механізмів авторизації та реєстрації користувачів, які забезпечують доступ до персоналізованих функцій вебсайту. Результати перевірки наведено на рисунку 2.8.

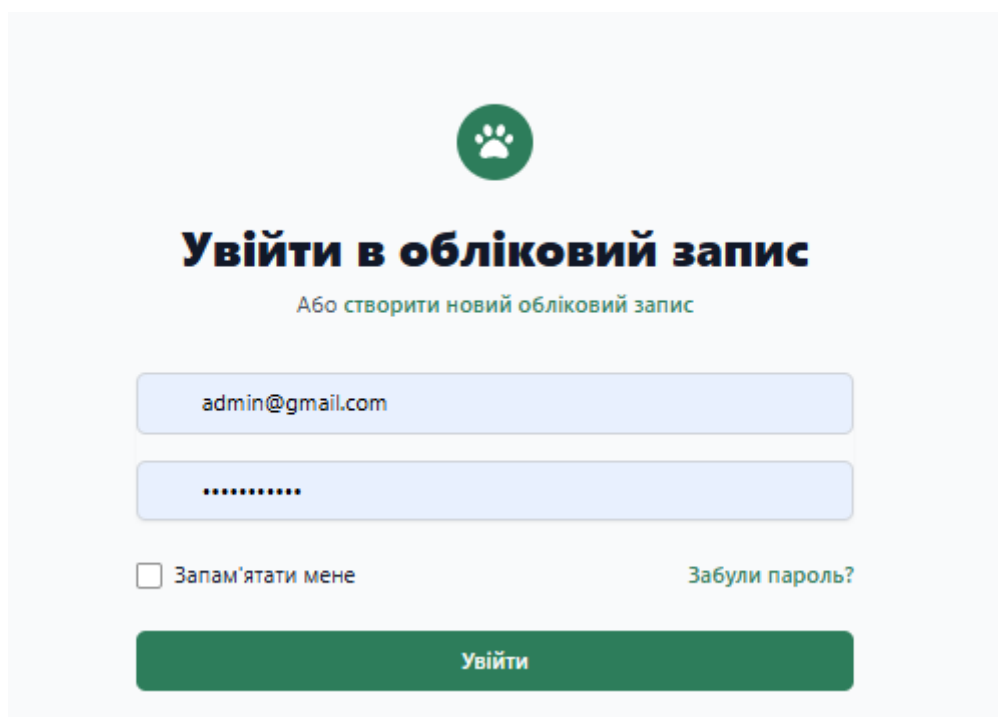


Рисунок 2.8 – Авторизація користувача

Під час тестування авторизації користувача було перевірено правильність обробки облікових даних, роботу форми входу та механізму перевірки прав доступу. Також перевірялася реакція системи на введення некоректних даних та спробу доступу до захищених сторінок без авторизації.

У результаті тестування встановлено, що система коректно виконує автентифікацію користувачів, забезпечує доступ до особистого кабінету після успішного входу та блокує доступ до захищених сторінок для неавторизованих користувачів.

Для перевірки функцій керування товарами було проведено тестування форми створення нового товару. Результати перевірки наведено на рисунку 2.9.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

Додати товар

НАЗВА ТОВАРУ

КАТЕГОРІЯ

ОПИС

ЦІНА, ₴

КІЛЬКІСТЬ НА СКЛАДІ

ПІДКАТЕГОРІЯ

ОБЕРІТЬ ФОТО З БІБЛІОТЕКИ

IMAGE URL FALLBACK

СМАК

Необов'язково. Використовується, якщо фото з бібліотеки не вибрано.

ДЛЯ КОГО

ВІКОВА ГРУПА

ТИП ТОВАРУ

РЕЙТИНГ

КІЛЬКІСТЬ ВІДГУКІВ

Екологічний товар
 Позначте товар як дружній до довкілля.

Додати товар

Рисунок 2.9 – Додавання нового товару

Тестування форми додавання товару передбачало перевірку коректності введення даних, завантаження фотографій, вибору категорії та збереження інформації у базі даних. Під час перевірки контролювалася робота всіх полів форми та правильність передачі даних на сервер.

У результаті тестування встановлено, що нові товари успішно створюються та відображаються в каталозі після збереження. Завантаження фотографій і збереження інформації виконуються без помилок.

Отже, проведене тестування підтвердило коректність роботи основних функцій вебсайту зоомагазину «ЛАПКА». Було перевірено роботу каталогу товарів, кошика покупок, оформлення замовлень, авторизації користувачів та адміністративної панелі. Результати тестування засвідчили стабільну роботу вебзастосунку та його відповідність поставленим функціональним вимогам.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розміщення сайту в Інтернеті

Розміщення вебсайту зоомагазину «ЛАПКА» у мережі Інтернет є завершальним етапом розробки та надає користувачам доступ до системи через доменне ім'я.

Основною метою цього етапу є забезпечення стабільної роботи вебсайту, його доступності для користувачів та коректного відображення інтерфейсу у сучасних браузерях.

На початковому етапі було обрано хмарні платформи Vercel та Render, які підтримують розгортання сучасних вебзастосунків на основі React і Node.js. Після створення облікових записів на відповідних сервісах було налаштовано окремі проєкти для клієнтської та серверної частин системи.

Далі виконано підключення GitHub-репозиторію та автоматичне розгортання вебсайту після оновлення програмного коду. Клієнтська частина вебсайту розміщується на платформі Vercel, а серверна частина та REST API — на Render.

До складу вебсайту входять:

- головна сторінка;
- каталог товарів;
- сторінка окремого товару;
- кошик покупок;
- сторінка авторизації та реєстрації;
- особистий кабінет користувача;
- адміністративна панель;
- стилі CSS та JavaScript-модулі;
- медіафайли та фотографії товарів.

Перед публікацією було проведено перевірку коректності маршрутів, API-

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

запитів, підключення бази даних MongoDB Atlas та адаптивності інтерфейсу для уникнення помилок після розгортання системи.

Після завершення налаштування вебсайт було прив'язано до доменного імені. Для цього у DNS-налаштуваннях були створені записи типу A та CNAME, які забезпечують коректне підключення домену до хмарних платформ. Це дозволило зробити вебсайт доступним для користувачів через мережу Інтернет. Результат розміщення вебсайту в мережі Інтернет наведено на рисунку 3.1.

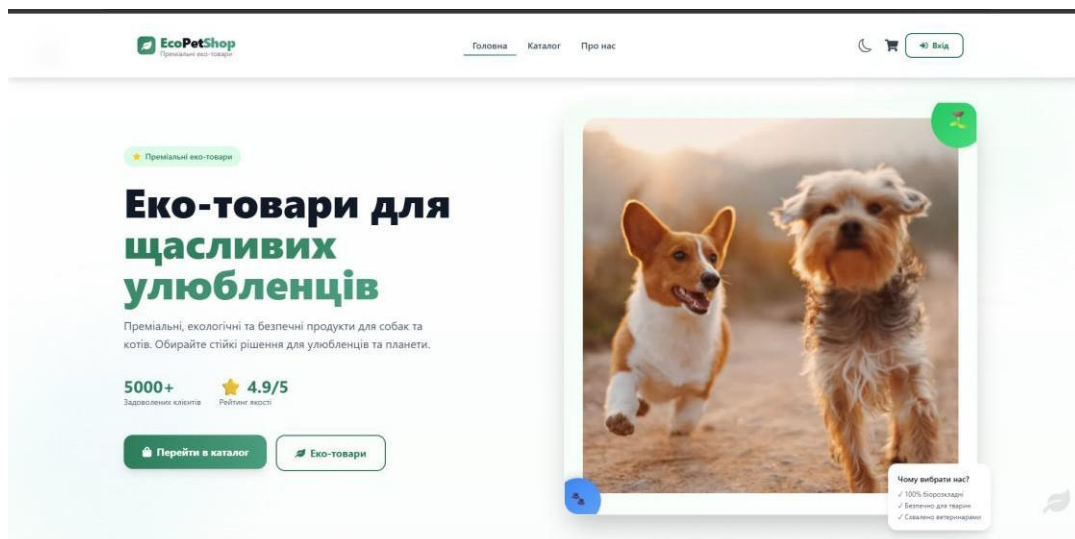


Рисунок 3.1 – Головна сторінка сайту після розміщення на хостингу

Для перевірки коректності роботи вебсайту було протестовано:

- відкриття сторінок через доменне ім'я;
- коректність роботи навігації між сторінками;
- адаптивність інтерфейсу на різних типах пристроїв;
- швидкість завантаження сторінок;
- коректність відображення контенту;
- роботу API-запитів між клієнтською та серверною частинами;
- функціонування авторизації користувачів;
- роботу адміністративної панелі та CRUD-операцій.

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

3.2 Інструкція з обслуговування та наповнення сайту

Адміністративна панель вебсайту зоомагазину «ЛАПКА» використовується для керування товарами, категоріями, фотографіями та замовленнями. Робота з даними виконується через вебінтерфейс адміністратора, тому внесення змін не потребує редагування програмного коду або безпосередньої роботи з базою даних.

Після авторизації адміністратор отримує доступ до розділів керування товарами, категоріями та замовленнями. Для кожного розділу доступні операції створення, перегляду, редагування та видалення записів.

Для додавання нового товару необхідно відкрити розділ керування товарами та натиснути кнопку створення нового запису. Після цього заповнюються основні характеристики товару: назва, опис, ціна, категорія, кількість на складі та фотографія. Після збереження інформація автоматично відображається у каталозі вебсайту.

Редагування товарів виконується через форму зміни даних, де адміністратор може оновити характеристики товару, змінити його вартість, кількість або фотографію. За потреби товар може бути видалений із системи.

Для роботи із замовленнями передбачено окремий розділ адміністративної панелі, у якому відображається інформація про покупців, склад замовлення, дату створення та поточний статус обробки. Адміністратор може переглядати замовлення та змінювати їх статус відповідно до етапу виконання.

Таким чином, адміністративна панель забезпечує зручне керування інформацією вебсайту та підтримання актуальності каталогу товарів без втручання у програмний код системи. Це дозволяє оперативно додавати нові товари, редагувати наявні дані, контролювати замовлення користувачів та ефективно підтримувати роботу інтернет-магазину. Завдяки цьому підвищується якість обслуговування клієнтів і спрощується процес адміністрування вебсайту. Рисунок 3.2 – Форма додавання нового товару в адміністративній панелі

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

НОВИЙ ТОВАР
Додати товар

НАЗВА ТОВАРУ

КАТЕГОРІЯ

ОПИС

ЦІНА, ₪

КІЛЬКІСТЬ НА СКЛАДІ

ПІДКАТЕГОРІЯ

ОБЕРІТЬ ФОТО З БІБЛІОТЕКИ

ІМAGE URL FALLBACK

СМАК

Необов'язково. Використовується, якщо фото з бібліотеки не вибрано.

ДЛЯ КОГО

ВІКОВА ГРУПА

ТИП ТОВАРУ

РЕЙТИНГ

КІЛЬКІСТЬ ВІДГУКІВ

ФОТО ТОВАРУ




Фото з бібліотеки або fallback URL
Основний сценарій: виберіть фото з медіабібліотеки. Поле Image URL можна лишити як резервний варіант.

Рисунок 3.2 – Форма додавання нового товару в адміністративній панелі

Редагування інформації виконується через адміністративний інтерфейс вебсайту.Адміністратор має можливість змінювати:

- назву товару;
- опис товару;
- вартість;
- категорію;
- кількість товару на складі;
- фотографії товарів.

Після внесення змін оновлена інформація автоматично відображається на сайті після оновлення сторінки, що дозволяє підтримувати актуальність каталогу товарів для користувачів.

У разі необхідності товар може бути видалений або прихований із каталогу шляхом:

- видалення запису з бази даних;
- зміни статусу відображення товару;
- архівування товарної позиції.

Для стабільної роботи вебсайту рекомендовано:

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

- підтримку HTTP/HTTPS-протоколів;
- не менше 1 ГБ оперативної пам'яті на сервері;
- стабільне підключення до мережі Інтернет;
- підтримку Node.js-середовища;
- підтримку MongoDB Atlas;
- використання SSL-сертифіката для захищеного з'єднання.

3.3 Інструкція з популяризації та підтримки сайту

Після публікації вебсайту зоомагазину «ЛАПКА» важливо забезпечити його популяризацію та підтримання актуальності, що сприятиме залученню нових користувачів і підвищенню видимості ресурсу в пошукових системах.

Основні методи просування вебсайту

SEO-оптимізація

SEO (Search Engine Optimization) передбачає оптимізацію вебсайту для пошукових систем з метою покращення позицій сайту у результатах пошуку. Для цього використовуються:

- ключові слова (наприклад: «зоомагазин», «товари для тварин», «корм для собак», «аксесуари для котів»);
- правильна структура заголовків (H1–H3);
- мета-теги (title, description);
- оптимізовані URL-адреси сторінок;
- оптимізація швидкості завантаження сайту;
- адаптивна верстка для мобільних пристроїв.

Просування у соціальних мережах

Для залучення нових клієнтів вебсайт може просуватися через Instagram, Facebook та TikTok. У соціальних мережах публікуються новини магазину, інформація про акції, нові товари та рекомендації щодо догляду за домашніми тваринами.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

Оновлення контенту

Регулярне оновлення каталогу товарів, фотографій, описів та акційних пропозицій позитивно впливає на активність користувачів і підтримує актуальність вебресурсу.

Реклама та аналітика

Для збільшення кількості відвідувачів можуть використовуватися рекламні кампанії Google Ads та Meta Ads. Також доцільно використовувати сервіси аналітики для відстеження поведінки користувачів, популярності товарів та ефективності рекламних кампаній.

Використання сучасних методів просування дозволяє підвищити популярність вебсайту, збільшити кількість користувачів та забезпечити ефективний розвиток інтернет-магазину «ЛАПКА».

Соціальні мережі

Просування вебсайту виконується через популярні соціальні мережі Instagram, Facebook та TikTok.

Використання соціальних платформ дозволяє привертати увагу потенційних покупців, інформувати користувачів про нові товари та підтримувати постійний зв'язок із клієнтами.

У соціальних мережах можуть публікуватися:

- інформація про нові товари та акції;
- фотографії товарів для домашніх тварин;
- короткі відеоогляди продукції;
- посилання на вебсайт;
- відгуки покупців.

Контент-маркетинг

Регулярне оновлення інформації на сайті позитивно впливає на популярність вебсайту та його позиції у пошукових системах. Для цього можуть використовуватися новини, інформаційні статті та рекомендації щодо догляду за домашніми тваринами.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

Контент-маркетинг дозволяє:

- підвищити довіру користувачів;
- покращити видимість сайту в Google;
- збільшити кількість відвідувачів;
- підтримувати актуальність вебресурсу.

Підтримка вебсайту

Технічна підтримка системи включає:

- оновлення інформації про товари;
- перевірку працездатності форм замовлення;
- оновлення контактної інформації;
- контроль коректного відображення сайту на різних пристроях.

Також рекомендується періодично перевіряти:

- швидкість завантаження сторінок;
- працездатність навігаційних посилань;
- коректність роботи вебсайту у різних браузерах.

Таким чином, спеціальний розділ описує повний процес впровадження та підтримки вебсайту зоомагазину «ЛАПКА» — від розміщення в мережі Інтернет до подальшого супроводу та популяризації ресурсу. Реалізовані механізми забезпечують стабільну роботу системи, зручність адміністрування та ефективну взаємодію з користувачами.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

4. ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини даного дипломного проєкту є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки вебсайту зоомагазину «ЛАПКА», прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є вебсайт зоомагазину «ЛАПКА», призначений для представлення асортименту товарів для домашніх тварин, надання інформації про продукцію, спрощення процесу оформлення замовлень та забезпечення зручної взаємодії між покупцями і магазином. Вебсайт надає можливість перегляду каталогу товарів, пошуку необхідної продукції, отримання актуальної інформації про наявність товарів та оформлення замовлень через мережу Інтернет.

Розрахунок вартості розробки виконується в декілька етапів:

- опис технологічного процесу розробки із зазначенням трудомісткості кожної операції;
- визначення суми витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислення витрат на електроенергію;
- нарахування суми амортизаційних відрахувань;
- визначення суми накладних витрат;
- складання кошторису та визначення собівартості робіт;
- розрахунок ціни робіт;
- визначення економічної ефективності та терміну окупності проєкту.

4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки вебсайту зоомагазину «ЛАПКА». Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

процесу звести у таблицю. Для визначення загальної тривалості проведення робіт дані щодо витрат часу наведені в таблиці 4.1.

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 Рm	9
		Інженер (І1)	9
2	Розробка технічного завдання	Кер. проєкту (Рm)	8
		Інженер (І1)	5
3	Дизайн інтерфейсу	Інженер (І1)	18
		Інженер (І2)	24
4	Розробка функціоналу	Інженер (І1)	46
5	Тестування та відладка	Тестувальник	10
6	Документування	Інженер (І1)	3
7	Розгортання та підтримка	Інженер (І2)	18
8	Управління проєктом	Кер. проєкту (Рm)	20
Разом			

Сумарний час виконання операцій технологічного процесу становить 171 години.

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 вебсайту зоомагазину «ЛАПКА».

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

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:

$$Z_{\text{осн.}} = T_c * K_T \quad (4.1)$$

де: T_c – тарифна ставка, грн. (приймаємо для керівника проекту (Pm) – 600 грн./год, інженера (I2) – 420 грн./год., інженера (I1) – 260 грн./год., тестувальник – 220 грн./год.; K_T – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

Керівника проекту (Pm) $Z_{\text{осн.}} = 37 * 600 = 22\,200$ грн.

Інженера (I2) $Z_{\text{осн.}} = 43 * 420 = 18\,060$ грн.

Інженера (I1) $Z_{\text{осн.}} = 81 * 260 = 21\,060$ грн.

Тестувальник $Z_{\text{осн.}} = 10 * 220 = 2\,200$ грн.

Сумарна основна заробітна плата становить

$$Z_{\text{осн.}} = 22\,200 + 18\,060 + 21\,060 + 2\,200 = 63\,520 \text{ грн.}$$

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

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

де: $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

Керівника проекту $Z_{\text{дод.2}} = 22\,200 * 0,1 = 2\,220$ грн.

Інженера (I2) $Z_{\text{дод.3}} = 18\,060 * 0,1 = 1\,806$ грн.

Інженера (I1) $Z_{\text{дод.4}} = 21\,060 * 0,1 = 2\,106$ грн.

Тестувальник $Z_{\text{дод.4}} = 2\,200 * 0,1 = 220$ грн.

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод.}} = 2\,220 + 1\,806 + 2\,106 + 220 = 6\,352 \text{ грн.}$$

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

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

$$B_{\text{о.п.}} = 63\,520 + 6\,352 = 69\,872 \text{ грн.}$$

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

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

$$V_{\text{ЄСВ}} = V_{\text{оп}} * 0,22 \quad (4.4)$$

$$V_{\text{ЄСВ}} = 69\,872 * 0,22 = 15\,372 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 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. 6 = 3+4+5
		Тарифна ставка, грн.	К-сть годин	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	ер. проєкту (Pm)	600	37	22 000	2 220	5 372	29 792
2	Інженера (І2)	420	43	18 060	1 806	4 371	24 237
3	Інженера (І1)	260	81	22 060	2 106	5 096	28 262
4	Тестувальник	220	10	2 200	220	532	2 952
Разом				63 520	6 352	15 372	85 243

Отже, загальні витрати на оплату праці становлять **85 243 грн.**

4.3. Розрахунок витрат на електроенергію

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

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

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

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні: комп'ютер – 0,82 кВт/год.

$$Z_{ек} = 0,82 * 171 * 15,94 = 2\,235 \text{ грн.}$$

Витрати на електроенергію становлять 2 235 грн.

4.4. Розрахунок суми амортизаційних відрахувань вебсайту зоомагазину «ЛАПКА»

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 амортизаційних відрахувань застосовуємо формулу:

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

де: А – амортизаційні відрахування за звітний період, грн.;

Б_В – балансова вартість групи основних фондів на початок звітного періоду, грн.;

Н_А – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ПК, вартістю 50000,00 грн., то сума амортизаційних відрахувань становитиме:

$$A = 50\,000,00 * 0,04 * 171 = 2\,280 \text{ грн.}$$

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

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.

$$H_B = V_{o.p.} * 0,2..0,6 \quad (4.7)$$

де: H_B – накладні витрати.

$$H_B = 69\,872 * 0,4 = 27\,949 \text{ грн.}$$

4.6. Складання кошторису витрат та визначення собівартості вебсайту зоомагазину «ЛАПКА»

Для складання кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.4.

Таблиця 4.4 - Кошторис витрат Кошторис витрат на розробку вебсайту зоомагазину «ЛАПКА»

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	85 243	72
2.	Витрати на електроенергію	2 235	2
3.	Амортизаційні відрахування	2 280	2
4.	Накладні витрати	27 949	24
5.	Собівартість	117 707	100

Собівартість (C_B) НДР розраховуємо за формулою:

$$C_B = V_{o.p.} + V_{c.z} + 3e + A + H_B \quad (4.8)$$

Отже, собівартість дорівнює $C_B = 69\,872 + 15\,371 + 2\,235 + 2\,280 + 27\,949 = 117\,707$ грн.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

4.7. Розрахунок ціни вебсайту зоомагазину «ЛАПКА»

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:

$$Ц = C_{\text{в}} * (1 + P_{\text{рен}}) * (1 + \text{ПДВ}), \quad (4.9)$$

де: $C_{\text{в}}$ – собівартість; $P_{\text{рен}}$ – рівень рентабельності; ПДВ – ставка податку на додану вартість.

$$Ц = 117\,707 * (1 + 0,3) * (1 + 0,2) = 183\,623 \text{ грн.}$$

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.

$$Г_{\text{п}} = Ц - C_{\text{в}} = 110\,660 - 70\,936 = 39\,724 \text{ грн.} \quad (4.10)$$

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$\text{ЧТВ} = -C_{\text{в}} + \sum_{t=1}^n \frac{Г_{\text{п}}}{(1+i)^t}, \quad (4.11)$$

де: $C_{\text{в}}$ – собівартість розробки; $Г_{\text{п}}$ – грошовий потік за t – ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$\text{ЧТВ} = -117\,707 + \frac{65\,916}{(1+0,1)^1} + \frac{65\,916}{(1+0,1)^2} + \frac{65\,916}{(1+0,1)^3} = 46\,252 \text{ грн.}$$

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

$$T_{\text{ок}} = T_{\text{пв}} + \frac{H_{\text{в}}}{Г_{\text{пР}}} \quad (4.12)$$

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

де: $T_{пв}$ – період до повного відшкодування витрат, років; $H_{в}$ – невідшкодовані витрати на початок року, грн.; $\Gamma_{пр}$ – грошовий потік на початок року, грн.

3 308

$$T_{ок} = 2 + \frac{65\,916}{3\,308} = 2,1 \text{ р.}$$

Всі дані внесемо в зведену таблицю 4.5.

Таблиця 4.5 – Техніко-економічні показники вебсайту зоомагазину «ЛІАПКА»

№ п/п	Показник	Значення
1.	Собівартість, грн.	117 707
2.	Плановий прибуток або грошовий потік, грн.	65 916
3.	Ціна, грн.	183 623
4.	Чиста тепершня вартість, грн.	46 252
5.	Термін окупності, рік	2,1

Прибутковість проєкту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом 2,1 року. Отже, на основі отриманих показників можна зробити висновок, що розробка вебсайту зоомагазину «ЛІАПКА» є доцільною з економічної точки зору.

Отже, розробка вебсайту зоомагазину «ЛІАПКА» є економічно доцільною. Отримані показники свідчать про позитивний економічний ефект від впровадження проєкту та його окупність протягом відносно короткого періоду часу

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Інструкції з охорони праці, їх поділ

Охорона праці є важливою складовою організації безпечної трудової діяльності працівників на підприємстві. Вона охоплює систему правових, організаційних, технічних, санітарно-гігієнічних та профілактичних заходів, спрямованих на збереження життя, здоров'я та працездатності людини під час виконання професійних обов'язків. У сфері інформаційних технологій та веброзробки дотримання правил охорони праці також має велике значення, оскільки працівники тривалий час працюють за комп'ютерною технікою, електрообладнанням та іншими цифровими пристроями.

Основним документом, який регламентує безпечне виконання робіт, є інструкція з охорони праці. Вона містить перелік вимог, правил і рекомендацій щодо безпечної поведінки працівника під час виконання службових обов'язків, користування обладнанням та перебування на робочому місці.

Інструкції з охорони праці поділяються на кілька основних видів:

- інструкції за професіями або посадами;
- інструкції за видами робіт;
- тимчасові інструкції;
- типові інструкції;
- локальні інструкції підприємства.

Інструкції за професіями або посадами розробляються окремо для кожної категорії працівників. Наприклад, для системного адміністратора, програміста або веброзробника. У них визначаються вимоги безпеки перед початком роботи, під час роботи та після її завершення.

Інструкції за видами робіт стосуються виконання певних робіт або використання конкретного обладнання. До них можуть належати інструкції під час

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

роботи з електрообладнанням, офісною технікою або мережевими пристроями.

Тимчасові інструкції вводяться на певний період часу у випадку впровадження нового обладнання або зміни технологічного процесу.

Типові інструкції створюються на державному або галузевому рівні та використовуються як основа для розробки локальних інструкцій на підприємстві.

Локальні інструкції розробляються безпосередньо організацією з урахуванням особливостей робочого середовища та специфіки діяльності працівників.

Важливим елементом системи охорони праці є проведення інструктажів. На підприємствах проводяться вступний, первинний, повторний, позаплановий та цільовий інструктажі. Їх метою є ознайомлення працівників із правилами безпеки, попередження нещасних випадків та формування навичок правильної поведінки у надзвичайних ситуаціях.

Таким чином, інструкції з охорони праці є невід'ємною частиною організації безпечної трудової діяльності та допомагають запобігати виробничим травмам.

5.2 Системи засобів і заходів забезпечення електробезпеки

Електробезпека є одним із найважливіших напрямів охорони праці, оскільки електричний струм становить серйозну небезпеку для життя та здоров'я людини. У сучасних умовах практично всі підприємства та офіси оснащені електрообладнанням, комп'ютерною технікою та мережевими пристроями.

Основними причинами ураження електричним струмом є:

- порушення правил експлуатації електрообладнання;
- використання несправних електроприладів;
- пошкодження ізоляції проводів;
- відсутність заземлення;
- недотримання правил техніки безпеки;
- робота з електрообладнанням у вологих умовах.

Системи засобів і заходів забезпечення електробезпеки включають комплекс

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

організаційних та технічних рішень, спрямованих на запобігання ураженню людини електричним струмом під час експлуатації електрообладнання.

До основних засобів забезпечення електробезпеки належать:

- ізоляція струмопровідних частин обладнання;
- використання захисного заземлення;
- застосування автоматичних вимикачів та запобіжників;
- використання пристроїв захисного вимкнення;
- застосування безпечної напруги;
- огороження небезпечних зон;
- використання засобів індивідуального захисту;
- проведення інструктажів та навчання працівників.

Одним із найефективніших способів захисту є захисне заземлення. У разі пошкодження ізоляції або появи напруги на корпусі обладнання електричний струм відводиться в землю, що допомагає зменшити ризик ураження людини електричним струмом.

Також важливу роль відіграють автоматичні системи захисту, які миттєво відключають електроживлення при короткому замиканні або перевантаженні мережі.

У приміщеннях, де використовується комп'ютерна техніка та серверне обладнання, необхідно регулярно проводити перевірку стану електропроводки, контролювати справність розеток і кабелів та не допускати перевантаження електромережі.

Важливим заходом забезпечення електробезпеки є проведення інструктажів з охорони праці та навчання персоналу правилам безпечної експлуатації електрообладнання.

Отже, дотримання правил електробезпеки, проведення інструктажів та контроль технічного стану електромереж дозволяють значно знизити ризик виникнення нещасних випадків і забезпечити безпечні умови праці.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено web-застосунок інтернет-магазину товарів для домашніх тварин «ЛАПКА». Під час виконання роботи проведено аналіз предметної області, визначено функціональні вимоги до системи та спроектовано структуру web-застосунку.

У процесі розробки створено адаптивний інтерфейс користувача, реалізовано каталог товарів, сторінку окремого товару, кошик покупок, систему авторизації, особистий кабінет користувача та адміністративну панель. Для реалізації клієнтської частини використано React, Vite та Tailwind CSS, а серверну частину побудовано на основі Node.js, Express.js та MongoDB Atlas.

У системі реалізовано REST API для взаємодії між frontend- та backend-частинами, JWT-авторизацію, механізми захисту адміністративних маршрутів і роботу з базою даних через Mongoose. Для зберігання фотографій товарів використано хмарний сервіс Cloudinary.

Особливу увагу приділено адаптивності інтерфейсу та зручності користування web-застосунком на різних типах пристроїв. Проведене тестування підтвердило коректність роботи основних функцій системи, навігації, форм, API-запитів, адміністративної панелі та механізмів авторизації.

За результатами виконаної роботи встановлено, що розроблений web-застосунок відповідає поставленим вимогам, працює стабільно та може використовуватися як сучасний інтернет-магазин товарів для домашніх тварин із можливістю подальшого розвитку та розширення функціоналу.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

ПЕРЕЛІК ПОСИЛАНЬ

1) Марціяш Г.Я., Слободян Р.О. Методичні вказівки до виконання кваліфікаційної роботи для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки». Тернопіль : ВСП «ТФК ТНТУ ім. І. Пулюя», 2026, 48 с. Java Persistence з Hibernate - Крістіан Бауер, Гевін Кінг та Гарі Грегорі. Manning Publications, 2007 (дата звернення: 15.10.2024).

2) PostgreSQL: Практичний вступ - Реджина О. Об та Лео С. Хсу. O'Reilly Media, 2014 (дата звернення 12.11.2024).

3) Render Documentation. Render. URL: <https://render.com/docs> (дата звернення: 10.06.2026)

4) Java Hashing and Hashmaps - Baeldung : вебсайт. URL: <https://www.baeldung.com/java-hashmap> (дата звернення: 1.12.2024).

5) Vercel Documentation. Vercel. URL: <https://vercel.com/docs> (дата звернення: 10.06.2026).

6) Вступ до Java Persistence API (JPA) - Oracle: вебсайт. URL: <https://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html> (дата звернення: 20.11.2024).

7) Apache Maven - Apache Maven : вебсайт. URL: <https://maven.apache.org/> (дата звернення: 10.12.2024).

8) React : вебсайт. URL: <https://react.dev/> (дата звернення: 14.06.2026).

9) MongoDB: вебсайт. URL: <https://www.mongodb.com/docs/> (дата звернення: 14.06.2026).

10) Node.js Documentation : вебсайт. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 14.06.2026).

11) Express.js: вебсайт. URL: <https://expressjs.com/> (дата звернення: 14.06.2026).

12) JSON Web Tokens Introduction : вебсайт. URL: <https://jwt.io/introduction> (дата звернення: 14.06.2026).

13) Cloudinary Documentation : вебсайт. URL: <https://cloudinary.com/documentation> (дата звернення: 14.06.2026).

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

14) Tailwind CSS Documentation : вебсайт. URL
<https://tailwindcss.com/docs> (дата звернення: 14.06.2026)

15) Freeman E., Robson E. Head First Design Patterns. Sebastopol : O'Reilly Media, 2020. 694 p.

16) Flanagan D. JavaScript: The Definitive Guide. 7th ed. Sebastopol: O'Reilly Media, 2020. 704 p.

17) Osmani A. Learning JavaScript Design Patterns. Sebastopol : O'Reilly Media, 2023. 254 p.

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

ДОДАТКИ

Додаток А. App.jsx

```
import { Routes, Route } from 'react-router-dom';
import Navbar from './components/Navbar';
import Footer from './components/Footer';
import HomePage from './pages/HomePage';
import CatalogPage from './pages/CatalogPage';
import ProductDetailsPage from './pages/ProductDetailsPage';
import CartPage from './pages/CartPage';
import LoginPage from './pages/LoginPage';
import RegisterPage from './pages/RegisterPage';
import AdminPage from './pages/AdminPage';
import AboutPage from './pages/AboutPage';
import ProtectedRoute from './components/ProtectedRoute';

function App() {
  return (
    <div className="min-h-screen flex flex-col">
      <Navbar />
      <main className="flex-grow">
        <Routes>
          <Route path="/" element={<HomePage />} />
          <Route path="/catalog" element={<CatalogPage />} />
          <Route path="/about" element={<AboutPage />} />
          <Route path="/products/:id" element={<ProductDetailsPage />} />
          <Route path="/cart" element={<CartPage />} />
          <Route path="/login" element={<LoginPage />} />
          <Route path="/register" element={<RegisterPage />} />
          <Route path="/admin" element={<ProtectedRoute role="admin"><AdminPage
/></ProtectedRoute>} />
        </Routes>
      </main>
      <Footer />
    </div>
  );
}

export default App;
```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

Додаток Б. main.jsx

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import './index.css';
import { AuthProvider } from './context/AuthContext';
import { CartProvider } from './context/CartContext';
import { BrowserRouter } from 'react-router-dom';

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <BrowserRouter future={{ v7_startTransition: true, v7_relativeSplatPath: true }}>
      <AuthProvider>
        <CartProvider>
          <App />
        </CartProvider>
      </AuthProvider>
    </BrowserRouter>
  </React.StrictMode>
);
```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

Додаток В. HomePage.jsx

```
import HeroSection from '../components/HeroSection';
import CategoryCard from '../components/CategoryCard';
import ProductCard from '../components/ProductCard';
import BenefitsSection from '../components/BenefitsSection';
import TestimonialsSection from '../components/TestimonialsSection';
import NewsletterSection from '../components/NewsletterSection';
import { useEffect, useState } from 'react';
import api from '../utils/api';
import { motion } from 'framer-motion';

const categories = ['Food', 'Toys', 'Eco-Friendly', 'Accessories', 'Health', 'Grooming'];

const HomePage = () => {
  const [featuredProducts, setFeaturedProducts] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    const fetchFeaturedProducts = async () => {
      try {
        const res = await api.get('/api/v1/products', {
          params: {
            limit: 8,
            ecoFriendly: true,
          },
        });
        setFeaturedProducts(res.data.data);
      } catch (err) {
        setError('Не вдалося завантажити рекомендовані товари');
        console.error(err);
      } finally {
        setLoading(false);
      }
    };

    fetchFeaturedProducts();
  }, []);

  const containerVariants = {
    hidden: { opacity: 0 },
    visible: {
```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

```

    opacity: 1,
    transition: { staggerChildren: 0.08 }
  }
};

const itemVariants = {
  hidden: { opacity: 0, y: 20 },
  visible: { opacity: 1, y: 0, transition: { duration: 0.5 } }
};

return (
  <div className="min-h-screen bg-gradient-to-b from-white via-green-50/30 to-
white dark:from-gray-900 dark:via-gray-800 dark:to-gray-900 overflow-hidden">
    {/* Decorative background elements */}
    <div className="fixed inset-0 pointer-events-none overflow-hidden">
      <div className="absolute top-0 right-0 w-96 h-96 bg-green-200/10 rounded-full
blur-3xl -z-10" />
      <div className="absolute bottom-0 left-0 w-96 h-96 bg-blue-200/10 rounded-
full blur-3xl -z-10" />
    </div>

    {/* Hero Section */}
    <HeroSection />

    {/* Categories Section */}
    <section className="py-20 relative">
      <div className="container mx-auto px-4">
        <motion.div
          initial={{ opacity: 0, y: 20 }}
          whileInView={{ opacity: 1, y: 0 }}
          transition={{ duration: 0.5 }}
          className="text-center mb-16"
        >
          <h2 className="text-4xl md:text-5xl font-extrabold text-gray-900 dark:text-
white mb-4">
            Обрати за категорією
          </h2>
          <p className="text-lg text-gray-600 dark:text-gray-300 max-w-2xl mx-
auto">
            Знайдіть товари відповідно до потреб вашого улюбленця
          </p>
        </motion.div>

        <motion.div

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

```

variants={containerVariants}
initial="hidden"
whileInView="visible"
viewport={{ once: true, margin: "-100px" }}
className="grid grid-cols-2 md:grid-cols-3 lg:grid-cols-6 gap-4"
>
  {categories.map((category) => (
    <motion.div key={category} variants={itemVariants}>
      <CategoryCard category={category} />
    </motion.div>
  ))}
</motion.div>
</div>
</section>

{/* Featured Products Section */}
<section className="py-20 bg-white/50 dark:bg-gray-800/50 backdrop-blur-sm
relative">
  <div className="container mx-auto px-4">
    <motion.div
      initial={{ opacity: 0, y: 20 }}
      whileInView={{ opacity: 1, y: 0 }}
      transition={{ duration: 0.5 }}
      className="text-center mb-16"
    >
      <h2 className="text-4xl md:text-5xl font-extrabold text-gray-900 dark:text-
white mb-4">
        Рекомендовані еко-товари
      </h2>
      <p className="text-lg text-gray-600 dark:text-gray-300 max-w-2xl mx-
auto">
        Вибір преміальних товарів для щасливих улюбленців
      </p>
    </motion.div>

    {loading ? (
      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6">
        {Array.from({ length: 8 }).map((_, i) => (
          <div key={i} className="h-96 bg-gray-200 dark:bg-gray-700 rounded-
3xl animate-pulse" />
        ))}
      </div>
    ) : error ? (

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

```

    <motion.div initial={{ opacity: 0 }} animate={{ opacity: 1 }}
className="text-center py-12">
    <p className="text-lg text-red-600 dark:text-red-400">{error}</p>
    </motion.div>
  ): (
    <motion.div
      variants={containerVariants}
      initial="hidden"
      whileInView="visible"
      viewport={{ once: true, margin: "-100px" }}
      className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6"
    >
      {featuredProducts.map((product) => (
        <motion.div key={product._id} variants={itemVariants}>
          <ProductCard product={product} />
        </motion.div>
      ))}
    </motion.div>
  )}
</div>
</section>

{/* Benefits Section */}
<BenefitsSection />

{/* Testimonials Section */}
<TestimonialsSection />

{/* Newsletter Section */}
<NewsletterSection />
</div>
);
};

export default HomePage;

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

Додаток Г. CatalogPage.jsx

```
import { useEffect, useMemo, useState } from 'react';
import { useSearchParams } from 'react-router-dom';
import api from '../utils/api';
import { AnimatePresence, motion } from 'framer-motion';
import { FaBoxOpen, FaFilter, FaLeaf, FaSearch, FaTimes } from 'react-icons/fa';
import ProductCard from '../components/ProductCard';
import { categories, categoryNames, emptyFilters, filterConfig } from
'../config/productFilters';

const PRICE_LIMIT = 2500;

const SkeletonCard = () => (
  <div className="overflow-hidden rounded-3xl border border-white/70 bg-white/80
shadow-soft">
    <div className="skeleton-loader h-64 rounded-none" />
    <div className="space-y-4 p-5">
      <div className="skeleton-loader h-5 w-4/5" />
      <div className="skeleton-loader h-4 w-2/5" />
      <div className="skeleton-loader h-10 w-full" />
    </div>
  </div>
);

const CheckOption = ({ checked, label, onChange }) => (
  <label className="group flex cursor-pointer items-center justify-between gap-3
rounded-xl px-2 py-2 text-sm text-slate-600 transition hover:bg-emerald-50 hover:text-
slate-900">
    <span>{label}</span>
    <input type="checkbox" checked={checked} onChange={onChange}
className="filter-checkbox" />
  </label>
);

const FilterPanel = ({ category, filters, activeCount, onToggle, onChange, onClear,
onClose }) => {
  const dynamicFilters = filterConfig[category] || [];

  return (
    <div className="h-full overflow-y-auto rounded-3xl border border-white/70 bg-
white/80 p-5 shadow-large backdrop-blur-xl">
      <div className="mb-6 flex items-center justify-between">
```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

```

<div>
  <p className="text-xs font-bold uppercase tracking-[0.2em] text-
primary">Параметри</p>
  <h2 className="mt-1 text-xl font-bold text-slate-900">Фільтри {activeCount
> 0 && `(${activeCount})`</h2>
</div>
{onClose && (
  <button onClick={onClose} className="rounded-full bg-slate-100 p-3 text-
slate-500" aria-label="Закрити фільтри">
    <FaTimes />
  </button>
)}
</div>

```

```

{dynamicFilters.map((group) => (
  <div key={group.key} className="mb-5 border-b border-slate-100 pb-5">
    <h3 className="mb-2 text-sm font-bold text-slate-900">{group.label}</h3>
    {group.options.map((option) => (
      <CheckOption
        key={typeof option === 'string' ? option : option.value}
        label={typeof option === 'string' ? option : option.label}
        checked={filters[group.key].includes(typeof option === 'string' ? option :
option.value)}
        onChange={() => onToggle(group.key, typeof option === 'string' ? option :
option.value)}
      />
    ))}
  </div>
))}

```

```

<div className="mb-5 border-b border-slate-100 pb-5">
  <div className="mb-3 flex items-center justify-between text-sm font-bold text-
slate-900">
    <span>Ціна</span>
    <span className="text-primary">{filters.minPrice} - {filters.maxPrice}
€</span>
  </div>
  <label className="mb-3 block text-xs text-slate-500">Мінімальна ціна</label>
  <input type="range" min="0" max={PRICE_LIMIT} step="50"
value={filters.minPrice} onChange={(event) => onChange('minPrice',
Number(event.target.value))} className="price-slider" />
  <label className="mb-3 mt-4 block text-xs text-slate-500">Максимальна
ціна</label>

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

```

    <input type="range" min="0" max={PRICE_LIMIT} step="50"
value={filters.maxPrice} onChange={(event) => onChange('maxPrice',
Number(event.target.value))} className="price-slider" />
  </div>

```

```

<div className="mb-5 border-b border-slate-100 pb-5">
  <h3 className="mb-3 text-sm font-bold text-slate-900">Рейтинг</h3>
  {[
    ['4', '★★★★☆ і вище'],
    ['3', '★★★☆☆ і вище'],
  ].map(([value, label]) => (
    <CheckOption key={value} label={label} checked={filters.rating === value}
onChange={() => onChange('rating', filters.rating === value ? '' : value)} />
  )))
</div>

```

```

<div className="space-y-1">
  <CheckOption label="В наявності" checked={filters.inStock} onChange={() =>
onChange('inStock', !filters.inStock)} />
  <CheckOption label="Тільки еко товари" checked={filters.ecoFriendly}
onChange={() => onChange('ecoFriendly', !filters.ecoFriendly)} />
</div>

```

```

<button onClick={onClear} className="mt-6 w-full rounded-2xl border border-
slate-200 px-4 py-3 text-sm font-bold text-slate-600 transition hover:border-primary
hover:text-primary">
  Очистити фільтри
</button>
</div>

```

```

);
};

```

```

const CatalogPage = () => {
  const [searchParams, setSearchParams] = useSearchParams();
  const [products, setProducts] = useState([]);
  const [count, setCount] = useState(0);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState("");
  const [drawerOpen, setDrawerOpen] = useState(false);
  const [search, setSearch] = useState(searchParams.get('search') || "");
  const [category, setCategory] = useState(searchParams.get('category') || 'All');
  const [filters, setFilters] = useState(() => ({
    ...emptyFilters,
    ecoFriendly: searchParams.get('ecoFriendly') === 'true',

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

```

    ));

    useEffect(() => {
      const nextCategory = searchParams.get('category') || 'All';
      const nextSearch = searchParams.get('search') || "";
      const nextEcoFriendly = searchParams.get('ecoFriendly') === 'true';

      setCategory((current) => (current === nextCategory ? current : nextCategory));
      setSearch((current) => (current === nextSearch ? current : nextSearch));
      setFilters((current) => (
        current.ecoFriendly === nextEcoFriendly
          ? current
          : { ...current, ecoFriendly: nextEcoFriendly }
      ));
    }, [searchParams]);

    const requestParams = useMemo(() => {
      const params = {};
      if (search.trim()) params.search = search.trim();
      if (category !== 'All') params.category = category;
      Object.entries(filters).forEach(([key, value]) => {
        if (Array.isArray(value) && value.length) params[key] = value.join(',');
        if (!Array.isArray(value) && value !== "" && value !== false) params[key] =
value;
      });
      return params;
    }, [category, filters, search]);

    const activeCount = useMemo(() => Object.entries(filters).reduce((total, [key, value])
=> {
      if (Array.isArray(value)) return total + value.length;
      if (key === 'minPrice') return total + (value > 0 ? 1 : 0);
      if (key === 'maxPrice') return total + (value < PRICE_LIMIT ? 1 : 0);
      return total + (value ? 1 : 0);
    }, 0), [filters]);

    useEffect(() => {
      const controller = new AbortController();
      const timer = setTimeout(async () => {
        try {
          setLoading(true);
          setError("");
          const response = await api.get('/api/v1/products', {
            params: requestParams,

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

```

    signal: controller.signal,
  });
  setProducts(response.data.data || []);
  setCount(response.data.count || 0);
  setSearchParams(requestParams, { replace: true });
} catch (err) {
  if (controller.signal.aborted) return;
  setError('Не вдалося завантажити товари. Спробуйте ще раз.');
```

console.error(err.response?.data || err.message);

```

} finally {
  if (!controller.signal.aborted) setLoading(false);
}
}, 180);
return () => {
  clearTimeout(timer);
  controller.abort();
};
}, [requestParams, setSearchParams]);
```

```

const handleCategoryChange = (nextCategory) => {
  setCategory(nextCategory);
  setFilters((current) => ({
    ...emptyFilters,
    minPrice: current.minPrice,
    maxPrice: current.maxPrice,
    rating: current.rating,
    inStock: current.inStock,
    ecoFriendly: current.ecoFriendly,
  }));
};
```

```

const handleToggle = (key, value) => {
  setFilters((current) => ({
    ...current,
    [key]: current[key].includes(value)
      ? current[key].filter((item) => item !== value)
      : [...current[key], value],
  }));
};
```

```

const handleChange = (key, value) => {
  setFilters((current) => {
    if (key === 'minPrice') return { ...current, minPrice: Math.min(value,
current.maxPrice) };
  });
};
```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

```

    if (key === 'maxPrice') return { ...current, maxPrice: Math.max(value,
current.minPrice) };
    return { ...current, [key]: value };
  });
};

const clearFilters = () => {
  setSearch("");
  setCategory('All');
  setFilters({ ...emptyFilters });
};

return (
  <div className="min-h-screen bg-gradient-to-br from-emerald-50 via-white to-
amber-50/50">
    <div className="container mx-auto px-4 py-10">
      <div className="mb-8 rounded-4xl bg-gradient-to-r from-primary to-secondary
px-6 py-8 text-white shadow-large md:px-10">
        <p className="mb-3 flex items-center gap-2 text-xs font-bold uppercase
tracking-[0.24em] text-emerald-100"><FaLeaf /> Каталог EcoPetShop</p>
        <h1 className="text-4xl font-extrabold md:text-5xl">Знайдіть найкраще для
улюбленця</h1>
        <p className="mt-3 max-w-2xl text-emerald-50">Добірні товари, зручні
категорії та точні фільтри для швидкого вибору.</p>
        <div className="relative mt-6 max-w-2xl">
          <FaSearch className="absolute left-4 top-1/2 -translate-y-1/2 text-emerald-
700" />
          <input value={search} onChange={(event) => setSearch(event.target.value)}
placeholder="Пошук за назвою або описом..." className="w-full rounded-2xl bg-
white px-12 py-4 text-sm text-slate-900 outline-none ring-4 ring-white/20
placeholder:text-slate-400 focus:ring-white/40" />
        </div>
      </div>

      <div className="mb-7 flex gap-2 overflow-x-auto pb-2">
        {categories.map((item) => (
          <button key={item} onClick={() => handleCategoryChange(item)}
className={`whitespace-nowrap rounded-full px-5 py-3 text-sm font-bold transition
${category === item ? 'bg-primary text-white shadow-medium' : 'bg-white text-slate-
600 shadow-soft hover:text-primary'}`}>
            {categoryNames[item]}
          </button>
        ))}
      </div>
    </div>
  </div>
);

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

```

<div className="mb-5 flex items-center justify-between lg:pl-[19rem]">
  <p className="text-sm font-semibold text-slate-500">Знайдено: <span
className="text-slate-900">{count}</span></p>
  <button onClick={() => setDrawerOpen(true)} className="flex items-center
gap-2 rounded-2xl bg-white px-4 py-3 text-sm font-bold text-primary shadow-soft
lg:hidden">
    <FaFilter /> Фільтри {activeCount > 0 && `(${activeCount})`}
  </button>
</div>

<div className="grid gap-7 lg:grid-cols-[17rem_1fr]">
  <aside className="hidden lg:block">
    <FilterPanel category={category} filters={filters} activeCount={activeCount}
onToggle={handleToggle} onChange={handleChange} onClear={clearFilters} />
  </aside>

  <main>
    {loading && products.length === 0 ? (
      <div className="grid gap-5 sm:grid-cols-2 xl:grid-cols-3">
        {Array.from({ length: 6 }).map((_, index) => <SkeletonCard key={index}
/>)}
      </div>
    ) : error ? (
      <div className="rounded-3xl bg-white p-12 text-center text-red-500
shadow-soft">{error}</div>
    ) : products.length ? (
      <motion.div initial="hidden" animate="visible" variants={{ visible: {
transition: { staggerChildren: 0.04 } } }} className={`grid gap-5 transition-opacity
duration-200 sm:grid-cols-2 xl:grid-cols-3 ${loading ? 'opacity-60' : 'opacity-100'}`>
        {products.map((product) => (
          <motion.div key={product._id} variants={{ hidden: { opacity: 0, y: 16 },
visible: { opacity: 1, y: 0 } }}>
            <ProductCard product={product} />
          </motion.div>
        ))}
      </motion.div>
    ) : (
      <div className="rounded-3xl border border-white bg-white/80 p-12 text-
center shadow-soft">
        <FaBoxOpen className="mx-auto mb-4 text-6xl text-emerald-200" />
        <h2 className="text-2xl font-bold text-slate-900">Товарів не
знайдено</h2>

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

```

    <p className="mt-2 text-sm text-slate-500">Спробуйте змінити
    параметри або очистити фільтри.</p>
    <button onClick={clearFilters} className="btn-primary mt-6">Показати
    всі товари</button>
  </div>
)}
</main>
</div>
</div>

```

```

<AnimatePresence>
  {drawerOpen && (
    <motion.div initial={{ opacity: 0 }} animate={{ opacity: 1 }} exit={{ opacity:
    0 }} className="fixed inset-0 z-50 bg-slate-900/40 backdrop-blur-sm lg:hidden"
    onClick={() => setDrawerOpen(false)}>
      <motion.div initial={{ x: '-100%' }} animate={{ x: 0 }} exit={{ x: '-100%' }}
      transition={{ type: 'spring', damping: 28 }} className="h-full w-[88%] max-w-sm p-
      3" onClick={(event) => event.stopPropagation()}>
        <FilterPanel category={category} filters={filters}
        activeCount={activeCount} onToggle={handleToggle} onChange={handleChange}
        onClear={clearFilters} onClose={() => setDrawerOpen(false)} />
      </motion.div>
    </motion.div>
  )}
</AnimatePresence>
</div>
);
};

```

```
export default CatalogPage;
```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		76

Додаток Д. ProductDetailsPage .jsx

```
import { useState, useEffect } from 'react';
import { useParams, Link } from 'react-router-dom';
import api from '../utils/api';
import { useCart } from '../context/CartContext';
import { FaCartPlus, FaLeaf, FaStar, FaStarHalfAlt, FaRegStar } from 'react-icons/fa';
import { getProductDescription, getProductLabel, getProductname } from
'../config/productTranslations';
import OptimizedImage from '../components/OptimizedImage';

const ProductDetailsPage = () => {
  const { id } = useParams();
  const [product, setProduct] = useState(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const [quantity, setQuantity] = useState(1);
  const { addToCart } = useCart();

  useEffect(() => {
    const fetchProduct = async () => {
      try {
        const res = await api.get(`/api/v1/products/${id}`);
        setProduct(res.data.data);
      } catch (err) {
        setError('Не вдалося завантажити деталі товару');
        console.error(err);
      } finally {
        setLoading(false);
      }
    };

    fetchProduct();
  }, [id]);

  const handleAddToCart = async () => {
    if (product.stock < 1) {
      alert('Немає в наявності');
      return;
    }
    try {
      await addToCart(id, quantity);
    }
  };
};
```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		77

```

    alert('Товар додано до кошика!');
  } catch (err) {
    console.error('Failed to add to cart:', err);
    alert('Не вдалося додати товар до кошика');
  }
};

const incrementQuantity = () => {
  if (quantity >= product.stock) {
    alert('Більше немає на складі');
    return;
  }
  setQuantity(quantity + 1);
};

const decrementQuantity = () => {
  if (quantity > 1) {
    setQuantity(quantity - 1);
  }
};

if (loading) {
  return (
    <div className="container mx-auto px-4 py-8">
      <div className="text-center py-8">Завантаження деталей товару...</div>
    </div>
  );
}

if (error) {
  return (
    <div className="container mx-auto px-4 py-8">
      <div className="text-center py-8 text-red-500">{error}</div>
    </div>
  );
}

if (!product) {
  return (
    <div className="container mx-auto px-4 py-8">
      <div className="text-center py-8">Товар не знайдено</div>
    </div>
  );
}

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		78

```

return (
  <div className="container mx-auto px-4 py-8">
    <div className="bg-white rounded-lg shadow-md overflow-hidden">
      <div className="md:flex">
        <div className="md:w-1/2">
          <OptimizedImage
            product={product}
            variant="full"
            alt={getProductName(product)}
            sizes="(max-width: 768px) 100vw, 50vw"
            fetchPriority="high"
            className="w-full h-96 md:h-full object-cover"
          />
        </div>
        <div className="md:w-1/2 p-6">
          <div className="flex justify-between items-start mb-4">
            <h1 className="text-2xl font-bold text-
primary">{getProductName(product)}</h1>
            {product.ecoFriendly && (
              <div className="eco-badge flex items-center space-x-1">
                <FaLeaf />
                <span>Еко-товари</span>
              </div>
            )}
          </div>

          <div className="flex items-center mb-4">
            <div className="flex text-yellow-400 mr-2">
              <FaStar />
              <FaStar />
              <FaStar />
              <FaStar />
              <FaStarHalfAlt />
            </div>
            <span className="text-gray-600">4.7 (128 відгуків)</span>
          </div>

          <p className="text-3xl font-bold text-primary mb-6">
            {product.price.toFixed(2)} ₪
          </p>

          <p className="text-gray-700 mb-6">{getProductDescription(product)}</p>

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

```

<div className="mb-6">
  <p className="text-gray-600 mb-2">
    <span className="font-semibold">Категорія:</span>
    {getProductLabel(product.category)}
  </p>
  <p className="text-gray-600">
    <span className="font-semibold">Наявність:</span> {product.stock}
  </p>
</div>

```

шт.

```

<div className="flex items-center mb-6">
  <div className="flex items-center border border-gray-300 rounded">
    <button
      onClick={decrementQuantity}
      className="px-3 py-2 hover:bg-gray-100"
    >
      -
    </button>
    <span className="px-4 py-2 border-l border-r border-gray-300">
      {quantity}
    </span>
    <button
      onClick={incrementQuantity}
      disabled={quantity >= product.stock}
      className="px-3 py-2 hover:bg-gray-100 disabled:cursor-not-allowed
disabled:opacity-40"
    >
      +
    </button>
  </div>
  <button
    onClick={handleAddToCart}
    disabled={product.stock < 1}
    className="btn-primary ml-4 flex items-center space-x-2 disabled:cursor-
not-allowed disabled:bg-slate-300"
  >
    <FaCartPlus />
    <span>{product.stock < 1 ? 'Немає в наявності' : 'До кошика'}</span>
  </button>
</div>

<div className="border-t pt-4">
  <Link to="/cart" className="btn-outline">

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		80

```
        Перейти до кошика
    </Link>
  </div>
</div>
</div>
</div>
```

```
<div className="mt-8 bg-white rounded-lg shadow-md p-6">
  <h3 className="text-xl font-bold mb-4">Деталі товари</h3>
  <p className="text-gray-700">
    {getProductDescription(product)} Цей екологічний продукт виготовлено зі
    сталих матеріалів
    і призначений для безпеки ваших вихованців та навколишнього
    середовища.
  </p>
</div>
</div>
);
};
```

```
export default ProductDetailsPage;
```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

Додаток Е. CartPage.jsx

```
import { useState, useEffect } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import api from '../utils/api';
import { useCart } from '../context/CartContext';
import { useAuth } from '../context/AuthContext';
import {
  FaTrash,
  FaMinus,
  FaPlus,
  FaLeaf,
  FaShoppingBag,
  FaArrowRight,
  FaTruck,
  FaLock,
  FaCheckCircle,
} from 'react-icons/fa';
import { SiVisa, SiMastercard, SiPaypal } from 'react-icons/si';
import { motion } from 'framer-motion';
import Modal from '../components/Modal';
import { useToast, ToastContainer } from '../components/Toast';
import { getProductNames } from '../config/productTranslations';
import OptimizedImage from '../components/OptimizedImage';
import { getProductImageUrl } from '../utils/unsplashImages';

const CartPage = () => {
  const { cart, loading, error, updateCartItem, removeFromCart, clearCart, getCartTotal } = useCart();
  const { user } = useAuth();
  const navigate = useNavigate();
  const { toasts, showToast, removeToast } = useToast();

  const [shippingInfo, setShippingInfo] = useState({
    address: "",
    city: "",
    postalCode: "",
    country: "",
    phone: "",
  });
  const [paymentMethod, setPaymentMethod] = useState('credit_card');
```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82

```

const [modal, setModal] = useState({ isOpen: false, type: 'info', title: "", message: "",
buttons: [] });

useEffect(() => {
  if (user?.address) {
    setShippingInfo({
      address: user.address.street || "",
      city: user.address.city || "",
      postalCode: user.address.postalCode || "",
      country: user.address.country || "",
      phone: user.phone || "",
    });
  }
}, [user]);

const handleQuantityChange = async (item, newQuantity) => {
  if (newQuantity < 1) return;
  if (newQuantity > item.product.stock) {
    showToast('Більше немає на складі', 'warning');
    return;
  }
  try {
    await updateCartItem(item._id, newQuantity);
  } catch (err) {
    showToast(err.message || 'Не вдалося оновити кількість', 'error');
  }
};

const handleRemoveItem = (itemId, productName) => {
  setModal({
    isOpen: true,
    type: 'warning',
    title: 'Видалити товар?',
    message: `Ви впевнені, що хочете видалити "${productName}" з кошика?`,
    buttons: [
      {
        label: 'Скасувати',
        variant: 'secondary',
        onClick: () => {},
      },
      {
        label: 'Видалити',
        variant: 'danger',
        onClick: async () => {

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

```

    try {
      await removeFromCart(itemId);
      showToast('Товар видалено з кошика', 'success');
    } catch (err) {
      showToast('Не вдалося видалити товар', 'error');
    }
  },
},
],
});
};

```

```

const handleClearCart = () => {
  setModal({
    isOpen: true,
    type: 'warning',
    title: 'Очистити кошик?',
    message: 'Це видалить усі товари з вашого кошика. Ця дія не можна скасувати.',
    buttons: [
      {
        label: 'Скасувати',
        variant: 'secondary',
        onClick: () => {},
      },
      {
        label: 'Очистити',
        variant: 'danger',
        onClick: async () => {
          try {
            await clearCart();
            showToast('Кошик очищено', 'success');
          } catch (err) {
            showToast('Не вдалося очистити кошик', 'error');
          }
        },
      },
    ],
  });
};

```

```

const validateCheckout = () => {
  if (!user) {
    setModal({

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

```

isOpen: true,
type: 'warning',
title: 'Увійти в систему',
message: 'Будь ласка, увійдіть в систему для оформлення замовлення.',
buttons: [
  { label: 'Скасувати', variant: 'secondary', onClick: () => {} },
  {
    label: 'Перейти до входу',
    onClick: () => navigate('/login'),
    closeOnClick: false,
  },
],
});
return false;
}

```

```

if (!cart?.items?.length) {
  setModal({
    isOpen: true,
    type: 'warning',
    title: 'Кошик порожній',
    message: 'Додайте товари до кошика для оформлення замовлення.',
    buttons: [
      {
        label: 'Перейти до каталогу',
        onClick: () => navigate('/catalog'),
        closeOnClick: false,
      },
    ],
  });
  return false;
}

```

```

if (!shippingInfo.address || !shippingInfo.city || !shippingInfo.postalCode ||
!shippingInfo.country) {
  showToast('Будь ласка, заповніть всю інформацію про доставку', 'error');
  return false;
}

```

```

const unavailableItem = cart.items.find((item) => item.quantity >
item.product.stock);
if (unavailableItem) {
  showToast(

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

```

    `На складі доступно лише ${unavailableItem.product.stock} шт. товару
    ${getProductName(unavailableItem.product)}`,
    'error'
  );
  return false;
}

return true;
};

const handleCheckout = async () => {
  if (!validateCheckout()) return;

  try {
    // Prepare order data
    const orderData = {
      shippingAddress: shippingInfo,
      paymentMethod,
      items: cart.items.map((item) => ({
        product: item.product._id,
        name: getProductName(item.product),
        image: getProductImageUrl(item.product, 'full'),
        quantity: item.quantity,
        price: item.price,
      })),
      total: getCartTotal(),
    };

    // Create order via API
    const { data } = await api.post('/api/v1/orders', orderData);

    if (data.success) {
      // Show success modal
      setModal({
        isOpen: true,
        type: 'success',
        title: '✔ Заовлення оформлено!',
        message: `Спасибо за покупку! Номер заовлення: ${data.data._id.slice(-
        6).toUpperCase()}. \nВи отримаєте конфірмацію на пошту.` ,
        buttons: [
          {
            label: 'Перейти на головну',
            onClick: async () => {
              await clearCart();
            }
          }
        ]
      });
    }
  }
};

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		86

```

        navigate('/');
    },
    closeOnClick: false,
  },
],
});

showToast('Замовлення успішно оформлено!', 'success');
}
} catch (err) {
  console.error('Checkout error:', err);
  const errorMessage = err.response?.data?.error || err.response?.data?.message || 'Не
вдалося оформити замовлення!';
  showToast(errorMessage, 'error');
  setModal({
    isOpen: true,
    type: 'error',
    title: '✘ Помилка',
    message: errorMessage,
    buttons: [
      { label: 'ОК', onClick: () => {} },
    ],
  });
}
};

const containerVariants = {
  hidden: { opacity: 0 },
  visible: {
    opacity: 1,
    transition: { staggerChildren: 0.1, delayChildren: 0.2 },
  },
};

const itemVariants = {
  hidden: { opacity: 0, y: 20 },
  visible: { opacity: 1, y: 0, transition: { duration: 0.4 } },
};

if (loading) {
  return (
    <motion.div
      initial={{ opacity: 0 }}

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		87

```

    animate={{ opacity: 1 }}
    className="min-h-screen bg-gradient-to-br from-gray-50 to-gray-100
dark:from-gray-900 dark:to-gray-800 pt-32 pb-12"
  >
    <div className="container mx-auto px-6 text-center">
      <motion.div
        animate={{ scale: [1, 1.1, 1] }}
        transition={{ duration: 2, repeat: Infinity }}
        className="text-6xl mb-4"
      >
        🛒
      </motion.div>
      <p className="text-lg text-gray-600 dark:text-gray-300">Завантаження
кошика...</p>
    </div>
  </motion.div>
);
}

if (error) {
  return (
    <motion.div
      initial={{ opacity: 0 }}
      animate={{ opacity: 1 }}
      className="min-h-screen bg-gradient-to-br from-gray-50 to-gray-100
dark:from-gray-900 dark:to-gray-800 pt-32 pb-12"
    >
      <div className="container mx-auto px-6 text-center">
        <p className="text-lg text-red-500 font-semibold">{error}</p>
        <Link to="/catalog" className="inline-block mt-4 btn-primary">
          Повернутися до каталогу
        </Link>
      </div>
    </motion.div>
  );
}

const isEmpty = !cart?.items || cart.items.length === 0;

return (
  <motion.div
    initial={{ opacity: 0 }}
    animate={{ opacity: 1 }}

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		88

```

className="min-h-screen bg-gradient-to-br from-gray-50 to-gray-100 dark:from-
gray-900 dark:to-gray-800 pt-32 pb-12"
>
<div className="container mx-auto px-6">
  { /* Header */ }
  <motion.div
    variants={containerVariants}
    initial="hidden"
    animate="visible"
    className="mb-12"
  >
    <motion.div variants={itemVariants}>
      <h1 className="text-4xl md:text-5xl font-extrabold text-gray-900 dark:text-
white mb-2">
        Ваш кошик
      </h1>
      <p className="text-gray-600 dark:text-gray-400">
        {isEmpty ? 'Кошик порожній' : `${cart.items.length} товарів`}
      </p>
    </motion.div>
  </motion.div>

  { /* Empty State */ }
  {isEmpty ? (
    <motion.div
      variants={containerVariants}
      initial="hidden"
      animate="visible"
      className="max-w-md mx-auto text-center"
    >
      <motion.div variants={itemVariants} className="mb-6">
        <motion.div
          animate={{ y: [0, -20, 0] }}
          transition={{ duration: 2, repeat: Infinity }}
          className="text-9xl mb-6"
        >
          <img alt="Shopping cart icon" data-bbox="238 738 262 754" />
        </motion.div>
        <h2 className="text-3xl font-bold text-gray-900 dark:text-white mb-3">
          Ваш кошик порожній
        </h2>
        <p className="text-gray-600 dark:text-gray-400 mb-8 text-lg">
          Додайте улюблені товари до кошика та оформіть замовлення!
        </p>
      </motion.div>
    </motion.div>
  )}

```

```

    <motion.div whileHover={{ scale: 1.05 }} whileTap={{ scale: 0.95 }}>
      <Link
        to="/catalog"
        className="inline-flex items-center gap-2 bg-gradient-to-r from-primary
to-accent text-white px-8 py-4 rounded-2xl font-bold text-lg shadow-lg hover:shadow-
xl transition-all"
      >
        <FaShoppingBag /> Переглянути товари
      </Link>
    </motion.div>
  </motion.div>
</motion.div>
):(
<motion.div
  variants={containerVariants}
  initial="hidden"
  animate="visible"
  className="grid grid-cols-1 lg:grid-cols-3 gap-8"
>
  { /* Cart Items */ }
  <motion.div variants={itemVariants} className="lg:col-span-2 space-y-4">
    { cart.items.map((item, index) => (
      <motion.div
        key={item._id}
        initial={{ opacity: 0, x: -20 }}
        animate={{ opacity: 1, x: 0 }}
        transition={{ delay: index * 0.1 }}
        whileHover={{ scale: 1.01 }}
        className="bg-white dark:bg-gray-800 rounded-2xl p-6 shadow-sm
hover:shadow-md transition-all border border-gray-200/50 dark:border-gray-700/50"
      >
        <div className="flex gap-6">
          { /* Image */ }
          <motion.div
            whileHover={{ scale: 1.05 }}
            className="flex-shrink-0"
          >
            <OptimizedImage
              product={item.product}
              alt={getProductName(item.product)}
              variant="preview"
              className="w-24 h-24 object-cover rounded-xl"
            />
          </motion.div>
        </div>
      </motion.div>
    ) ) }
  </motion.div>

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		90

```

    {/* Product Info */}
    <div className="flex-1">
      <div className="flex justify-between items-start mb-2">
        <div>
          <h3 className="text-lg font-bold text-gray-900 dark:text-white">
            {getProductName(item.product)}
          </h3>
          {item.product.ecoFriendly && (
            <motion.div
              initial={{ opacity: 0 }}
              animate={{ opacity: 1 }}
              className="inline-flex items-center gap-1 mt-2 bg-green-100
dark:bg-green-900/30 text-green-700 dark:text-green-300 px-3 py-1 rounded-full text-xs
font-semibold"
            >
              <FaLeaf /> Еко
            </motion.div>
          )}
        </div>
        <motion.button
          whileHover={{ scale: 1.1, rotate: 20 }}
          whileTap={{ scale: 0.9 }}
          onClick={() => handleRemoveItem(item._id,
getProductName(item.product))}
          className="p-2 hover:bg-red-50 dark:hover:bg-red-900/20
rounded-lg transition-colors group"
        >
          <FaTrash className="text-red-500 group-hover:text-red-600 text-
lg" />
        </motion.button>
      </div>

    {/* Price & Quantity */}
    <div className="flex items-center justify-between">
      <div>
        <p className="text-2xl font-bold text-primary">
          {(item.price * item.quantity).toFixed(2)} ₴
        </p>
        <p className="text-sm text-gray-600 dark:text-gray-400">
          {item.price.toFixed(2)} ₴ за одиницю
        </p>
      </div>

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		91

```

        {/* Quantity Controls */}
        <div className="flex items-center gap-3 bg-gray-100 dark:bg-gray-
700 p-1 rounded-full">
            <motion.button
                whileHover={{ scale: 1.1 }}
                whileTap={{ scale: 0.9 }}
                onClick={() => handleQuantityChange(item, item.quantity - 1)}
                disabled={item.quantity <= 1}
                className="w-8 h-8 rounded-full hover:bg-gray-200
dark: hover:bg-gray-600 flex items-center justify-center disabled:opacity-50
disabled:cursor-not-allowed transition-colors"
            >
                <FaMinus className="text-sm" />
            </motion.button>
            <span className="w-8 text-center font-bold text-gray-900
dark:text-white">
                {item.quantity}
            </span>
            <motion.button
                whileHover={{ scale: 1.1 }}
                whileTap={{ scale: 0.9 }}
                onClick={() => handleQuantityChange(item, item.quantity + 1)}
                aria-disabled={item.quantity >= item.product.stock}
                className="w-8 h-8 rounded-full hover:bg-gray-200
dark: hover:bg-gray-600 flex items-center justify-center transition-colors"
            >
                <FaPlus className="text-sm" />
            </motion.button>
        </div>
    </div>
    <p className="mt-3 text-xs font-semibold text-gray-500 dark:text-
gray-400">
        Доступно на складі: {item.product.stock} шт.
    </p>
</div>
</div>
</motion.div>
)}}

{/* Cart Actions */}
<motion.div
    variants={itemVariants}
    className="flex gap-3 mt-8"
    >

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		92

```

<motion.div whileHover={{ scale: 1.05 }} whileTap={{ scale: 0.95 }}>
  <Link
    to="/catalog"
    className="inline-flex items-center gap-2 px-6 py-3 rounded-xl
border-2 border-gray-300 dark:border-gray-600 text-gray-900 dark:text-white font-bold
hover:bg-gray-100 dark: hover:bg-gray-700 transition-colors"
  >
    Продовжити покупки
  </Link>
</motion.div>
<motion.button
  whileHover={{ scale: 1.05 }}
  whileTap={{ scale: 0.95 }}
  onClick={handleClearCart}
  className="inline-flex items-center gap-2 px-6 py-3 rounded-xl text-
red-600 dark:text-red-400 font-bold hover:bg-red-50 dark: hover:bg-red-900/20
transition-colors"
  >
  <FaTrash /> ОЧИСТИТИ КОШИК
</motion.button>
</motion.div>
</motion.div>

{/* Sidebar - Checkout */}
<motion.div variants={itemVariants} className="lg:col-span-1">
  <motion.div
    className="sticky top-32 space-y-6"
  >
    {/* Order Summary */}
    <div className="bg-white dark:bg-gray-800 rounded-2xl p-6 shadow-sm
border border-gray-200/50 dark: border-gray-700/50 backdrop-blur-lg">
      <h2 className="text-xl font-bold text-gray-900 dark:text-white mb-6">
        Підсумок замовлення
      </h2>

      <div className="space-y-4 mb-6 pb-6 border-b border-gray-200
dark: border-gray-700">
        <div className="flex justify-between text-gray-600 dark:text-gray-
400">
          <span>Проміжний підсумок</span>
          <span className="font-semibold">{getCartTotal().toFixed(2)}
€</span>
        </div>
    </div>
  </div>

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		93

```

400">
    <div className="flex justify-between text-gray-600 dark:text-gray-
    <span className="flex items-center gap-2">
      <FaTruck className="text-primary" />
      Доставка
    </span>
    <span className="font-semibold text-green-600 dark:text-green-
400">
      Безкоштовно
    </span>
  </div>
</div>

<div className="flex justify-between mb-6">
  <span className="text-xl font-bold text-gray-900 dark:text-
white">Всього</span>
  <span className="text-2xl font-extrabold bg-gradient-to-r from-
primary to-accent bg-clip-text text-transparent">
    {getCartTotal().toFixed(2)} ₪
  </span>
</div>

{/* Trust Badges */}
<div className="space-y-3 pt-4 border-t border-gray-200 dark:border-
gray-700">
  <div className="flex items-center gap-3 text-sm text-gray-700
dark:text-gray-300">
    <FaLock className="text-green-600 dark:text-green-400 text-lg" />
    Безпечна оплата
  </div>
  <div className="flex items-center gap-3 text-sm text-gray-700
dark:text-gray-300">
    <FaTruck className="text-primary text-lg" />
    Швидка доставка
  </div>
  <div className="flex items-center gap-3 text-sm text-gray-700
dark:text-gray-300">
    <FaCheckCircle className="text-primary text-lg" />
    Гарантія якості
  </div>
</div>
</div>

{/* Shipping Info */}

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		94

```

<div className="bg-white dark:bg-gray-800 rounded-2xl p-6 shadow-sm
border border-gray-200/50 dark:border-gray-700/50">
  <h3 className="font-bold text-lg text-gray-900 dark:text-white mb-4">
    Адреса доставки
  </h3>
  <div className="space-y-3">
    <input
      type="text"
      placeholder="Вулиця, будинок, квартира"
      value={shippingInfo.address}
      onChange={(e) => setShippingInfo({ ...shippingInfo, address:
e.target.value })}
      className="w-full px-4 py-3 rounded-xl border border-gray-300
dark:border-gray-600 bg-white dark:bg-gray-700 text-gray-900 dark:text-white
placeholder-gray-500 dark:placeholder-gray-400 focus:outline-none focus:ring-2
focus:ring-primary/50 transition-all"
    />
    <div className="grid grid-cols-2 gap-3">
      <input
        type="text"
        placeholder="Місто"
        value={shippingInfo.city}
        onChange={(e) => setShippingInfo({ ...shippingInfo, city:
e.target.value })}
        className="px-4 py-3 rounded-xl border border-gray-300
dark:border-gray-600 bg-white dark:bg-gray-700 text-gray-900 dark:text-white
placeholder-gray-500 dark:placeholder-gray-400 focus:outline-none focus:ring-2
focus:ring-primary/50 transition-all"
      />
      <input
        type="text"
        placeholder="Поштовий індекс"
        value={shippingInfo.postalCode}
        onChange={(e) => setShippingInfo({ ...shippingInfo, postalCode:
e.target.value })}
        className="px-4 py-3 rounded-xl border border-gray-300
dark:border-gray-600 bg-white dark:bg-gray-700 text-gray-900 dark:text-white
placeholder-gray-500 dark:placeholder-gray-400 focus:outline-none focus:ring-2
focus:ring-primary/50 transition-all"
      />
    </div>
    <input
      type="text"
      placeholder="Країна"

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		95

```

        value={shippingInfo.country}
        onChange={(e) => setShippingInfo({ ...shippingInfo, country:
e.target.value })}
        className="w-full px-4 py-3 rounded-xl border border-gray-300
dark:border-gray-600 bg-white dark:bg-gray-700 text-gray-900 dark:text-white
placeholder-gray-500 dark:placeholder-gray-400 focus:outline-none focus:ring-2
focus:ring-primary/50 transition-all"
      />
      <input
        type="tel"
        placeholder="Телефон (необов'язково)"
        value={shippingInfo.phone}
        onChange={(e) => setShippingInfo({ ...shippingInfo, phone:
e.target.value })}
        className="w-full px-4 py-3 rounded-xl border border-gray-300
dark:border-gray-600 bg-white dark:bg-gray-700 text-gray-900 dark:text-white
placeholder-gray-500 dark:placeholder-gray-400 focus:outline-none focus:ring-2
focus:ring-primary/50 transition-all"
      />
    </div>
  </div>

  {/* Payment Method */}
  <div className="bg-white dark:bg-gray-800 rounded-2xl p-6 shadow-sm
border border-gray-200/50 dark:border-gray-700/50">
    <h3 className="font-bold text-lg text-gray-900 dark:text-white mb-4">
      Спосіб оплати
    </h3>
    <div className="space-y-3">
      {[
        { value: 'credit_card', label: 'КРЕДИТНА КАРТКА', icon: <SiVisa
className="text-xl" /> },
        { value: 'mastercard', label: 'Mastercard', icon: <SiMastercard
className="text-xl" /> },
        { value: 'paypal', label: 'PayPal', icon: <SiPaypal className="text-xl"
/> },
        { value: 'cash_on_delivery', label: 'Оплата при отриманні', icon: '📦'
},
      ].map((method) => (
        <motion.label
          key={method.value}
          whileHover={{ backgroundColor: 'rgba(0,0,0,0.02)' }}

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		96

```

        className="flex items-center gap-3 p-3 rounded-lg border-2
border-gray-200 dark:border-gray-700 cursor-pointer hover:border-primary/50
dark:hover:border-primary/50 transition-all"
      >
        <input
          type="radio"
          value={method.value}
          checked={paymentMethod === method.value}
          onChange={(e) => setPaymentMethod(e.target.value)}
          className="w-5 h-5 accent-primary"
        />
        <span className="text-xl">{method.icon}</span>
        <span className="flex-1 font-medium text-gray-900 dark:text-
white">
          {method.label}
        </span>
        {paymentMethod === method.value && (
          <motion.div
            initial={{ scale: 0 }}
            animate={{ scale: 1 }}
            className="w-5 h-5 rounded-full bg-gradient-to-r from-primary
to-accent"
          />
        )}
      </motion.label>
    )}
  </div>
</div>

{/* Checkout Button */}
<motion.button
  whileHover={{ scale: 1.05, boxShadow: '0 20px 25px -5px rgba(45, 125,
91, 0.3)' }}
  whileTap={{ scale: 0.95 }}
  onClick={handleCheckout}
  className="w-full bg-gradient-to-r from-primary to-accent text-white
py-4 px-6 rounded-2xl font-bold text-lg shadow-lg hover:shadow-xl transition-all flex
items-center justify-center gap-2"
  >
  <FaArrowRight /> Оформити замовлення
</motion.button>
</motion.div>
</motion.div>
</motion.div>

```

					2026.КВР.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		97

```

    })
  </div>

  { /* Modal */ }
  <Modal
    isOpen={modal.isOpen}
    title={modal.title}
    message={modal.message}
    type={modal.type}
    buttons={modal.buttons}
    onClose={() => setModal({ ...modal, isOpen: false })}
  />

  { /* Toast Notifications */ }
  <ToastContainer toasts={toasts} onRemove={removeToast} />
</motion.div>
);
};

export default CartPage;

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		98

Додаток К. server.js

```
'use strict';

Object.defineProperty(exports, '__esModule', { value: true });

var React = require('react');
var router = require('@remix-run/router');
var reactRouter = require('react-router');
var reactRouterDom = require('react-router-dom');

function _interopNamespace(e) {
  if (e && e.__esModule) return e;
  var n = Object.create(null);
  if (e) {
    Object.keys(e).forEach(function (k) {
      if (k !== 'default') {
        var d = Object.getOwnPropertyDescriptor(e, k);
        Object.defineProperty(n, k, d.get ? d : {
          enumerable: true,
          get: function () { return e[k]; }
        });
      }
    });
  }
  n["default"] = e;
  return Object.freeze(n);
}

var React__namespace = /*#__PURE__*/_interopNamespace(React);

/**
 * A `` that may not navigate to any other location. This is useful
 * on the server where there is no stateful UI.
 */
function StaticRouter({
  basename,
  children,
  location: locationProp = "/",
  future
}) {
  if (typeof locationProp === "string") {
    locationProp = reactRouterDom.parsePath(locationProp);
  }
}
```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		99

```

}
let action = router.Action.Pop;
let location = {
  pathname: locationProp.pathname || "/",
  search: locationProp.search || "",
  hash: locationProp.hash || "",
  state: locationProp.state != null ? locationProp.state : null,
  key: locationProp.key || "default"
};
let staticNavigator = getStatelessNavigator();
return /*#__PURE__*/React__namespace.createElement(ReactDOM.Router, {
  basename: basename,
  children: children,
  location: location,
  navigationType: action,
  navigator: staticNavigator,
  future: future,
  static: true
});
}
/**
 * A Data Router that may not navigate to any other location. This is useful
 * on the server where there is no stateful UI.
 */
function StaticRouterProvider({
  context,
  router: router$1,
  hydrate = true,
  nonce
}) {
  !(router$1 && context) ? process.env.NODE_ENV !== "production" ?
  router.UNSAFE_invariant(false, "You must provide `router` and `context` to
  <StaticRouterProvider>") : router.UNSAFE_invariant(false) : void 0;
  let dataRouterContext = {
    router: router$1,
    navigator: getStatelessNavigator(),
    static: true,
    staticContext: context,
    basename: context.basename || "/"
  };
  let fetchersContext = new Map();
  let hydrateScript = "";
  if (hydrate !== false) {
    let data = {

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		100

```

loaderData: context.loaderData,
actionData: context.actionData,
errors: serializeErrors(context.errors)
};
// Use JSON.parse here instead of embedding a raw JS object here to speed
// up parsing on the client. Dual-stringify is needed to ensure all quotes
// are properly escaped in the resulting string. See:
// https://v8.dev/blog/cost-of-javascript-2019#json
let json = htmlEscape(JSON.stringify(JSON.stringify(data)));
hydrateScript = `window.__staticRouterHydrationData = JSON.parse(${json});`;
}
let {
state
} = dataRouterContext.router;
return
/*#__PURE__*/React__namespace.createElement(React__namespace.Fragment, null,
/*#__PURE__*/React__namespace.createElement(reactRouterDom.UNSAFE_DataRouterContext.Provider, {
value: dataRouterContext
}),
/*#__PURE__*/React__namespace.createElement(reactRouterDom.UNSAFE_DataRouterStateContext.Provider, {
value: state
}),
/*#__PURE__*/React__namespace.createElement(reactRouterDom.UNSAFE_FetchersContext.Provider, {
value: fetchersContext
}),
/*#__PURE__*/React__namespace.createElement(reactRouterDom.UNSAFE_ViewTransitionContext.Provider, {
value: {
isTransitioning: false
}
}),
/*#__PURE__*/React__namespace.createElement(reactRouterDom.Router, {
basename: dataRouterContext.basename,
location: state.location,
navigationType: state.historyAction,
navigator: dataRouterContext.navigator,
static: dataRouterContext.static,
future: {
v7_relativeSplatPath: router$1.future.v7_relativeSplatPath
}
}),
/*#__PURE__*/React__namespace.createElement(DataRoutes, {
routes: router$1.routes,

```

						2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			101

```

    future: router$1.future,
    state: state
  ))))))) , hydrateScript ? (/*#__PURE__*/React__namespace.createElement("script", {
    suppressHydrationWarning: true,
    nonce: nonce,
    dangerouslySetInnerHTML: {
      __html: hydrateScript
    }
  ))) : null);
}
function DataRoutes({
  routes,
  future,
  state
}) {
  return reactRouter.UNSAFE_useRoutesImpl(routes, undefined, state, future);
}
function serializeErrors(errors) {
  if (!errors) return null;
  let entries = Object.entries(errors);
  let serialized = {};
  for (let [key, val] of entries) {
    // Hey you! If you change this, please change the corresponding logic in
    // deserializeErrors in react-router-dom/index.tsx :)
    if (router.isRouteErrorResponse(val)) {
      serialized[key] = {
        ...val,
        __type: "RouteErrorResponse"
      };
    } else if (val instanceof Error) {
      // Do not serialize stack traces from SSR for security reasons
      serialized[key] = {
        message: val.message,
        __type: "Error",
        // If this is a subclass (i.e., ReferenceError), send up the type so we
        // can re-create the same type during hydration.
        ...(val.name !== "Error" ? {
          __subType: val.name
        } : {})
      };
    } else {
      serialized[key] = val;
    }
  }
}

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		102

```

return serialized;
}
function getStatelessNavigator() {
return {
  createHref,
  encodeLocation,
  push(to) {
    throw new Error(`You cannot use navigator.push() on the server because it is a
stateless ` + `environment. This error was probably triggered when you did a ` +
`navigate(${JSON.stringify(to)})` somewhere in your app.`);
  },
  replace(to) {
    throw new Error(`You cannot use navigator.replace() on the server because it is a
stateless ` + `environment. This error was probably triggered when you did a ` +
`navigate(${JSON.stringify(to)}, { replace: true })` somewhere ` + `in your app.`);
  },
  go(delta) {
    throw new Error(`You cannot use navigator.go() on the server because it is a
stateless ` + `environment. This error was probably triggered when you did a ` +
`navigate(${delta})` somewhere in your app.`);
  },
  back() {
    throw new Error(`You cannot use navigator.back() on the server because it is a
stateless ` + `environment.`);
  },
  forward() {
    throw new Error(`You cannot use navigator.forward() on the server because it is a
stateless ` + `environment.`);
  }
};
}
function createStaticHandler(routes, opts) {
return router.createStaticHandler(routes, {
  ...opts,
  mapRouteProperties: reactRouter.UNSAFE_mapRouteProperties
});
}
function createStaticRouter(routes, context, opts = {}) {
let manifest = {};
let dataRoutes = router.UNSAFE_convertRoutesToDataRoutes(routes,
reactRouter.UNSAFE_mapRouteProperties, undefined, manifest);
// Because our context matches may be from a framework-agnostic set of
// routes passed to createStaticHandler(), we update them here with our
// newly created/enhanced data routes

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		103

```

let matches = context.matches.map(match => {
  let route = manifest[match.route.id] || match.route;
  return {
    ...match,
    route
  };
});
let msg = method => `You cannot use router.${method}() on the server because it is a
stateless environment`;
return {
  get basename() {
    return context.basename;
  },
  get future() {
    return {
      v7_fetcherPersist: false,
      v7_normalizeFormMethod: false,
      v7_partialHydration: opts.future?.v7_partialHydration === true,
      v7_prependBasename: false,
      v7_relativeSplatPath: opts.future?.v7_relativeSplatPath === true,
      v7_skipActionErrorRevalidation: false
    };
  },
  get state() {
    return {
      historyAction: router.Action.Pop,
      location: context.location,
      matches,
      loaderData: context.loaderData,
      actionData: context.actionData,
      errors: context.errors,
      initialized: true,
      navigation: router.IDLE_NAVIGATION,
      restoreScrollPosition: null,
      preventScrollReset: false,
      revalidation: "idle",
      fetchers: new Map(),
      blockers: new Map()
    };
  },
  get routes() {
    return dataRoutes;
  },
  get window() {

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		104

```

    return undefined;
  },
  initialize() {
    throw msg("initialize");
  },
  subscribe() {
    throw msg("subscribe");
  },
  enableScrollRestoration() {
    throw msg("enableScrollRestoration");
  },
  navigate() {
    throw msg("navigate");
  },
  fetch() {
    throw msg("fetch");
  },
  revalidate() {
    throw msg("revalidate");
  },
  createHref,
  encodeLocation,
  getFetcher() {
    return router.IDLE_FETCHER;
  },
  deleteFetcher() {
    throw msg("deleteFetcher");
  },
  dispose() {
    throw msg("dispose");
  },
  getBlocker() {
    return router.IDLE_BLOCKER;
  },
  deleteBlocker() {
    throw msg("deleteBlocker");
  },
  patchRoutes() {
    throw msg("patchRoutes");
  },
  _internalFetchControllers: new Map(),
  _internalActiveDeferreds: new Map(),
  _internalSetRoutes() {
    throw msg("_internalSetRoutes");
  }

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		105

```

    }
  };
}
function createHref(to) {
  return typeof to === "string" ? to : reactRouterDom.createPath(to);
}
function encodeLocation(to) {
  let href = typeof to === "string" ? to : reactRouterDom.createPath(to);
  // Treating this as a full URL will strip any trailing spaces so we need to
  // pre-encode them since they might be part of a matching splat param from
  // an ancestor route
  href = href.replace(/ $/, "%20");
  let encoded = ABSOLUTE_URL_REGEX.test(href) ? new URL(href) : new
URL(href, "http://localhost");
  return {
    pathname: encoded.pathname,
    search: encoded.search,
    hash: encoded.hash
  };
}
const ABSOLUTE_URL_REGEX = /^(?:[a-z][a-z0-9+.-]*:|\\)/i;
// This utility is based on https://github.com/zertosh/htmlescape
// License:
https://github.com/zertosh/htmlescape/blob/0527ca7156a524d256101bb310a9f970f630
78ad/LICENSE
const ESCAPE_LOOKUP = {
  "&": "\\u0026",
  ">": "\\u003e",
  "<": "\\u003c",
  "\u2028": "\\u2028",
  "\u2029": "\\u2029"
};
const ESCAPE_REGEX = /[&><\u2028\u2029]/g;
function htmlEscape(str) {
  return str.replace(ESCAPE_REGEX, match => ESCAPE_LOOKUP[match]);
}

exports.StaticRouter = StaticRouter;
exports.StaticRouterProvider = StaticRouterProvider;
exports.createStaticHandler = createStaticHandler;
exports.createStaticRouter = createStaticRouter;

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		106

Додаток II. admin.js

```
"use strict";
Object.defineProperty(exports, "__esModule", { value: true });
exports.Admin = void 0;

const add_user_1 = require("./operations/add_user");
const execute_operation_1 = require("./operations/execute_operation");
const list_databases_1 = require("./operations/list_databases");
const remove_user_1 = require("./operations/remove_user");
const run_command_1 = require("./operations/run_command");
const validate_collection_1 = require("./operations/validate_collection");

class Admin {
  constructor(db) {
    this.s = { db };
  }
  ...
  async command(command, options) {
    return (0, execute_operation_1.executeOperation)(
      this.s.db.client,
      new run_command_1.RunCommandOperation(
        this.s.db,
        command,
        { dbName: 'admin', ...options }
      )
    );
  }

  async buildInfo(options) {
    return this.command({ buildinfo: 1 }, options);
  }

  async serverInfo(options) {
    return this.command({ buildinfo: 1 }, options);
  }

  async serverStatus(options) {
    return this.command({ serverStatus: 1 }, options);
  }

  async ping(options) {
```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		107

```

    return this.command({ ping: 1 }, options);
}

async addUser(username, passwordOrOptions, options) {
    options =
        options != null && typeof options === 'object'
            ? options
            : passwordOrOptions != null && typeof passwordOrOptions === 'object'
                ? passwordOrOptions
                : undefined;

    const password =
        typeof passwordOrOptions === 'string'
            ? passwordOrOptions
            : undefined;

    return (0, execute_operation_1.executeOperation)(
        this.s.db.client,
        new add_user_1.AddUserOperation(
            this.s.db,
            username,
            password,
            { dbName: 'admin', ...options }
        )
    );
}

async removeUser(username, options) {
    return (0, execute_operation_1.executeOperation)(
        this.s.db.client,
        new remove_user_1.RemoveUserOperation(
            this.s.db,
            username,
            { dbName: 'admin', ...options }
        )
    );
}

async validateCollection(collectionName, options = {}) {
    return (0, execute_operation_1.executeOperation)(
        this.s.db.client,
        new validate_collection_1.ValidateCollectionOperation(
            this,
            collectionName,

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		108

```

        options
    )
);
}

async listDatabases(options) {
    return (0, execute_operation_1.executeOperation)(
        this.s.db.client,
        new list_databases_1.ListDatabasesOperation(
            this.s.db,
            options
        )
    );
}

async replSetGetStatus(options) {
    return this.command({ replSetGetStatus: 1 }, options);
}
...

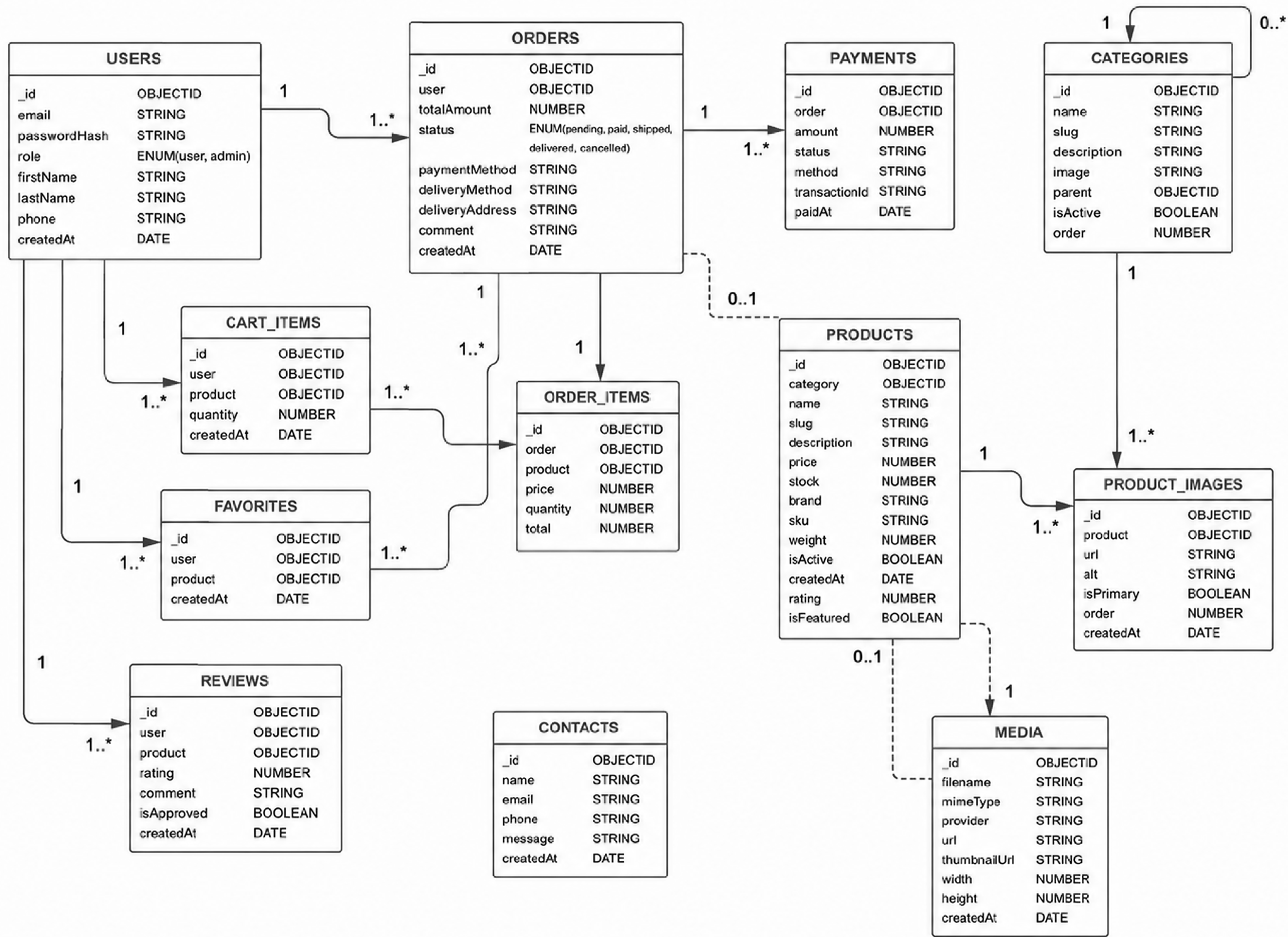
}

exports.Admin = Admin;

```

					2026.KBP.122.421.21.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		109

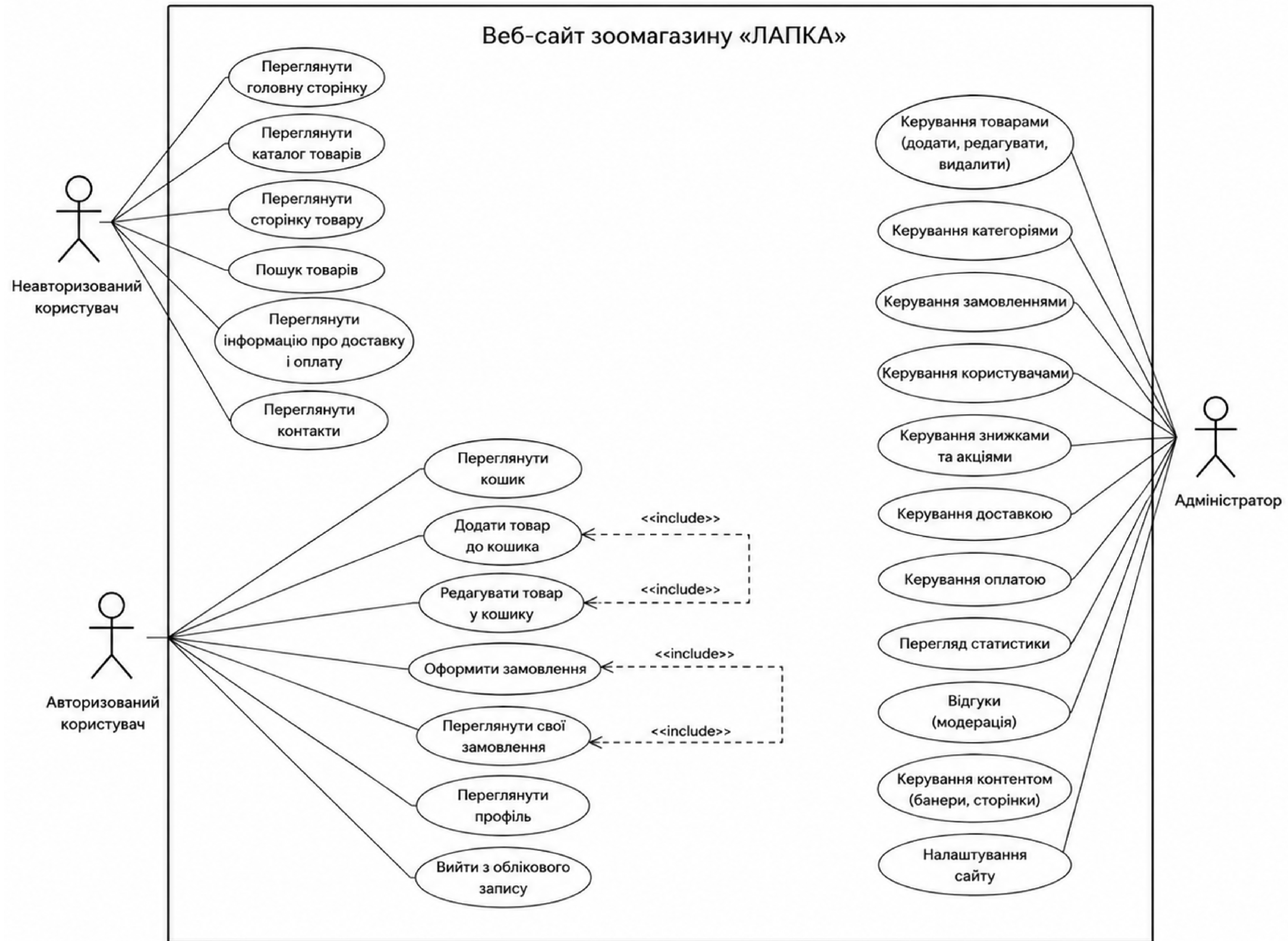
ER-діаграма бази даних вебсайту



				2026.KBP.122.4212100.00 БД				
Зм.	Арх.	№ докум.	Підп.	Дата	Розробка вебсайту замовнику "Лопка"	Лист	Маса	Масштаб
Розроб.	Арх.	Чокан В. В.			ER-діаграма бази даних вебсайту	Архів	Архів	1
Перев.	Арх.	Марціняк Г. Я.						
І.контр.								
Рецензент								
І.контр.		Приймак В. А.						
Затв.								
						ВСП ТФК ТНТУ КН-421		
						м. Тернопіль		
						Формат А1		

Лист № докум. / Архів / Маса / Масштаб / Підп. / Дата / Арх. / Архів / 1 / ВСП ТФК ТНТУ / КН-421 / м. Тернопіль / Формат А1

UML-діаграма варіантів використання програми



Перш. запис.
Стор. №
Підп. і дата
Зем. від. №
Міс. від. №
Лист. № архіву

				2026.KBP.122.421.2100.00 ДВ				
Зм.	Арх.	№ док.	Підп.	Дата	Розробка вебсайту зоомагазину "Лапка"	Лист	Маса	Масштаб
Розроб.		Чижан В. В.						
Перев.		Марціняк Г. Я.			UML-діаграма варіантів використання програми	Архив	Архив	1
Г.контр.								
Рецензент								
Нхконтр.		Примічак В.А.						
Затв.								
						ВСП ТФК ТНТУ КН-421 м. Тернопіль		
						Копія		
						Формат А1		

Таблиця техніко-економічних показників

№	Показник	Одиниця вимірювання	Значення
1	Середовище розробки	–	Visual Studio Code
2	Тип інтерфейсу	–	Веб-інтерфейс
3	Frontend	–	React
4	Backend	–	Node.js / Express.js
5	База даних	–	MongoDB Atlas
6	Хмарне сховище зображень	–	Cloudinary
7	Загальний розмір програми	МБ	61,4
8	Содівартість розробки	грн	70936
9	Плановий річний грошовий потік	грн	39724
10	Ціна програмного продукту	грн	110660
11	Чиста теперішня вартість	грн	28050
12	Термін окупності	рік	1,8

				2026.KBP.122.4212100.00 ТБ				
Зм.	Арх.	№ док.	Підп.	Дата	Розробка вебсайту зомозину "Ліпка"	Лит	Маса	Масштаб
Розроб.	Чокан В. В.							
Перев.	Марціняк Г. Я.				Таблиця техніко-економічних показників			
І.контр.						Архив	Архив	1
Рецензент						ВСП ТФК ТНТУ КН-421		
І.контр.	Приймак В.А.					м. Тернопіль		
Затв.						Формат А1		