

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка інтернет-магазину Dolittle

Виконав: студент IV курсу, групи СН-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Гайда С.-Н.О.

(прізвище та ініціали)

Керівник

(підпис)

Щербак Л.М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Голотенко О.С.

(прізвище та ініціали)

Тернопіль  
2024

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(прізвище та ініціали)

«   » 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Гайді Софії-Наталії Олегівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтернет-магазину Dolittle

Керівник роботи Щербак Леонід Миколайович, д.т.н., професор кафедри КН  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 29 » квітня 2024 року № 4/7-470.

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Літературні та інтернет джерела.

4. Зміст роботи

Вступ. 1. Аналіз вимог та постановка задачі розробки інтернет-магазину Dolittle. 1.1 Аналіз предметної області. 1.2. Формування вимог інтернет-магазину. 1.3. Варіанти використання для інтернет-магазину. 1.4. Вибір середовища розробки. 1.5. Обґрунтування вибраного стеку технологій. 1.7. Висновок до першого розділу. 2. Проектування інтернет-магазину Dolittle. 2.1. Архітектура інтернет-магазину. 2.2. Аналіз функціональної моделі. 2.3. Моделювання схем для інтернет-магазину. 2.4. Структура бази даних. 2.5. Бібліотеки для розробки клієнтської та серверної частин проекту. 2.6. Файлова структура проекту. 2.7. Обґрунтування кольорової схеми інтерфейсу користувача. 2.8. Висновок до другого розділу. 3. Реалізація інтернет-магазину Dolittle. 3.1 Розробка серверної частини на базі CMS Strapi. 3.2. Розробка клієнтської частини на базі фреймворка React. 3.3. Наповнення інформацією про товари бази даних інтернет-магазину. 3.4. Розгортання програмної інфраструктури інтернет-магазину. 3.5.Валідація розмітки інтерфейсу користувача та тестування інтернет-магазину.3.6. Висновок до третього розділу. 4. Безпека життєдіяльності, основи охорони праці. 4.1. До лікарська допомога при ураженні електричним струмом. 4.2. Психологічне розвантаження для працівників. 4.3. Висновок до четвертого розділу. Висновки. Перелік джерел. 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці			

7. Дата видачі завдання 29 січня 2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	Виконано
2.	Підбір джерел про розробку інтернет-магазинів	31.01.2024-03.02.2024	Виконано
3.	Опрацювання джерел по темі кваліфікаційної роботи	04.02.2024-06.02.2024	Виконано
4.	Виконання дослідження щодо інтернет-магазинів Розроблення інтернет-магазину Dolittle	07.02.2024-11.02.2024	Виконано
5.	Оформлення розділу «Аналіз вимог та постановка задачі інтернет-магазину Dolittle»	03.06.2024-05.06.2024	Виконано
6.	Оформлення розділу «Проектування інтернет- магазину Dolittle»	06.06.2024-08.06.2024	Виконано
7.	Оформлення розділу «Реалізація інтернет-магазину Dolittle»	09.06.2024-11.06.2024	Виконано
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»	12.06.2024-13.06.2024	Виконано
9.	Виконання завдання до підрозділу «Основи охорони праці»	14.06.2024-15.06.2024	Виконано
10.	Оформлення кваліфікаційної роботи	16.06.2024-17.06.2024	Виконано
11.	Нормоконтроль	18.06.2024-19.06.2024	Виконано
12.	Перевірка на плагіат	20.06.2024	Виконано
13.	Попередній захист кваліфікаційної роботи	21.06.2024	Виконано
14.	Захист кваліфікаційної роботи		

Студент

\_\_\_\_\_ (підпис)

Гайда С.-Н.О.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Щербак Л.М.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Розробка інтернет-магазину Dolittle // Кваліфікаційна робота освітнього рівня «Бакалавр» // Гайда Софія-Наталія Олегівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2024 // С. – 70, рис. – 31, табл. – 3, слайди – 14, додат. – 8, бібліогр. – 39.

**Ключові слова:** інтернет-магазин, React, CMS Strapi, товари для тварин, електронна комерція, UI/UX дизайн, Lighthouse.

У першому розділі виконано аналіз предметної області, сформульовано вимоги до розробки інтернет-магазину, побудовано та досліджено діаграму варіантів використання. Також обрано середовище розробки та проаналізовано переваги використаного стеку технологій.

Другий розділ присвячено проектуванню архітектури інтернет-магазину. Тут описано функціональні моделі, змодельовано схеми інтернет-магазину, відображено структуру бази даних, досліджено основні бібліотеки та файлової структуру проєкту. Також обґрунтовано вибір кольорової схеми та шаблонів для інтерфейсу користувача.

У третьому розділі детально описано процес реалізації інтернет-магазину Dolittle. Зокрема, розроблено серверну частину на базі CMS Strapi та клієнтську частину на базі фреймворка React. Здійснено наповнення даними бази даних, описано процес розгортання проєкту на сервері, а також проведено валідацію розмітки інтерфейсу користувача та тестування функціоналу в цілому.

У розділі «Безпека життєдіяльності, основи охорони праці» розглянуто алгоритм надання першої допомоги при ураженні електричним струмом та рекомендації щодо допомоги постраждалому. Також описуються методи нематеріальної мотивації. Розділ підкреслює важливість здоров'я працівників, пропонуючи рішення для зниження стресу та підвищення мотивації.

## ANNOTATION

Development of the Dolittle Online Store // Qualification work of the education level “Bachelor” // Haida Sofiia-Nataliia Olehivna // Ternopil Ivan Pului National Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Science Department, group SN–41 // Ternopil, 2024 // P. – 70, fig. – 31, tabl. – 3, slides – 14, annexes. – 8, references – 39.

**Keywords:** online store, React, CMS Strapi, products for animals, e-commerce, UI/UX design, Lighthouse.

In the first section, an analysis of the subject area was conducted, requirements for the development of the online store were formulated, and a use case diagram was constructed and examined. Additionally, the development environment was selected, and the advantages of the utilized technology stack were analyzed.

The second section is dedicated to the design of the online store's architecture. It describes the functional models, models the online store's schemas, displays the database structure, and examines the main libraries and the project's file structure. The choice of the color scheme and templates for the user interface is also justified.

The third section provides a detailed description of the implementation process of the Dolittle online store. Specifically, the server-side was developed using the CMS Strapi, and the client-side was developed using the React framework. The database was populated with data, the project deployment process on the server was described, and the validation of the user interface markup and functional testing were carried out.

The «Life Safety, Basics of Occupational Safety» section examines the algorithm for providing first aid in case of electric shock and offers recommendations for assisting the victim. It also describes methods of non-material motivation. The section emphasizes the importance of employees' health, proposing solutions for stress reduction and increased motivation.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API (англ. Application programming interface) – прикладний програмний інтерфейс.

Axios – JS-бібліотека для виконання HTTP запитів в Node.js.

CMS (англ. Content Management System) – система управління контентом.

CSS (англ. Cascading Style Sheets) – каскадні таблиці стилів.

HTML (англ. HyperText Markup Language) – мова гіпертекстової розмітки.

HTTP (англ. HyperText Transfer Protocol) – протокол прикладного рівня передачі даних.

React – відкрита JS-бібліотека, призначена для розробки користувацьких інтерфейсів.

URL (англ. Uniform Resource Locator) – унікальний ідентифікатор, який використовується для пошуку ресурсу в Інтернеті.

VS Code – Visual Studio Code.

БД – база даних.

ПЗ – програмне забезпечення.

ТЗ – технічне завдання.

Фреймворк – ПЗ, яке полегшує розробку шляхом поєднання в собі різних функціональних можливостей.

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1 АНАЛІЗ ВИМОГ ТА ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ DOLITTLE .....	10
1.1 Аналіз предметної області.....	10
1.2 Формування вимог до проєкту інтернет-магазину .....	12
1.3 Варіанти використання для інтернет-магазину.....	13
1.4 Вибір середовища розробки .....	16
1.5 Огрунтування вибраного стеку технологій .....	18
1.6 Висновок до першого розділу .....	19
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ DOLITTLE .....	20
2.1 Архітектура інтернет-магазину.....	20
2.2 Аналіз функціональної моделі .....	23
2.3 Моделювання схем для інтернет-магазину .....	24
2.4 Структура бази даних.....	30
2.5 Бібліотеки для розробки клієнтської та серверної частин проєкту ...	32
2.6 Файлова структура проєкту.....	35
2.7 Обгрунтування кольорової схеми інтерфейсу користувача.....	36
2.8 Висновок до другого розділу .....	38
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ DOLITTLE .....	39
3.1 Розробка серверної частини на базі CMS Strapi.....	39
3.2 Розробка клієнтської частини на базі фреймворка React.....	42
3.3 Наповнення інформацією про товари бази даних інтернет-магазину	50
3.4 Розгортання програмної інфраструктури інтернет-магазину .....	52
3.5 Валідація розмітки інтерфейсу користувача та тестування інтернет- магазину.....	53
3.6 Висновок до третього розділу .....	56
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .	57

4.1 Долікарська допомога при ураженні електричним струмом .....	57
4.2 Психофізіологічне розвантаження для працівників. ....	59
4.3 Висновок до четвертого розділу .....	62
ВИСНОВКИ.....	63
ПЕРЕЛІК ДЖЕРЕЛ.....	65
ДОДАТКИ	



## ВСТУП

**Актуальність теми.** Інтернет-магазин є ключовим інструментом взаємодії між продавцем та покупцем в галузі сучасної електронної комерції. За останні роки інтернет-торгівля стрімко розвивалася, інтернет-магазини стали основним каналом продажів для багатьох компаній і підприємств. Цей тренд свідчить про те, що віртуальна торгівля стає домінуючим способом здійснення покупок для споживачів у всьому світі [1]. Відкриття інтернет-магазину дозволяє компаніям залучати більше клієнтів, збільшувати обсяги продажів і масштабуватися глобально, не обмежуючись географічними межами. Тому розробка нових інтернет-магазинів залишається надзвичайно важливим завданням для бізнесу, розв'язання якого забезпечить йому конкурентоздатність та дозволить успішно функціонувати і розвиватися в сучасному цифровому середовищі.

**Мета і задачі дослідження.** Метою кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка проєкту інтернет-магазину. Для досягнення мети потрібно розв'язати певні задачі:

- проаналізувати обрану предметну область та сформулювати вимоги до проєкту;
- визначити оптимальні варіанти використання;
- спроектувати архітектуру та базу даних інтернет-магазину;
- розробити клієнтську та серверну частини проєкту відповідно до поставлених вимог;
- виконати розгортання інфраструктури проєкту на сервері;
- провести повне тестування усіх функціональних можливостей інтернет-магазину.

**Об'єкт дослідження.** Процес проєктування та розробки інтернет-магазину з використанням сучасних інструментів, технологій та методологій.

**Предмет дослідження.** Аналіз вимог, проєктування архітектури, реалізація функціоналу, розгортання, тестування та впровадження інтернет-магазину.

**Методи дослідження.** Для реалізації проєкту використано сучасні інструменти, методології та технології розробки. Зокрема, ключовою технологією в проєкті є React, який використовується для створення користувацького інтерфейсу інтернет-магазину. Також застосовано Strapi для управління даними та контентом інтернет-магазину. Третьою ключовою технологією в проєкті є PostgreSQL, що використовується як система управління базами даних для зберігання та організації даних, які використовуються в інтернет-магазині.

Тестування інтерфейсу користувача здійснено за допомогою інструменту Lighthouse. З метою оцінки рівня продуктивності та якості вебсторінок. Lighthouse надає аналіз різних аспектів інтернет-магазину, включаючи швидкість завантаження, доступність та інші показники продуктивності. Крім того, використовувався інструмент W3C Validator для валідації CSS, щоб перевірити правильність написання стилів відповідно до стандартів W3C та виявити синтаксичні помилки.

**Практичне значення одержаних результатів.** Розроблений в результаті виконання кваліфікаційної роботи інтернет-магазин спрямований на ефективну співпрацю з клієнтами. Він забезпечує клієнтам інтернет-магазину зручний інтерфейс для перегляду асортименту товарів, швидкий пошук потрібних позицій та зручність оформлення замовлення. Крім того, інтернет-магазин Dolittle покликаний підтримувати інтерактивний зв'язок з клієнтами через зручні способи зв'язку, щоб допомагати клієнтам зробити правильний вибір. Розроблений проєкт сприятиме росту клієнтської бази, популяризації онлайн-торгівлі та підвищенню лояльності клієнтів до бренду через задоволення від покупок.

## РОЗДІЛ 1 АНАЛІЗ ВИМОГ ТА ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ DOLITTLE

### 1.1 Аналіз предметної області

Інтернет-магазин – це вебплатформа, в якій доступна можливість здійснення покупок для користувачів. Під час розробки інтернет-магазину необхідно провести аналіз предметної області з метою забезпечення ефективного та задовільного досвіду в онлайн-покупках для клієнтів.

Одним із ключових компонентів інтернет-магазину є зручний інтерфейс, який дозволяє користувачам легко орієнтуватися по сайту, здійснювати пошук товарів та ознайомлюватися з їх деталями. Інтернет-магазину необхідно дотримуватися чіткої структури категорій товарів, пошукову систему, фільтри та сортування, щоб допомогти користувачам швидко знайти потрібний їм товар.

Окрім цього, в інтернет-магазині має бути можливість реєстрації та авторизації користувачів. Це дозволяє кожному користувачеві створити особистий обліковий запис, зберігати інформацію про замовлення, відстежувати їх статус та здійснювати оплату. Авторизованим користувачам може надаватись доступ до особистих налаштувань, таких як адреси доставки, історія замовлень та збережені способи оплати.

Детальний виклад інформації про товари є ще одним аспектом успішного інтернет-магазину. Кожен товар повинен мати свою сторінку з описом, характеристиками, фотографіями та ціною. Додатково, можуть бути надані відгуки та оцінки від інших користувачів, що допомагають прийняти рішення щодо покупки.

Функціонал кошика покупок є необхідним компонентом інтернет-магазину. Користувачі мають можливість додавати товари до кошика, вибирати кількість одиниць товару, видаляти або змінювати їх кількість перед здійсненням покупки. Крім того, необхідно забезпечити розрахунок вартості покупки, включаючи податки, вартість доставки та інші додаткові витрати.

Інтернет-магазини роблять товари доступнішими для покупців, адже їх можна купити не виходячи з дому. Клієнти мають змогу замовляти товари з інтернет-магазинів у зручний для них момент. Це особливо зручно для людей з особливими потребами або тих, хто мешкає в віддалених районах.

Також однією з головних переваг інтернет-магазинів є різноманітний вибір товарів. Оскільки вони не обмежені фізичним простором, інтернет-магазини можуть пропонувати великий вибір товарів у порівнянні зі звичайними магазинами. Клієнти можуть легко переглядати різні варіанти товарів, порівнювати їх характеристики та ціни, що допомагає їм знайти найкращу пропозицію. Клієнти можуть шукати товари за ключовими словами, використовувати фільтри для точного підбору, додавати товари до кошика та здійснювати оплату онлайн. Багато інтернет-магазинів пропонують зручну доставку до дверей клієнтів, що забезпечує швидкий та зручний спосіб отримання придбаних товарів.

Інтернет-магазини, незважаючи на свої переваги, також мають деякі недоліки в порівнянні зі звичайними магазинами [2]. Першим недоліком є відсутність можливості фізичного огляду товару. Клієнти не можуть взяти товар у руки, приміряти одяг або перевірити його якість перед покупкою. Вони повинні покладатися на фотографії та описи товарів, що може бути менш точним і не дозволяє зробити повноцінне рішення про покупку.

Другим недоліком є ризик не доставки або пошкодження товару. Інтернет-магазини залежать від служб доставки, існує ризик затримки, помилкової доставки або пошкодження під час транспортування. Це може призвести до незадоволеності клієнтів, нестачі товару або потреби в поверненні та обміні, що може бути незручним та зайняти більше часу.

Ще одним недоліком при покупках в інтернет-магазинах є потенційний ризик безпеки банківської картки. Коли клієнти роблять покупки в Інтернеті, їм часто потрібно вводити дані своєї картки, включаючи номер картки, термін дії та код безпеки CVV. Ці дані можуть бути піддані ризику крадіжки або злочинного

використання, особливо якщо інтернет-магазин не має належних заходів безпеки.

## 1.2 Формування вимог до проєкту інтернет-магазину

Для того, щоб сформулювати ТЗ інтернет-магазину потрібно спочатку описати його основні функціональні можливості [3]. Існують кілька комплексних вимог для формування ТЗ:

- Можливість авторизації та реєстрації користувачів.
- Мінімум кроків для здійснення замовлення.
- Можливість шукати, ставити вподобайки та додавати в кошик товари.
- Можливість надання швидкого відгуку від інтернет-магазину.
- Можливість крос-платформності.

Структурні особливості інтернет-магазину полягають:

- Приватність списків вподобаних товарів та кошика.
- Обмежений доступ до даних входу та замовлень.
- Клієнтська та адміністративна частина мають бути поєднані.
- Невеликий рівень вкладеності.
- Товари мають бути розділені за категоріями, але без надмірної деталізації [4].
- Сторінки авторизації та реєстрації повинні бути винесені на окремі сторінки інтернет-магазину.

Також варто згадати про вимоги до розробки інтерфейсу інтернет-магазину Dolittle:

- Інтерфейс має бути простим у використанні для всіх категорій користувачів, незалежно від їхнього рівня технічної підготовки. Меню і організація структури сайту повинні бути логічними і зрозумілими для того, щоб користувачі могли легко знаходити необхідні товари та інші розділи.

- Можливість швидкого пошуку та фільтрації товарів за різними параметрами, такими як категорія, підкатегорія, новий товар, ціновий діапазон.

– Інтерфейс інтернет-магазину повинен оптимізуватися для використання на мобільних пристроях, забезпечуючи зручний та ефективний доступ до всіх функцій.

– Інтернет-магазин повинен належним чином відображатися і працювати у різних веббраузерах, наприклад, Chrome, Opera, Mozilla Firefox, Safari, Microsoft Edge тощо.

– Зареєстрованим користувачам необхідно впровадити особистий кабінет, де вони зможуть переглядати історію замовлень, змінювати особисті дані та налаштування облікового запису.

– Інтеграція різних способів оплати та вибір способів доставки товарів (кур'єрська доставка, самовивіз, поштові послуги) для зручності клієнтів.

– Користувачам має бути надана можливість залишати відгуки та оцінювати товари, а також переглядати відгуки інших користувачів для прийняття обдуманого рішення при покупці.

– Зручний механізм додавання товарів до кошика з автоматичним розрахунком загальної вартості покупки сприятиме зручності користувачів та полегшить процес покупки.

– Інтернет-магазин має містити контактну інформацію для зв'язку з магазином та можливість швидкої зворотного зв'язку через онлайн-чат або електронну пошту для вирішення будь-яких питань чи проблем користувачів.

При розробці інтернет-магазину необхідно враховувати всі вищезазначені вимоги, щоб забезпечити ефективну співпрацю інтернет-магазину зі своїми клієнтами.

### **1.3 Варіанти використання для інтернет-магазину**

Метою цього підрозділу є визначити функціональність системи та спрогнозувати можливі сценарії взаємодії між акторами. Цей етап допомагає підвищити якість та ефективність інтернет-магазину, забезпечуючи відповідність потребам користувачів [5]. Інтернет-магазин включає двох дійових

осіб, які взаємодіють з нею: клієнт і менеджер. На рисунку 1.1 представлено діаграму варіантів використання для актора «Клієнт».

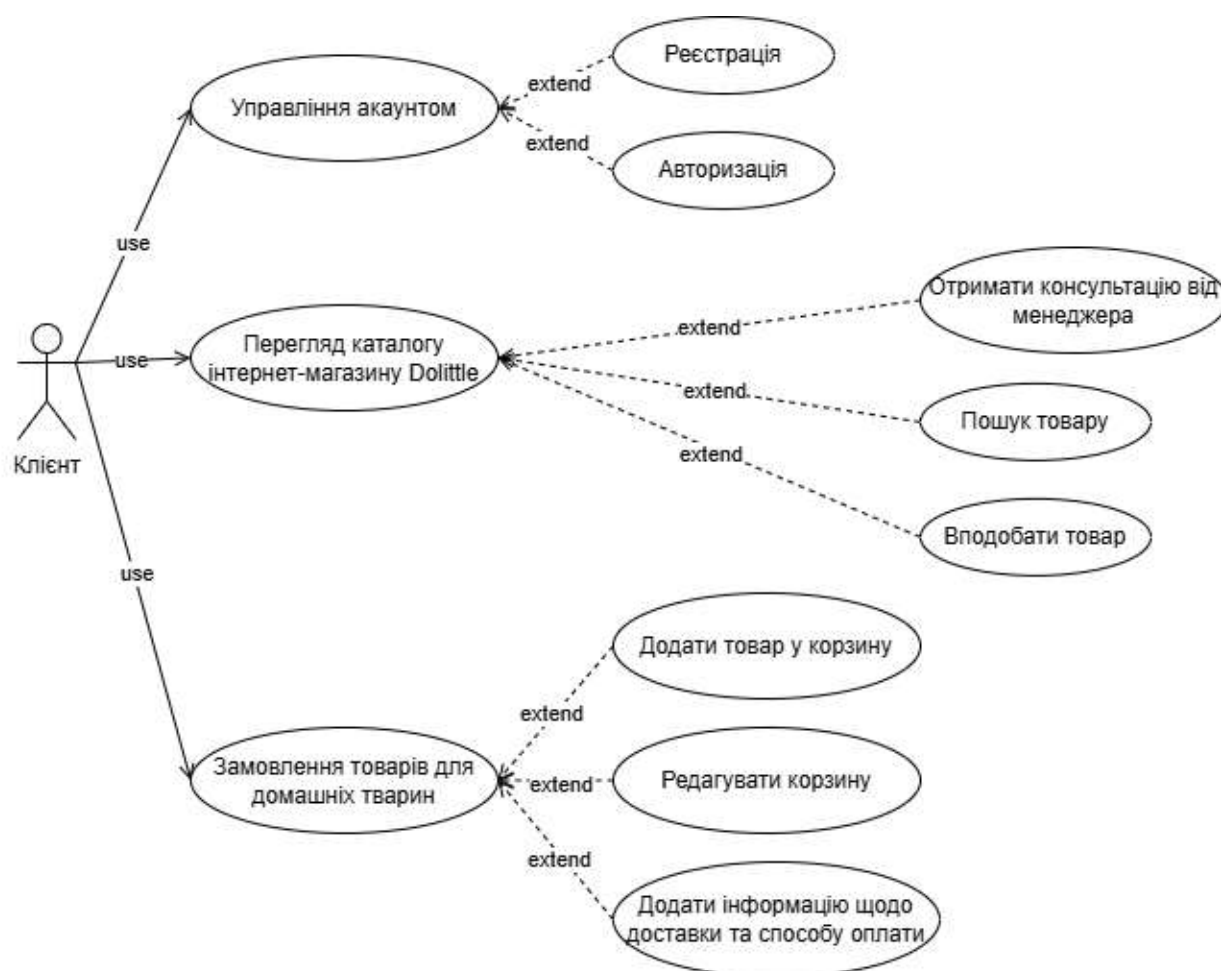


Рисунок 1.1 – Діаграма варіантів використання для актора «Клієнт» інтернет-магазину Dolittle

Інтернет-магазин повинен реалізовувати вищезазначений функціонал для актора «Клієнт». Це включає можливість реєстрації та авторизації, управління акаунтом, перегляд каталогу товарів, отримання консультацій від менеджера, пошук та вподобання товарів, додавання товарів у кошик, редагування кошика, а також додавання інформації щодо доставки та способу оплати під час замовлення товарів для домашніх тварин.

Короткий опис варіантів використання для актора «Клієнт» наведено в таблиці 1.1.

Таблиця 1.1 – Варіанти використання для актора «Клієнт»

Актор	Найменування	Пояснення
Клієнт	Управління акаунтом	Дозволяє користувачу авторизуватись або зареєструватись, щоб отримати додаткові функції такі, як перегляд історії замовлень.
	Перегляд каталогу інтернет-магазину Dolittle	Клієнт може переглядати, вподобати, фільтрувати та шукати товари, а також отримувати консультацію від менеджера.
	Замовлення товарів для домашніх тварин	Клієнт може додавати товари в корзину, редагувати її вміст і здійснювати замовлення, вказуючи адресу доставки, спосіб оплати та персональну інформацію.

На рисунку 1.2 представлено діаграму варіантів використання для актора «Менеджер» інтернет-магазину Dolittle.

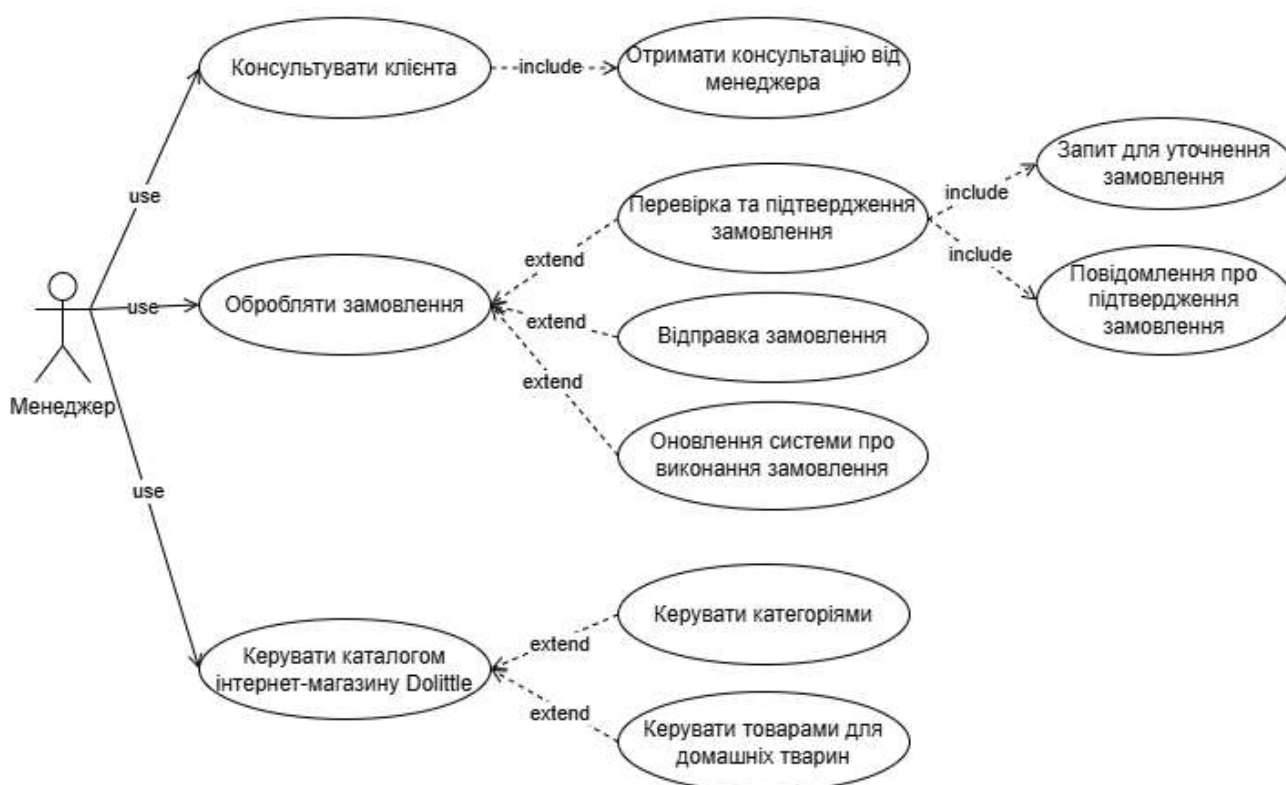


Рисунок 1.2 – Діаграма варіантів використання для актора «Менеджер» інтернет-магазину Dolittle



Інтернет-магазин має виконувати відповідні функції для актора «Менеджер». Короткий опис ВВ для актора «Менеджер» наведено в таблиці 1.2.

Таблиця 1.2 – Варіанти використання для актора «Менеджер»

<b>Актор</b>	<b>Найменування</b>	<b>Пояснення</b>
Менеджер	Консультувати клієнта	Менеджер надає клієнту вичерпні відповіді на запитання щодо товару, його характеристик, використання, ціни, терміну придатності тощо.
	Обробляти замовлення	До цього варіанту використання входить перевірка та підтвердження замовлень, відправка замовлення, оновлення системи про виконання замовлення.
	Керувати каталогом інтернет-магазину Dolittle	Менеджер також має можливість керувати категоріями та товарами для домашніх тварин, включаючи їх додавання, редагування та видалення.

Ці таблиці надають інформацію про різні варіанти використання для акторів «Клієнт» і «Менеджер» в контексті інтернет-магазину Dolittle. Аналізуючи таблиці варіантів використання для різних акторів, можна усвідомити взаємодію цих акторів з системою.

#### 1.4 Вибір середовища розробки

Для розробки інтернет-магазину важливо обрати оптимальне середовище розробки, яке дозволить зручно створювати, тестувати і підтримувати вебдодатки. Два з найпопулярніших вибору серед розробників – WebStorm від JetBrains та VS Code від Microsoft.

WebStorm є інтегрованим середовищем розробки, спеціально призначене для роботи з вебтехнологіями. Воно включає широкий асортимент інструментів для розробки вебзастосунків, включаючи підтримку JavaScript, HTML, CSS, і фреймворки як Angular, React, та інші. WebStorm має корисні функції, зокрема

автоматичне доповнення коду, рефакторинг, інспекцію коду та інші інструменти для покращення продуктивності розробника.

VS Code – це легкий, але потужний розширюваний інструментарій для розробки ПЗ з відкритим вихідним кодом. Середовище розробки сумісне з широким спектром мов програмування, серед яких JavaScript, TypeScript, HTML, CSS, і багато інших [6]. Одним з головних переваг VS Code є його широкий екосистема розширень, яка дозволяє налаштувати і доповнити його функціональність під конкретні потреби розробки.

Щоб провести об'єктивне порівняння середовищ розробки WebStorm і VS Code, можна скласти таблицю з основними характеристиками кожного з них (таблиця 1.3).

Таблиця 1.3 – Основні характеристики WebStorm та VS Code

<b>Характеристика</b>	<b>WebStorm</b>	<b>VS Code</b>
Підтримувані мови	JavaScript, HTML, CSS, TypeScript, інші вебтехнології	JavaScript, TypeScript, Python, Java, C#, інші мови
Спеціалізація	Веброзробка	Універсальне середовище, яке підтримує безліч мов програмування
Розширюваність	Можливості розширення обмежені, але є плагіни для популярних фреймворків	Широка екосистема розширень, дозволяє налаштувати практично будь-які функції
Швидкодія	Вимагає більше ресурсів, може бути менш швидким на великих проєктах	Легкий і швидкий, забезпечує швидку реакцію на дії користувача
Функціональність	Багатий набір інструментів для веброзробки, автодоповнення, інспекція коду, рефакторинг	Широкий вибір функцій, включаючи контроль версій, дебагер, автодоповнення, рефакторинг
Спільнота	Менша спільнота користувачів, але активна підтримка веброзробників	Велика спільнота користувачів, багато активних розробників і підтримка

Для кваліфікаційної роботи обрано VS Code. Враховуючи його легкість використання, велику спільноту користувачів, розширену підтримку різних мов програмування і широкий вибір розширень, VS Code забезпечує більш гнучку і універсальну платформу для розробки. Також, його зручний інтерфейс дозволяє швидко адаптуватися і почати працювати з проектом без великих зусиль. Цей вибір базується на потребах проекту та особистих перевагах, що забезпечують ефективну роботу розробника.

### **1.5 Огрунтування вибраного стеку технологій**

При виборі технологій для розробки інтернет-магазину обрано React, PostgreSQL і Strapi з огляду на їхні переваги та підходи до створення потужного та масштабованого вебдодатку.

React був обраний для клієнтської частини інтернет-магазину через його модульність, ефективність і активну спільноту розробників. React дозволяє побудувати інтерактивний і динамічний інтерфейс користувача, що легко масштабується і підтримується. Компонентний підхід React сприяє чистоті коду і прискорює розробку, дозволяючи розробникам використовувати повторювані компоненти, що зменшує кількість помилок і полегшує підтримку коду. Крім того, React забезпечує високу продуктивність завдяки віртуальному DOM і ефективному управлінню станом проекту, що робить його ідеальним вибором для створення сучасних інтернет-магазинів [7].

PostgreSQL був вибраний для бази даних інтернет-магазину через його надійність, можливості оптимізації та сумісність з багатьма іншими інструментами [8]. PostgreSQL забезпечує повнотекстовий пошук, транзакційну безпеку та гнучкість в роботі з складними структурами даних, що робить його оптимальним вибором для зберігання інформації про товари, замовлення та користувачів.

Strapi був обраний як CMS для управління даними і контентом інтернет-магазину. Завдяки тому, що Strapi побудований на базі Node.js, він відмінно

підходить для інтеграції з React та іншими frontend технологіями, що дозволяє розробникам створювати сучасні та ефективні інтерфейси користувача. Більше того, гнучкість у налаштуванні API дозволяє пристосувати функціональність Strapi до конкретних потреб інтернет-магазину та дозволяє ефективно керувати контентом магазину без необхідності написання власного backend з нуля [9].

Застосування цих технологій забезпечує високу продуктивність, гнучкість та масштабованість інтернет-магазину Dolittle, що в кінцевому підсумку сприяє підвищенню задоволеності користувачів і спрощує процес розробки та підтримки системи.

## **1.6 Висновок до першого розділу**

У першому розділі кваліфікаційної роботи досліджено предметну область інтернет-магазину, а саме описано переваги та їх недоліки порівняно із звичайними магазинами. Визначено основні вимоги до розробки інтернет-магазину.

При описі варіантів використання виділено два актори, а саме клієнт та менеджер, описано їх взаємозв'язки з інтернет-магазином. Клієнт здійснює дії, пов'язані з переглядом, вибором та замовленням товарів, тоді як менеджер відповідає за обробку замовлень, консультації та управління асортиментом магазину. Такий підхід дозволяє чітко окреслити ролі та функції кожного актора, забезпечуючи зручність та ефективність взаємодії з інтернет-магазином.

Вибір технологій та середовища розробки був обґрунтований на основі їхньої ефективності, зручності та відповідності сучасним стандартам веброзробки. Обране середовище VS Code забезпечує інтеграцію з численними розширеннями, полегшує налагодження та підтримує сучасні інструменти розробки, що робить його ідеальним вибором для створення якісного програмного забезпечення. Технології розробки включають React для frontend, Strapi як CMS для управління даними, та PostgreSQL для бази даних, що забезпечує надійне зберігання і обробку інформації.

## РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ DOLITTLE

### 2.1 Архітектура інтернет-магазину

Щоб забезпечити ефективний та організований процес розробки, інтернет-магазин Dolittle буде побудований на базі тривірневої архітектури, яка включає представлення, бізнес-логіку та доступ до даних [10]. На рисунку 2.1 наведена структура тривірневої архітектури для інтернет-магазину Dolittle.

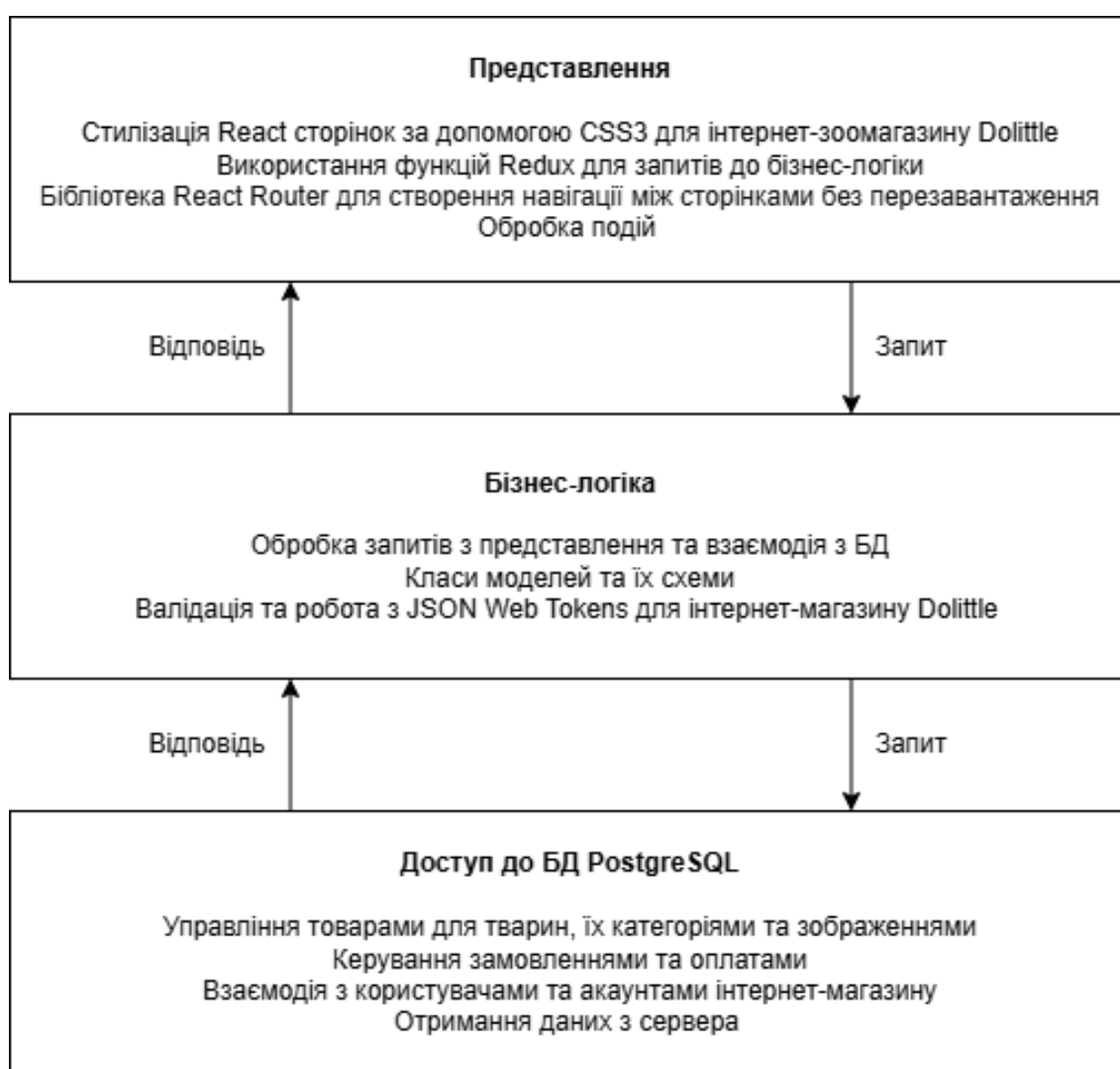


Рисунок 2.1 – Структура тривірневої архітектури для інтернет-магазину Dolittle

Схема складається з трьох основних рівнів: представлення, бізнес-логіка і доступ до бази даних.

Перший рівень – представлення. Тут відбувається стилізація сторінок за допомогою CSS3, використання Redux для запитів до бізнес-логіки, бібліотека React Router забезпечує навігацію між сторінками без перезавантаження. Також тут обробляються події.

Другий рівень – бізнес-логіка. Він відповідає за обробку запитів з рівня представлення та взаємодію з базою даних. Тут визначаються класи моделей та їх схеми, а також здійснюється валідація і робота з JSON Web Tokens.

Третій рівень – доступ до бази даних PostgreSQL. На цьому рівні відбувається управління товарами для тварин, їх категоріями та зображеннями, керування замовленнями та платежами, взаємодія з користувачами та акаунтами магазину, а також отримання даних з сервера [11].

Трирівнева архітектура є оптимальною для інтернет-магазинів, оскільки не має значення, з чого починати: спершу розробляти серверну частину і підключити до бази даних, а потім клієнтську частину, або навпаки [12]. Вона дозволяє розробляти обидві частини паралельно, завдяки чому розробка може бути більш ефективною і швидкою. На рисунку 2.2 наведено трирівневу архітектуру з впровадженими технологіями.

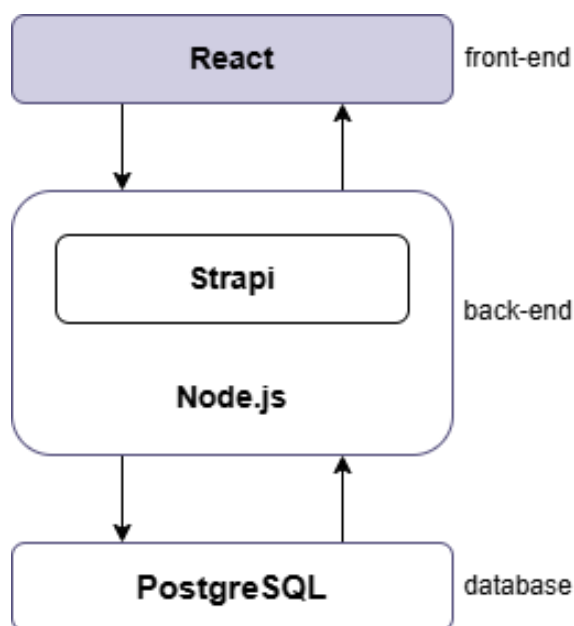


Рисунок 2.2 – Трирівнева архітектура з впровадженими технологіями

Трирівнева клієнт-серверна архітектура включає три основні компоненти: представлення (клієнтська частина), бізнес-логіка (серверна частина) та доступ до даних (БД). Для аналізу клієнт-серверної архітектури взято наукову статтю, яка має назву «Програмне забезпечення на основі клієнт-серверної архітектури для обліку реалізації товарів в торгівлі» [13].

У розробці інтернет-магазину Dolittle, представлення відповідає за створення інтерфейсу користувача та взаємодію з ним [14]. Цей рівень включає вебсторінки, компоненти і модулі, що відображають інформацію про товари, корзину покупок, форми замовлень та інші елементи, які необхідні для коректної взаємодії з інтернет-магазином. За допомогою React, ми створюємо компоненти, які обробляють логіку відображення даних, взаємодіють з користувачем і забезпечують зручний інтерфейс для покупок та навігації в магазині.

Бізнес-логіка відповідає за обробку даних та виконання бізнес-правил магазину. Вона забезпечує перевірку даних під час реєстрації чи оформлення замовлення, обробку замовлень, таких як перевірка наявності товарів, розрахунок вартості та оновлення статусу замовлення. Логіка взаємодії з базою даних включає створення запитів для збереження замовлень, оновлення цін та перевірки існування користувачів. Такий підхід забезпечує ефективну та надійну роботу з даними. Для реалізації бізнес-логіки використовується Strapi на платформі Node.js, що дозволяє швидко розробляти API та обробляти HTTP-запити.

Доступ до даних забезпечує взаємодію з базою даних PostgreSQL. Цей рівень включає в себе моделі даних, які відображають структуру і зв'язки між об'єктами даних, а також об'єктно-реляційне відображення (ORM) для зручної роботи з базою даних через Node.js. Тут реалізована логіка доступу до даних, яка включає створення і виконання SQL-запитів для збереження нових записів, оновлення існуючих даних або отримання інформації з бази даних для подальшого використання в інтернет-магазині. Цей рівень забезпечує ефективну роботу з даними і їхню надійність для інтернет-магазину.

## 2.2 Аналіз функціональної моделі

Функціональна схема для інтернет-магазину розробляється для узгодження та візуалізації основних функцій та процесів, які мають забезпечити інтернет-магазин [15]. Вона є важливим інструментом для комунікації між розробниками, дизайнерами, менеджерами проєкту та замовниками, допомагаючи забезпечити розуміння та узгодження щодо того, як повинен функціонувати інтернет-магазин перед початком активної розробки (рис. 2.3).

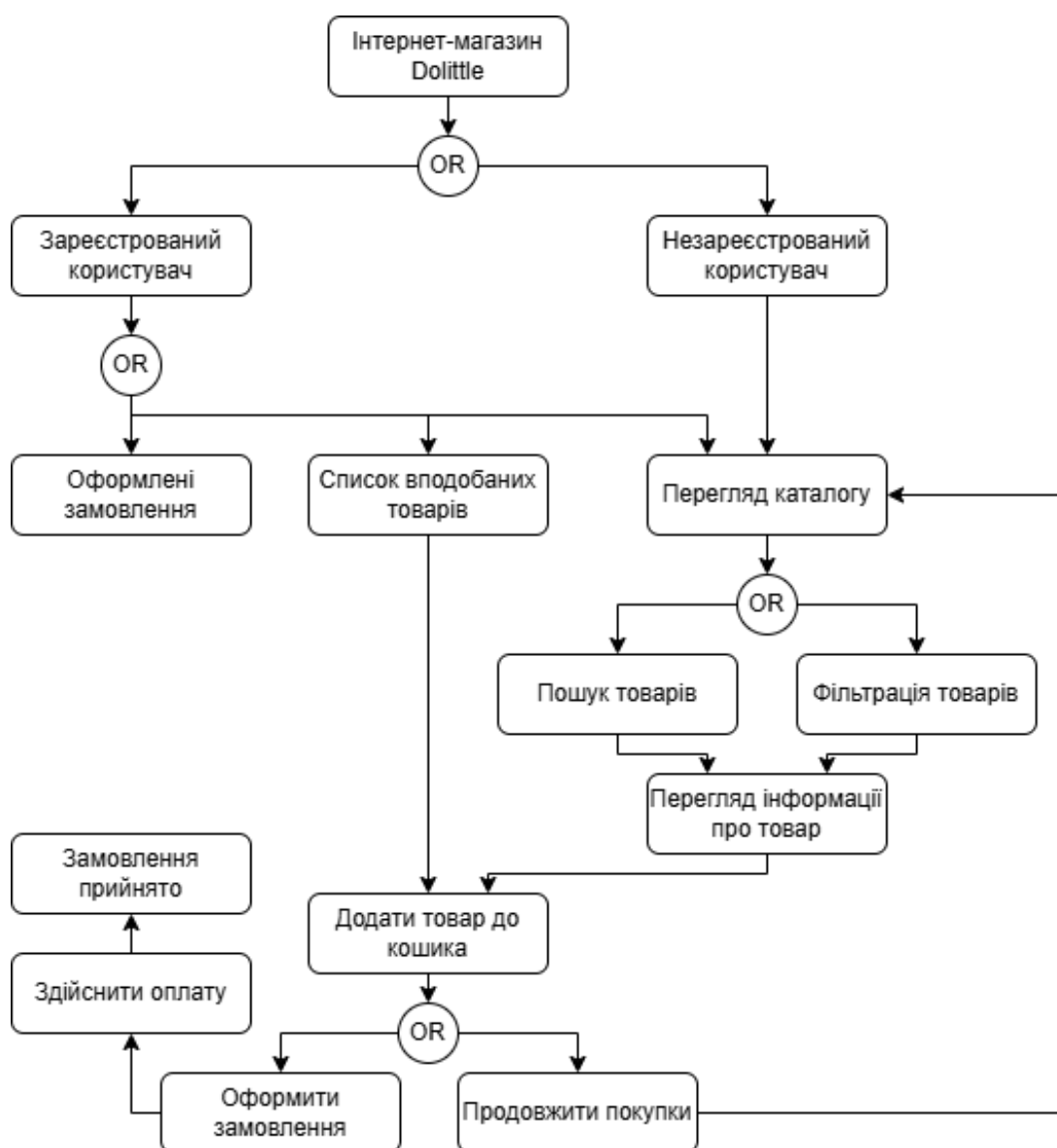


Рисунок 2.3 – Функціональна схема для інтернет-магазину Dolittle



Після побудови функціональної схеми, опишемо процес оформлення замовлення в інтернет-магазині.

Зареєстрований користувач має більше можливостей в інтернет-магазині: доступ до оформлених замовлень, списку вподобань і каталогу товарів з зручним пошуком та фільтрацією за різними критеріями. Клієнт може додавати товари з каталогу до кошика.

Незареєстрований користувач може переглядати каталог, використовувати пошук і фільтрацію, а також додавати товари до кошика, але не має доступу до оформлених замовлень і списку вподобань. Для цього потрібно зареєструватися або увійти в систему.

У кошику користувач має можливість переглянути обрані товари, а також здійснювати редагування цих товарів. Він може змінювати кількість обраних одиниць кожного товару або видаляти їх з кошика. Після внесення змін користувач може продовжити покупки, додавши ще товарів до кошика, або перейти до оформлення замовлення. В процесі замовлення товарів користувач повинен вказати дані для доставки, такі як адреса доставки та контактні дані. Він також обирає метод оплати, наприклад, кредитною карткою, платіжною системою або готівкою при отриманні. Після введення всіх необхідних даних і підтвердження замовлення, інтернет-магазин оформлює замовлення зі статусом "прийняте". Після успішного оформлення замовлення користувач може отримувати інформацію про статус виконання замовлення, наприклад, про підготовку до відправлення, доставку чи отримання. Інтернет-магазин повідомляє користувача про будь-які зміни статусу замовлення, забезпечуючи зручне відстеження та контроль за процесом покупок.

### **2.3 Моделювання схем для інтернет-магазину**

Роль діаграми станів в контексті інтернет-магазину полягає у візуалізації складних процесів, взаємодій між користувачем і системою, що дозволяє легше розуміти і моделювати ці процеси. Ця діаграма допомагає виявляти можливі

проблеми або покращення в інтерфейсі користувача та оптимізувати роботу системи, забезпечуючи більш зручний та ефективний досвід покупців [13].

На рисунку 2.4 відображено діаграму станів для інтернет-магазину Dolittle.

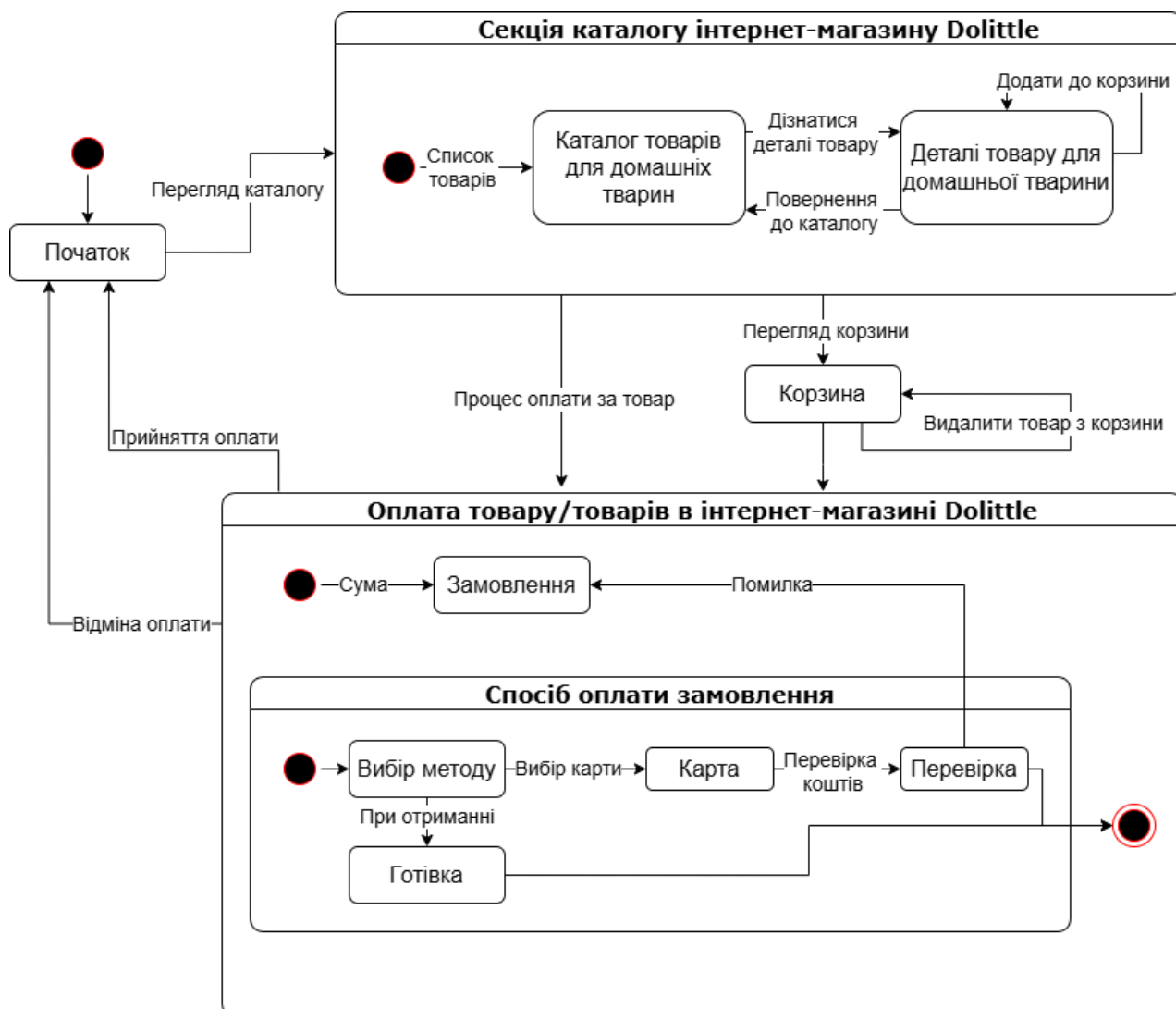


Рисунок 2.4 – Діаграма станів для інтернет-магазину Dolittle

Наступним етапом моделювання схем для інтернет магазину є створення діаграма послідовності (рис. 2.5). Діаграма послідовності ілюструє, як взаємодіють об'єкти системи у певній послідовності часу, що відповідає виконанню певних дій або операцій [16]. Ця діаграма дозволяє візуалізувати послідовність викликів методів та обмін повідомленнями між об'єктами в рамках конкретного сценарію взаємодії.

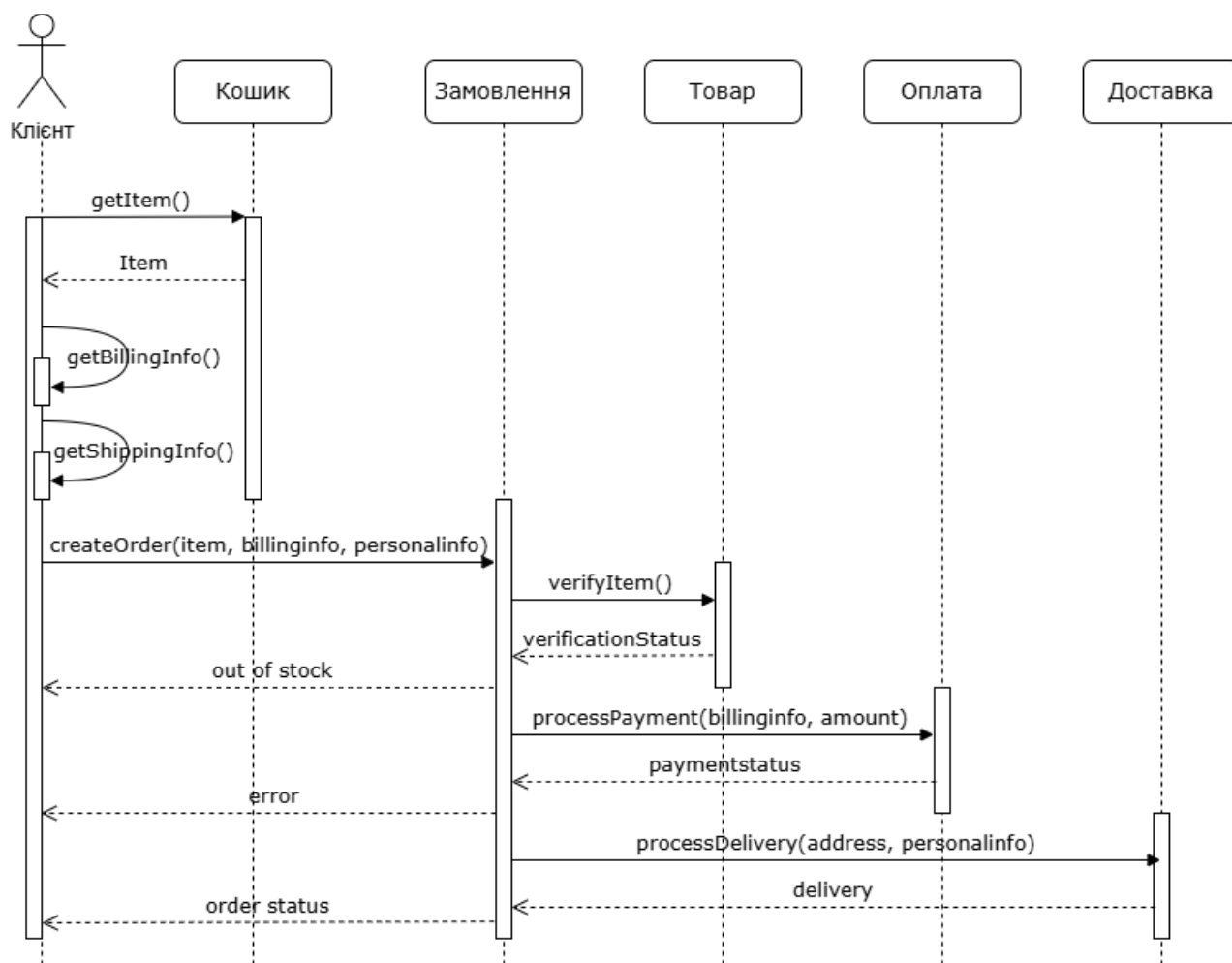


Рисунок 2.5 – Діаграма послідовності «оформлення замовлення» для інтернет-магазину

Діаграма послідовності «оформлення замовлення» включає взаємодію між клієнтом, кошиком для покупок, товаром, замовленням, оплатою та доставкою. Процес починається з того, що клієнт надсилає запит до кошика для отримання обраних товарів. Кошик надсилає клієнту запит на платіжну адресу для отримання платіжної інформації. Додатково надсилається запит на отримання інформації про доставку.

Після цього об'єкт замовлення надсилає запит на створення замовлення, включаючи дані про товар, платіжну інформацію та персональну інформацію клієнта. Замовлення перевіряє наявність товару на складі та повідомляє про статус перевірки. Якщо товар відсутній, клієнту надсилається відповідне повідомлення. В іншому випадку надсилається запит на обробку платежу та

здійснення оплати. Після обробки платежу надходить відповідь зі статусом транзакції. У разі помилки клієнт отримує повідомлення про невдачу оплати. Якщо оплата пройшла успішно, відправляється запит до об'єкта доставки для обробки відправлення товару, і повертається відповідь зі статусом доставки. В кінці клієнта інформують про остаточний статус замовлення.

Також діаграму послідовності можна подати для процесу пошуку товару (рис. 2.6).

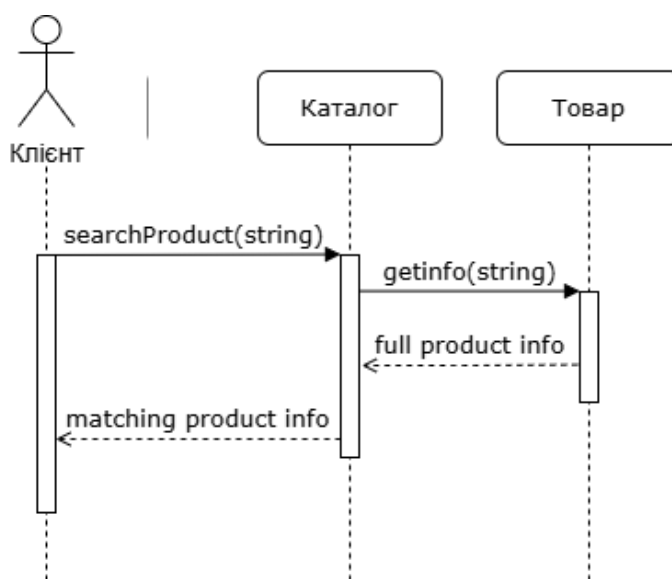


Рисунок 2.6 – Діаграма послідовності «пошук товару» для інтернет-магазину

Для онлайн-покупок створимо діаграму послідовності для випадку використання «пошуку товару». У цій діаграмі беруть участь: клієнт, каталог та товар. Клієнт надсилає запит до каталогу для отримання інформації про товар. Каталог обробляє запит і повертає клієнту повну інформацію про знайдений товар.

Також для інтернет-магазину корисно розробити діаграму класів (рис. 2.7). Діаграма класів допомагає візуалізувати структуру системи за допомогою класів, їх властивостей, методів та взаємозв'язків між ними. Це дозволяє розробникам краще зрозуміти внутрішню логіку та структуру ПЗ, визначити залежності й взаємодію між різними компонентами, а також полегшує подальший розвиток і підтримку системи [17].

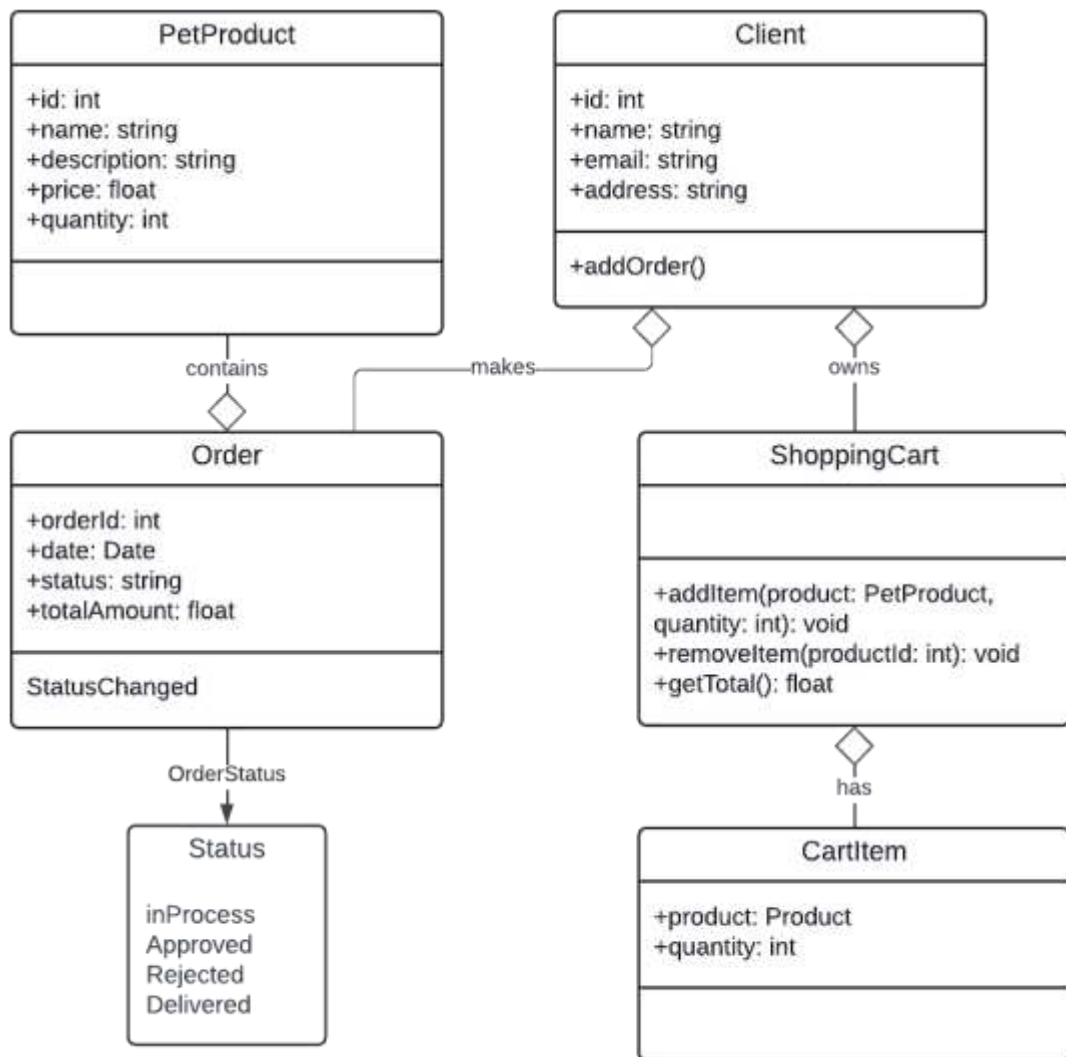


Рисунок 2.7 – Діаграма класів для інтернет-магазину Dolittle

Діаграма класів, наведена на рисунку, представляє структуру інтернет-магазину з п'ятьма основними класами: PetProduct, Client, Order, ShoppingCart та CartItem, а також один клас Status для статусів замовлень.

Клас PetProduct інтернет-магазину:

- Містить атрибути: id (ідентифікатор товару), name (назва товару), description (опис товару), price (ціна товару) та quantity (кількість на складі).
- Цей клас використовується для зберігання інформації про товари, які доступні для покупки.

Клас Client інтернет-магазину:

- Містить атрибути: id (ідентифікатор клієнта), name (ім'я клієнта), email (електронна адреса клієнта) та address (адреса клієнта).

- Має метод `addOrder()`, який дозволяє клієнту створити нове замовлення.
- Відповідає за зберігання даних про клієнтів інтернет-магазину та управління їх замовленнями.

Клас `Order` інтернет-магазину:

- Містить атрибути: `orderId` (ідентифікатор замовлення), `date` (дата замовлення), `status` (статус замовлення) та `totalAmount` (загальна сума замовлення).
- Цей клас відповідає за зберігання та обробку інформації про замовлення.
- Подія `StatusChanged` генерується при зміні статусу замовлення.
- Має асоціації з класами `Client` та `Product`, що відображає зв'язок між клієнтом та замовленими товарами.

Клас `ShoppingCart` інтернет-магазину:

- Містить методи: `addItem(product: Product, quantity: int)` для додавання товарів до кошика, `removeItem(productId: int)` для видалення товарів з кошика, та `getTotal()` для отримання загальної суми товарів у кошику.
- Відповідає за управління вмістом кошика покупця перед оформленням замовлення.
- Має асоціацію з класом `CartItem`, що представляє окремі товари в кошику.

Клас `CartItem` інтернет-магазину:

- Містить атрибути: `product` (товар) та `quantity` (кількість товару).
- Відповідає за зберігання інформації про конкретні товари та їх кількість у кошику.

Клас `Status` інтернет-магазину:

- Є перерахуванням (`enum`) з можливими статусами замовлень: `inProcess` (в процесі), `Approved` (затверджено), `Rejected` (відхилено) та `Delivery` (доставка).
- Використовується для відстеження та управління станом замовлень.

Ця діаграма класів відображає взаємозв'язки між основними компонентами інтернет-магазину, показуючи, як клієнти взаємодіють з товарами, кошиком та замовленнями, а також як ці елементи взаємодіють між собою.

## 2.4 Структура бази даних

Спочатку для проєктування БД потрібно описати основні типи сутностей інтернет-магазину [18].

На рисунку 2.8 зображено модель «Сутність-Зв'язок» в інтернет-магазині, також відома як ER-діаграма БД.

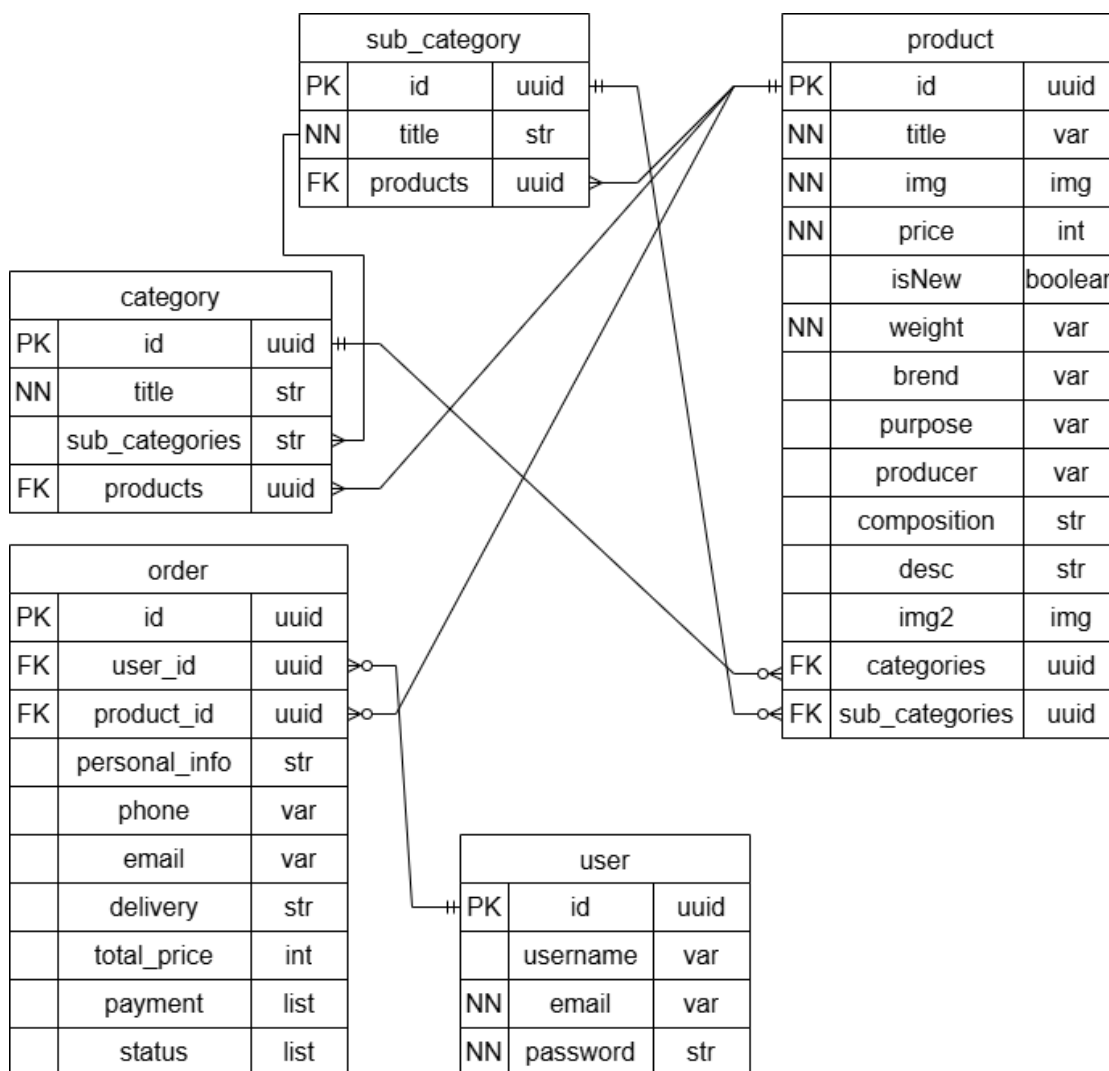


Рисунок 2.8 – ER-діаграма БД для інтернет-магазину Dolittle

Category – сутність, яка інформує про категорії товарів для домашніх тварин. Вона містить такі атрибути, як унікальний ідентифікатор (id), назву категорії (title), список товарів (products), які належать до цієї категорії, та

підкатегорії (sub\_categories). Ця сутність дозволяє структурувати асортимент товарів, що полегшує користувачам навігацію та пошук необхідних товарів.

SubCategory – сутність, яка інформує про підкатегорії товарів, що входять до складу певних категорій. Вона включає атрибути, такі як унікальний ідентифікатор (id), список товарів (products), які належать до підкатегорії. Це дозволяє створити більш детальну ієрархічну структуру товарів для кращої організації і зручності користувачів.

Product – сутність, яка надає детальну інформацію про товари, включаючи їх приналежність до певних категорій та підкатегорій. Основні атрибути включають унікальний ідентифікатор (id), назву товару (title), зображення (img), ціну (price), індикатор новизни (isNew), вагу (weight), бренд (brend), призначення (purpose), виробника (producer), склад (composition), опис (desc), додаткове зображення (img2), та приналежність до категорій (categories) і підкатегорій (sub\_categories). Ця сутність забезпечує повну інформацію про кожен товар, що дозволяє користувачам приймати обґрунтовані рішення про покупку.

User – сутність зареєстрованого користувача інтернет-магазину. Вона включає такі атрибути: унікальний ідентифікатор (id), ім'я користувача (username), електронну пошту (email), пароль (password). Ця інформація дозволяє ідентифікувати користувачів, керувати їх доступом до різних функцій магазину, а також забезпечувати безпеку та конфіденційність їхніх даних.

Order – сутність, яка надає інформацію про оформлені замовлення користувачів та їх дані. Вона містить атрибути, такі як унікальний ідентифікатор замовлення (id), список товарів (product\_id), зареєстрованого користувача (user\_id), персональну інформацію про замовника (personal\_info), номер телефону (phone), електронну пошту (email), інформацію про доставку (delivery), спосіб оплати (payment) та статус замовлення (status). Ця сутність забезпечує повний опис кожного замовлення, що дозволяє ефективно обробляти та відстежувати замовлення, забезпечуючи якісний сервіс для користувачів інтернет-магазину.



## 2.5 Бібліотеки для розробки клієнтської та серверної частин проєкту

Основні бібліотеки включають різноманітні компоненти і технології, які використовуються для побудови, розгортання та функціонування інтернет-магазину. Вони створюють основу для функціонування інтернет-магазинів, гарантуючи зручний та безпечний процес онлайн-покупок для користувачів.

Залежності інтернет-магазину визначають його архітектуру, швидкодію та надійність. Вони забезпечують необхідну інфраструктуру для роботи магазину та забезпечують зручний і безпечний досвід для користувачів під час покупок онлайн. На рисунку 2.9 відображено список залежностей клієнтської частини для інтернет-магазину Dolittle.

```
"dependencies": {  
  "@babel/runtime": "^7.23.2",  
  "@react-oauth/google": "^0.12.1",  
  "@reduxjs/toolkit": "^1.9.7",  
  "axios": "^1.6.0",  
  "bootstrap": "^5.3.2",  
  "react": "^18.2.0",  
  "react-bootstrap": "^2.9.1",  
  "react-dom": "^18.2.0",  
  "react-redux": "^8.1.3",  
  "react-router-dom": "^6.16.0",  
  "react-scripts": "^5.0.1",  
  "react-select": "^5.8.0",  
  "react-toastify": "^9.1.3",  
  "reactstrap": "^9.2.1",  
  "redux": "^4.2.1",  
  "redux-persist": "^6.0.0",  
  "sass": "^1.68.0"  
},
```

Рисунок 2.9 – Список бібліотек клієнтської частини для інтернет-магазину Dolittle

Зробимо детальний опис кожної залежності зі списку клієнтської частини інтернет-магазину:

- @babel/runtime (7.23.2). Залежність використовується для підтримки нових функцій JavaScript, які не підтримуються старішими версіями браузерів.

Вона дозволяє трансформувати сучасний код JavaScript до формату, який зрозумілий більшості браузерів.

- `@react-oauth/google` (0.12.1). Бібліотека дозволяє реалізувати аутентифікацію користувачів через Google OAuth в додатку React. Вона надає зручний інтерфейс для авторизації користувачів за допомогою облікових записів Google.

- `@reduxjs/toolkit` (1.9.7). Бібліотека надає інструменти для розробки і керування станом додатку за допомогою Redux.

- `axios` (1.6.0). За допомогою бібліотеки Axios можна легко робити HTTP-запити до вебсерверів. Вона дозволяє виконувати асинхронні запити і отримувати дані з сервера [19].

- `bootstrap` (5.3.2), `react-bootstrap` (2.9.1), `reactstrap` (9.2.1). Бібліотеки CSS використовуються для стилізації інтерфейсу користувача. Також вони містять готові компоненти і стилі.

- `react-dom` (18.2.0). React DOM використовується для відображення React компонентів в DOM браузера.

- `react-router-dom` (6.16.0). Бібліотека для реалізації навігації і маршрутизації в React.

- `react-select` (5.8.0). Надає компоненти для створення випадаючих списків і вибору елементів зі списку.

- `react-toastify` (9.1.3). Дозволяє показувати повідомлення про успішні або помилкові дії в додатку.

- `redux-persist` (6.0.0): Ця бібліотека дозволяє зберігати стан Redux в локальному сховищі браузера, щоб зберігати дані між сеансами користувача.

- `sass` (1.68.0). Sass - це препроцесор CSS, який дозволяє використовувати змінні, вкладені стилі, міксини та інші функції для полегшення стилізації компонентів.

Також варто згадати про залежності для серверної частини. Залежності в цьому контексті включають різноманітні бібліотеки, фреймворки та сервіси, які забезпечують функціонування серверної логіки інтернет-магазину.

На рисунку 2.10 зображено список залежностей серверної частини для інтернет-магазину Dolittle.

```
"dependencies": {
  "@strapi/plugin-cloud": "4.20.1",
  "@strapi/plugin-i18n": "4.20.1",
  "@strapi/plugin-users-permissions": "4.20.1",
  "@strapi/provider-upload-cloudinary": "^4.20.1",
  "@strapi/strapi": "4.20.1",
  "pg": "8.8.0",
  "react": "^18.0.0",
  "react-dom": "^18.0.0",
  "react-router-dom": "5.3.4",
  "styled-components": "5.3.3"
},
```

Рисунок 2.10 – Список залежностей серверної частини для інтернет-магазину Dolittle

Ось обґрунтування кожної залежності зі списку серверної частини інтернет-магазину:

- @strapi/plugin-cloud (4.20.1). Залежність дозволяє інтегрувати хмарні сервіси з Strapi для зберігання та керування мультимедійним контентом, таким як зображення та відео, у інтернет-магазині.

- @strapi/plugin-i18n (4.20.1). Плагін забезпечує підтримку i18n в Strapi, що дозволяє створювати багатомовні контент і керувати перекладами в додатку.

- @strapi/plugin-users-permissions (4.20.1). Розширює можливості автентифікації і авторизації в Strapi, дозволяючи створювати різні ролі користувачів і керувати правами доступу [20].

- @strapi/provider-upload-cloudinary (4.20.1). Інтегрує сервіс Cloudinary з Strapi для управління зображеннями та мультимедійним контентом в інтернет-магазині.

- styled-components (5.3.3). Дозволяє використовувати CSS в JavaScript, що спрощує стилізацію компонентів React. Вона дозволяє створювати реактивні стилі для компонентів з допомогою вбудованих JavaScript шаблонів.

– pg (8.8.0). Бібліотека дозволяє взаємодіяти з базою даних PostgreSQL з допомогою Node.js.

Ці залежності необхідні для розробки повноцінного інтернет-магазину з серверної та клієнтської частинами, забезпечуючи керування користувачами та їхніми правами, завантаження та оптимізацію мультимедійного контенту, взаємодію з базою даних і створення користувацького інтерфейсу на React зі стилізованою компонентною архітектурою.

## 2.6 Файлова структура проєкту

Подібно до дерева, файлова структура інтернет-магазину організована ієрархічно. Є головна папка (корінь дерева), яка містить підпапки (гілки), що в свою чергу можуть містити інші підпапки або файли. Ця структура відображає логічні взаємозв'язки між компонентами і ресурсами інтернет-магазину (рис. 2.11).

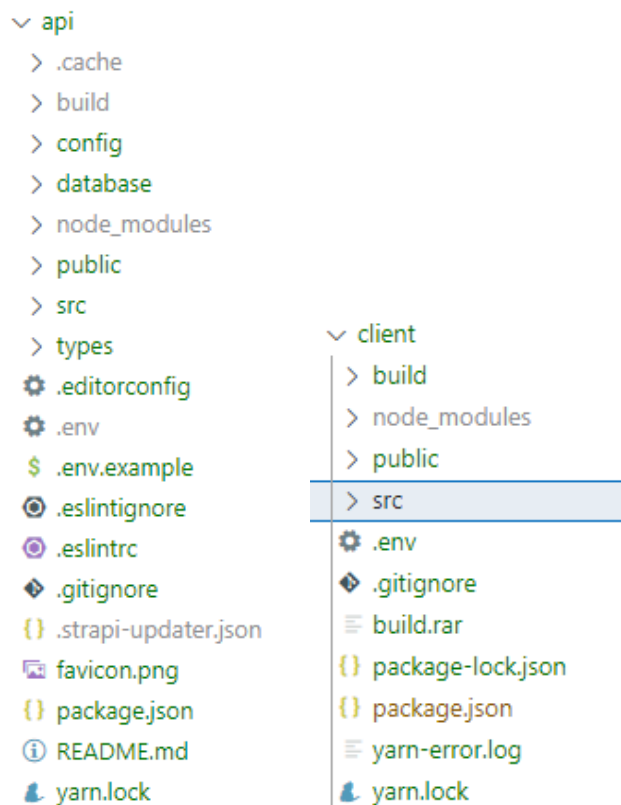


Рисунок 2.11 – Файлова структура інтернет-магазину Dolittle

Опишемо основні папки і файли інтернет-магазину:

- `.strapi` – папка містить конфігураційні файли, пов'язані з проєктом Strapi.
- `build` – у цій папці зберігаються файли, які створюються або компілюються під час збірки проєкту.
- `config` – конфігураційні файли для Strapi, такі як налаштування сервера або бази даних. `database` – файли, пов'язані з базою даних, такі як схеми БД або файли міграції. `node_modules` – усі залежності проєкту, які встановлені за допомогою `yarn`.
- `public` – статичні файли, які доступні для загального доступу, наприклад, зображення, CSS-файли або файли JavaScript. `src` – основна папка, де розміщений вихідний код серверної або клієнтської частини.
- `.editorconfig` – конфігураційний файл для редактора коду, який визначає стиль форматування коду для всього проєкту.
- `.env` – файл конфігурації, де зберігаються змінні середовища для налаштування проєкту (наприклад, з'єднання з БД).
- `.eslintignore` – конфігураційний файл ESLint для виключення файлів або папок з перевірки стилю коду. `eslinttrc` – конфігураційний файл для ESLint, який визначає правила перевірки стилю коду.
- `.gitignore` – файл, який вказує Git, які файли або папки ігнорувати під час відстеження змін в репозиторії Git. `package.json` – файл конфігурації проєкту Node.js, де зберігаються метадані про проєкт та список його залежностей.

Файлова структура спрощує розробку, тестування, супровід та розгортання інтернет-магазину, організовуючи код та ресурси згідно зі стандартами та кращими практиками розробки програмного забезпечення.

## 2.7 Обґрунтування кольорової схеми інтерфейсу користувача

Загальний підхід до вибору кольорової схеми повинен враховувати естетичні та маркетингові аспекти, спрямовані на створення позитивного іміджу та відчуття товарів для домашніх тварин у інтернет-магазині Dolittle [21].

Колір #00A60E (зелений) часто асоціюється з природою, свіжістю та життєвим енергетичним потенціалом. Його використання може підкреслити екологічні аспекти товарів для домашніх тварин, створюючи враження здорового та екологічно чистого підходу до догляду за тваринами. Зелений колір був використаний у розробці інтернет-магазину Dolittle для логотипу, кнопок, іконки корзини, зображень категорій.

Колір #654321 (темний коричневий) може викликати асоціації з природою, землею та натуральністю. Використання цього кольору може додати теплоти та відчуття земного зв'язку, підкреслюючи натуральність і надійність товарів для домашніх тварин. Під час розробки інтернет-магазину Dolittle цей колір був застосований для заголовків, деяких допоміжних зображень.

Чорний колір, як класичний і елегантний, додає контрасту та витонченості до дизайну. Використання чорного кольору може підкреслити важливість товарів і привертати увагу клієнтів.

Після кольорової схеми варто описати структуру шаблонів інтерфейсу для інтернет-магазину. Шаблони дизайну були розроблені вручну з використанням принципів UI-дизайну.

Головною складовою вибору певного розташування елементів на головній сторінці була зручність для клієнтів інтернет-магазину (див. додаток Ж). Навігаційне меню розташоване у верхній частині сторінки, що є стандартним та інтуїтивно зрозумілим для клієнтів. Пошуковий рядок знаходиться на видному місці, що полегшує користувачам пошук конкретних товарів. Іконки категорій товарів, розташовані після банеру, забезпечують швидкий доступ до основних категорій продукції. Використання великого, жирного шрифту для заголовків та більш дрібного для додаткової інформації забезпечує чітку і зрозумілу ієрархію інформації [22].

На сторінці каталогу кожен товар містить назву, короткий опис та ціну, що дозволяє користувачам отримати основну інформацію без необхідності переходу на іншу сторінку (див. додаток З). Це знижує кількість кроків для здійснення покупки [23]. Розташування фільтрів та категорій у лівій частині сторінки

дозволяє користувачам легко знайти потрібний товар. Ліва колонка зазвичай є стандартом для вебдизайну, оскільки користувачі звикли до цього розташування. Фільтри за ціною, категоріями, та підкатегоріями допомагають швидко звузити пошук до потрібного товару.

Узагальному дизайн сторінок інтернет-магазину адаптований для різних розмірів екрану, що забезпечує зручність використання на мобільних пристроях та планшетах. Відсутність зайвих елементів дозволяє зосередити увагу на товарах, що полегшує прийняття рішення про покупку. Усі елементи дизайну (ікони, кнопки, шрифти, кольори) відповідають загальній стилістиці сайту та створюють відчуття цілісності та професійності.

## **2.8 Висновок до другого розділу**

У другому розділі всебічно розглянуто та реалізовано всі етапи проєктування інтернет-магазину Dolittle. Це включає технічні аспекти, такі як архітектура, бази даних та файлової структури, а також аспекти користувацького досвіду такі, як інтерфейс та кольорова схема.

Моделювання схем дало можливість візуалізувати та спланувати взаємодію між різними компонентами системи. Це включає діаграму станів, діаграми послідовності та діаграму класів. Діаграма станів допомагає відобразити зміни станів об'єктів у системі, діаграми послідовності ілюструють порядок взаємодій між об'єктами, а діаграма класів показує статичну структуру системи, включаючи класи та їх взаємозв'язки. Розробка структури БД була зображена за допомогою моделі "Сутність-Зв'язок", яка включала створення таблиць, визначення їх взаємозв'язків та обмежень, що забезпечують цілісність даних.

Отримані результати демонструють завершене проєктування інтернет-магазину Dolittle з урахуванням всіх важливих аспектів, що забезпечують задоволення потреб клієнтів, зручність користування інтернет-магазином та ефективність роботи системи.

## РОЗДІЛ 3 РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ DOLITTLE

### 3.1 Розробка серверної частини на базі CMS Strapi

Для розробки серверної частини використовується CMS Strapi. Щоб створити проєкт потрібно запустити сценарій встановлення та обрати спосіб налаштування (рис. 3.1). Інсталяція quick start налаштовує Strapi з базою даних SQLite. Для інтернет-магазину обрано БД PostgreSQL, тому замість quick start потрібно обрати custom.

```

yarn create v1.22.19
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Installed "create-strapi-app@4.24.2" with binaries:
  - create-strapi-app
? Choose your installation type Custom (manual settings)
You are running Node.js 19.9.0
Strapi only supports LTS versions of Node.js, other versions may not be compatible.
? Choose your preferred language (Use arrow keys)
? Choose your preferred language JavaScript
? Choose your default database client
? Choose your default database client postgres
? Database name: dolittledb_t08w
? Database name: dolittledb_t08w
? Host: dpq-cp49f721hb1s73etui7g-a.oregon-postgres.render.com
? Host: dpq-cp49f721hb1s73etui7g-a.oregon-postgres.render.com
? Port: 5432
? Port: 5432
? Username: dolittledb_user
? Username: dolittledb_user
? Password: *****
? Password: *****
? Enable SSL connection: (y/N) n
? Enable SSL connection: No

Creating a project with custom database options.
```

Рисунок 3.1 – Налаштування Strapi для інтернет-магазину

Зображення демонструє послідовність кроків для створення нового проєкту Strapi з підключенням до бази даних PostgreSQL за допомогою Yarn.



Відображається попередження, що Strapi підтримує лише LTS (Long Term Support) версії Node.js. Вибір мови програмування обрано JavaScript, а бази даних – PostgreSQL. Після вибору БД проведено налаштування: запис назви, порту, хоста, паролю, назви клієнта. Після введення всіх необхідних даних, система повідомляє про створення проєкту з кастомними налаштуваннями бази даних.

Strapi надає повнофункціональний інтерфейс командного рядка (CLI), який дозволяє швидко створювати та керувати проєктами. Основними командами CLI для реалізації Strapi є `develop`, `build` та `start` [24]. Детальна інформація про кожну з них:

- `develop` – команда запускає Strapi-додаток у режимі розробки з увімкненим автоматичним перезавантаженням. Це означає, що будь-які зміни у вашому коді автоматично застосовуватимуться без необхідності вручну перезапускати сервер.

- `start` – команда запускає Strapi-додаток у продуктивному режимі з вимкненим автоматичним перезавантаженням. Цей режим оптимізований для виконання на живому сервері, де стабільність важливіша за зручність розробки.

- `build` – команда збирає адміністративну панель Strapi для розгортання. Вона компілює весь необхідний код і ресурси для адміністрування, що дозволяє забезпечити оптимальну продуктивність і зменшити час завантаження.

Ознайомимось детальніше з виконанням команди `start`. Після запуску цієї команди отримаємо проєктну інформацію та активні дії (рис. 3.2).

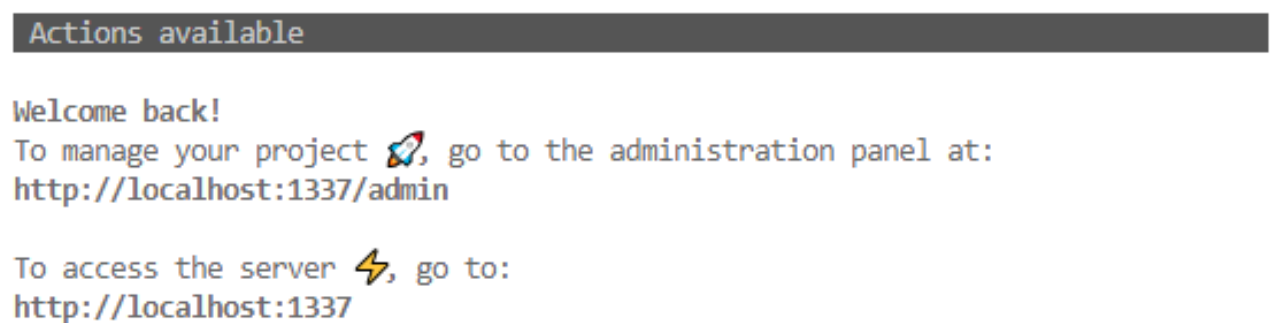


Рисунок 3.2 – Посилання для входу як адміністратор

Після переходу за посиланням панелі адміністратора, користувач бачить вітальну форму для входу в Strapi. Це стандартна сторінка входу, яка запрошує користувача увійти в свій обліковий запис Strapi, щоб отримати доступ до адміністративної панелі.

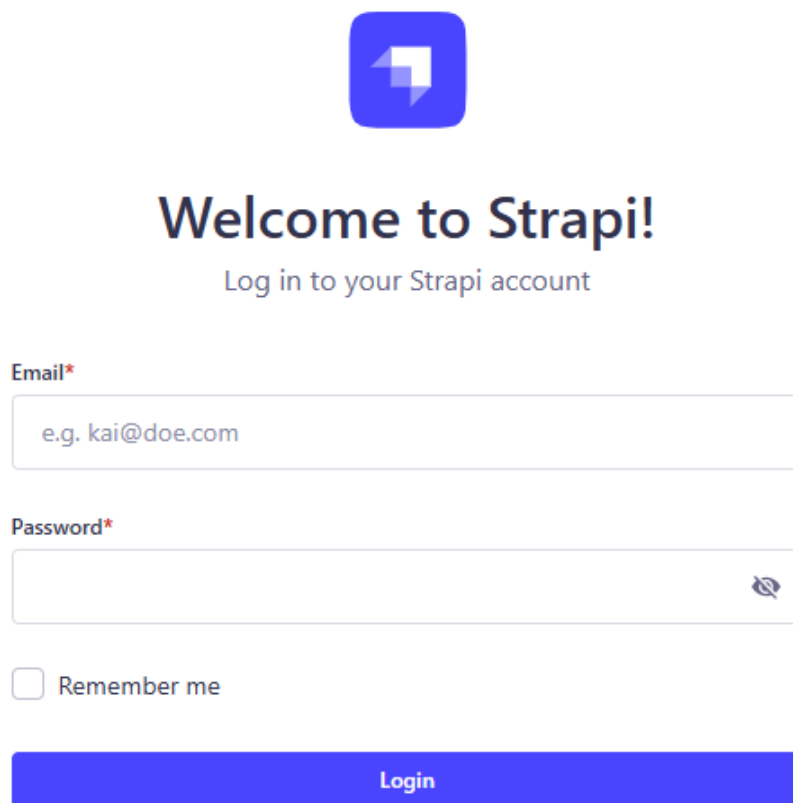


Рисунок 3.3 – Вхід в панель адміністратора інтернет-магазину

На сторінці входу в панель адміністратора розміщено логотип Strapi у верхній частині, що допомагає користувачу швидко ідентифікувати платформу. Для входу в систему, користувачу потрібно ввести свою електронну адресу та пароль у відповідні поля форми.

Під полем для введення пароля є чекбокс "Remember me", який дозволяє користувачу зберегти свої облікові дані для майбутніх входів, що спрощує процес доступу до панелі адміністратора. У нижній частині форми розташована кнопка "Login", натиснувши на яку, користувач може виконати вхід в систему. Якщо введені облікові дані правильні, користувач буде перенаправлений на

панель адміністратора, де він зможе керувати контентом і налаштуваннями свого Strapi-додатку.

### 3.2 Розробка клієнтської частини на базі фреймворка React

Після ініціалізації серверної частини інтернет-магазину, можна почати розробку компонентів користувацького інтерфейсу, підключення до серверної частини через Axios, та налаштування маршрутизації для забезпечення плавної навігації по інтернет-магазину. Для створення нового проєкту на основі React.js потрібно запустити термінал та прописати команду «`npx create-react-app client`» (рис. 3.4).

```
PS E:\ТНТУ\Диплом\D> npx create-react-app client
Need to install the following packages:
  create-react-app@5.0.1
Ok to proceed? (y) y
npm WARN deprecated fstream-ignore@1.0.5: This package is no longer supported.
npm WARN deprecated uid-number@0.0.6: This package is no longer supported.
npm WARN deprecated fstream@1.0.12: This package is no longer supported.
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.

Creating a new React app in E:\ТНТУ\Диплом\D\client.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1490 packages in 6m

259 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.

Installing template dependencies using npm...
```

Рисунок 3.4 – Виконання команди для створення проєкту на базі React.js

Після виконання цієї команди в терміналі, вказана папка «client» буде створена і заповнена всіма початковими файлами та структурами, необхідними для початку розробки на React. Це включає в себе початковий код, конфігураційні файли, а також базові залежності, такі як React та React-DOM.

Цей процес значно спрощує початкове налаштування проєкту, дозволяючи розробникам зосередитися на написанні коду, а не на конфігурації середовища.

Для розробки клієнтської частини інтернет-магазину використовуватимуться сторонні бібліотеки, що спрощують процес розробки та підвищують функціональність додатку. Однією з таких важливих бібліотек є Axios (рис. 3.5).

```
PS E:\ТНТУ\Диплом\D> npm install axios
added 9 packages in 5s
1 package is looking for funding
run `npm fund` for details
```

Рисунок 3.5 – Інсталяція бібліотеки Axios для інтернет-магазину

Axios підтримує всі основні методи HTTP, такі як GET, POST, PUT, DELETE, що робить її універсальним інструментом для інтеграції frontend з backend. Однією з ключових переваг Axios є її здатність обробляти запити і відповіді в зручному форматі JSON, що значно полегшує роботу з даними [25]. Крім того, бібліотека надає можливості для налаштування заголовків запитів, що дозволяє легко додавати токени автентифікації або інші необхідні параметри.

Крім власноруч створених компонентів, у клієнтській частині також використовуються готові компоненти з бібліотеки “Material UI”. Ця бібліотека має кілька важливих переваг: широкий асортимент елементів верстки, значний функціонал кожного з них та легкість у модифікації поведінки будь-якого компонента відповідно до потреб.

В інтернет-магазині була реалізована функція діапазону цін, що дозволяє користувачам вибирати товари у межах від 10 до 50000 гривень. Ця функція представлена у вигляді повзунка, який можна налаштувати для вибору потрібного цінового діапазону (див. лістинг 3.1). Використовуючи цей повзунок, покупці можуть швидко знайти товари, що відповідають їхньому бюджету, що значно спрощує процес пошуку та робить його більш зручним.

### Лістинг 3.1 – Функція для фільтрації товарів за діапазоном цін

```
<div className="inputItem">
  <div className="top"><span>10</span><span>{maxPrice}</span>
</div>
  <input type="range" min={10} max={50000}
onChange={ (e) => setMaxPrice(e.target.value) } /></div>
```

Впровадження функції діапазону цін допомагає користувачам звузити вибір та знайти товари, які відповідають нижчій ціновій категорії (рис. 3.6).

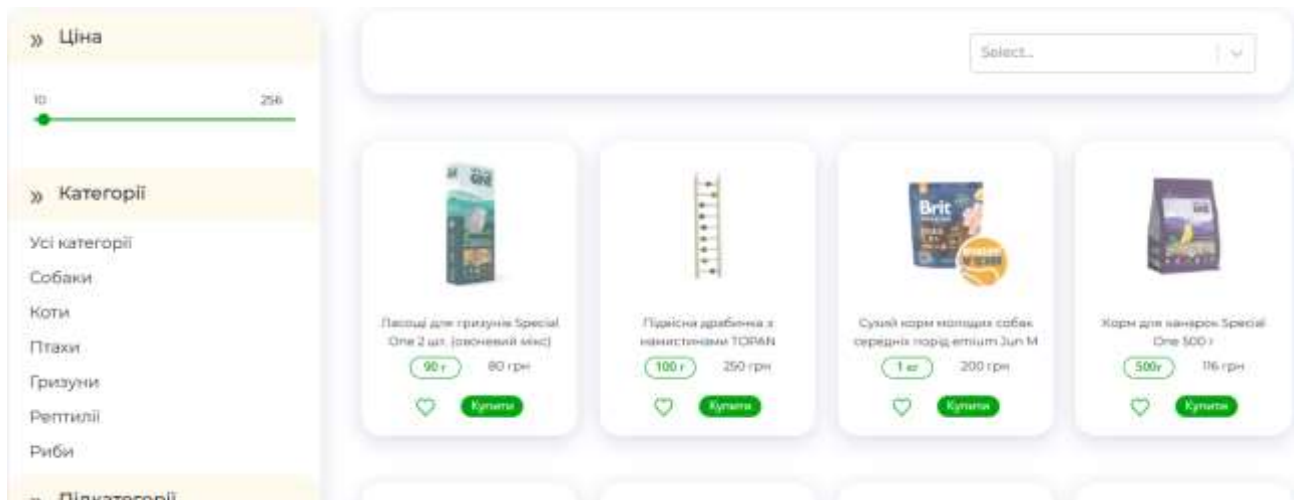


Рисунок 3.6 – Застосування функції фільтрації діапазону цін

Також розроблено функцію сортування товарів, що дозволяє користувачам упорядковувати товари за різними критеріями (див. додаток А). Для цього у системі використовується список фільтрів, серед яких можна обрати сортування за новизною, за зростанням ціни, за спаданням ціни або ж відсутність сортування (див. лістинг 3.2).

### Лістинг 3.2 – Фільтр для сортування товарів інтернет-магазину

```
const [selectedFilter, setSelectedFilter] = useState('');
const handleFilterChange = (selectedOption) => {
  setSelectedFilter(selectedOption.value);
};
const filterOptions = [
  { value: 'new', label: 'Нові' },
  { value: 'asc', label: 'За зростанням' },
  { value: 'desc', label: 'За спаданням' },
  { value: 'none', label: 'Без сортування' },
];
<div className="sort font-motserrat">
```

```

<p></p>
<div className="select">
<Select options={filterOptions} className='select font-
motserrat' onChange={handleFilterChange}
value={filterOptions.find((option) => option.value ===
selectedFilter)} />
<div class="select_arrow"></div>
</div></div>

```

Функція сортування реалізована через випадаючий список, де користувачі можуть обрати потрібний варіант. Зміна вибору активує відповідний фільтр і оновлює відображення товарів згідно з обраним критерієм (рис. 3.7). Це значно покращує зручність користування інтернет-магазином, дозволяючи швидко знаходити необхідні товари за бажаними характеристиками.

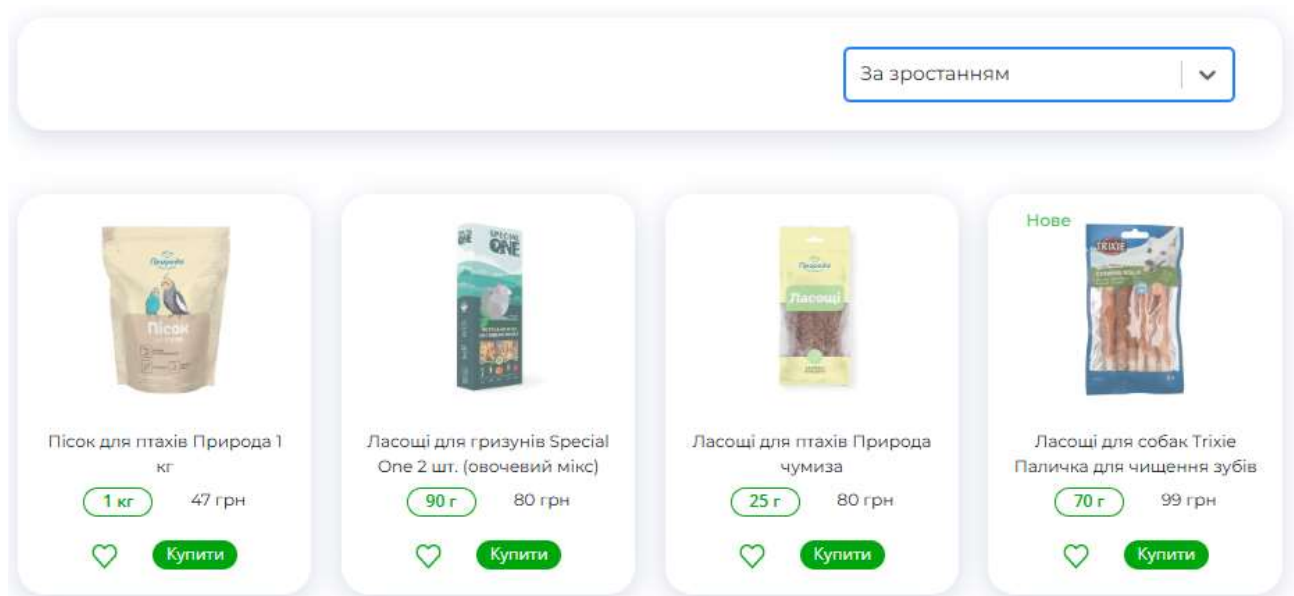


Рисунок 3.7 – Вибір сортування товарів в інтернет-магазині

Реалізація фільтрації за категоріями та підкатегоріями в інтернет-магазині забезпечує зручний спосіб вибору товарів (див. додаток А). Користувачі можуть обирати основні категорії, такі як «Усі категорії», «Собаки», «Коти», «Птахи», «Гризуни», «Рептилії» та «Риби». Кожна з цих категорій представлена у вигляді посилання, яке перенаправляє користувача до відповідної сторінки з товарами. Підкатегорії відображені у вигляді checkbox (рис. 3.8). Користувачі можуть

обирати одну або декілька підкатегорій, щоб додатково звужити свій вибір. При виборі підкатегорії відбувається виклик функції `handleChange` [26], яка оновлює стан вибору підкатегорій.

Функція фільтрації за категоріями і підкатегоріями інтегрована у зручний інтерфейс, що робить процес пошуку простішим і швидшим. Користувачам не потрібно переглядати всі товари на сайті, оскільки вони можуть легко відфільтрувати потрібні товари за кілька дій. Це особливо корисно для великих інтернет-магазинів з широким асортиментом продукції.

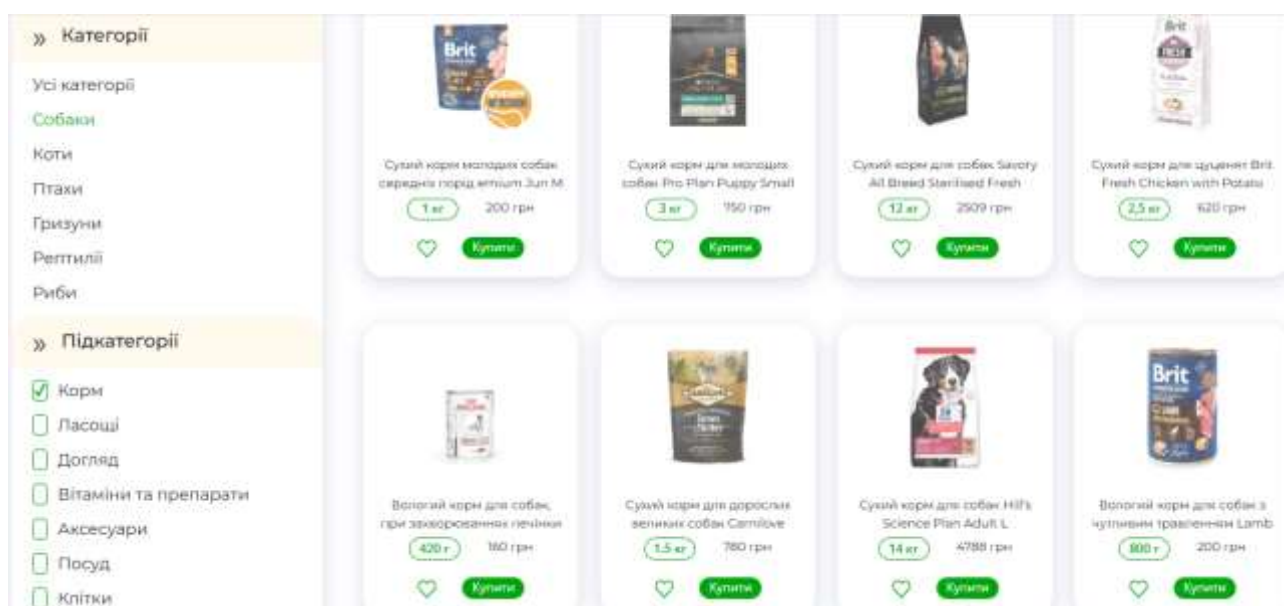


Рисунок 3.8 – Застосування фільтрації товарів за категорією «Собаки» та підкатегорією «Корм»

На сторінці з інформацією про товар зображена лаконічна та зручна для користувача панель. Верхня частина сторінки включає зображення товару, а поруч розміщена назва товару. Нижче наведена ціна – 80 грн за упаковку вагою 90 г. Користувач може регулювати кількість товару для покупки за допомогою кнопок «+» та «-». Іконка у вигляді серця дозволяє додати товар до списку вподобаних, що зручно для користувачів, які хочуть зберегти цей товар для майбутніх покупок.

Сторінка товару також включає опис продукту, його склад та характеристики (див. додаток К). Далі йде блок з інформацією про доставку: доступні варіанти доставки включають «Укрпошту» та «Нову Пошту», з безкоштовною доставкою при замовленні від 500 грн. Розділ «Повернення» надає інструкції щодо умов обміну та повернення товару, а також інформацію щодо захисту прав споживачів відповідно до законодавства України. У додатку Б наведено лістинг для сторінки товару інтернет-магазину Dolittle.

Коли користувач натискає кнопку «Додати в кошик», товар автоматично додається до кошика. У кошику відображається коротка інформація про товар, включаючи назву, кількість (1 шт) та ціну (80 грн) (див. додаток В). Іконка смітника дозволяє видалити товар з кошика, якщо користувач захоче змінити замовлення (рис. 3.9).

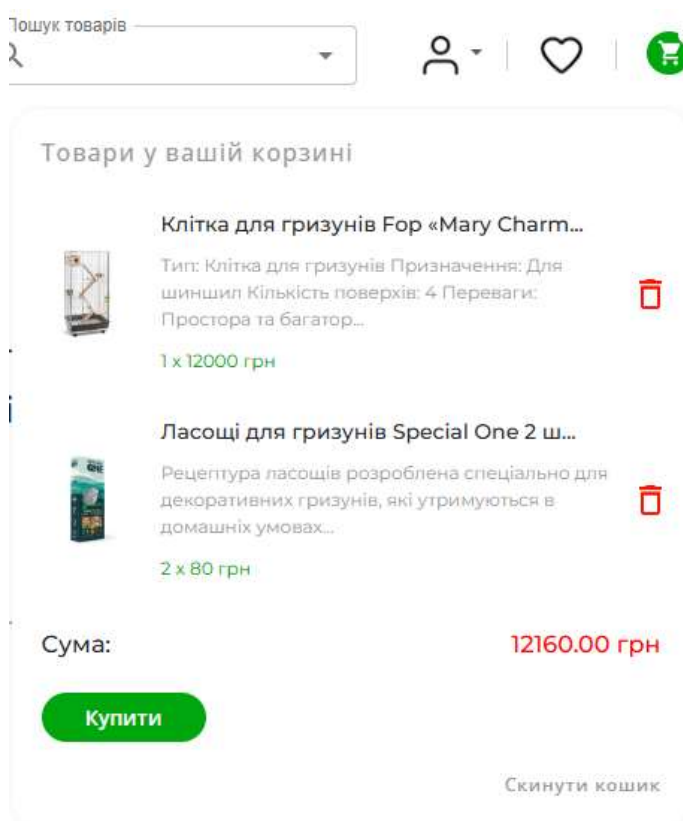


Рисунок 3.9 – Заповнення кошика інтернет-магазину

Функція вподобання товарів реалізована через контейнер (див. додаток Д). За допомогою функції «map» здійснюється ітерація через масив вподобаних



товарів «productslike», і для кожного з них створюється окрема картка (див. лістинг 3.3).

### Лістинг 3.3 – Функція вподобання товарів в інтернет-магазині

```
<div className="items">
  {productslike?.map(item => (
    <div className="cards font-motserrat" key={item.id}>
      <div className="image">
        <div className="size"><img src={`\${item.img}`} alt="" />
      </div></div>
      <h3>{item?.title}</h3>
      <div className="price">
        <div className="item"><p className="font-
opensans">{item.weight}</p></div><h4>{item?.price} грн</h4></div>
        <div className="item1">
           dispatch(removeLike(item.id))} />
          <div className="item2" ><p className="font-
opensans">Купити</p></div></div></div>)) }</div>
```

Таким чином, процес вподобання товарів забезпечує зручність порівняння та легкість в управлінні вподобаними товарами (рис. 3.10).



Рисунок 3.10 – Список вподобаних товарів в інтернет-магазині

Додаючи товари до списку вподобань, можна легко знайти їх пізніше, не витрачаючи час на пошук. Це економить час та покращує загальний досвід користування інтернет-магазином.

Процес реалізації пошуку товарів в інтерфейсі користувача був здійснений за допомогою бібліотек React та MUI (Material-UI) (див. додаток E). Цей підхід дозволив створити зручний і функціональний компонент пошуку, який інтегрується з API для отримання даних про товари (рис. 3.11).

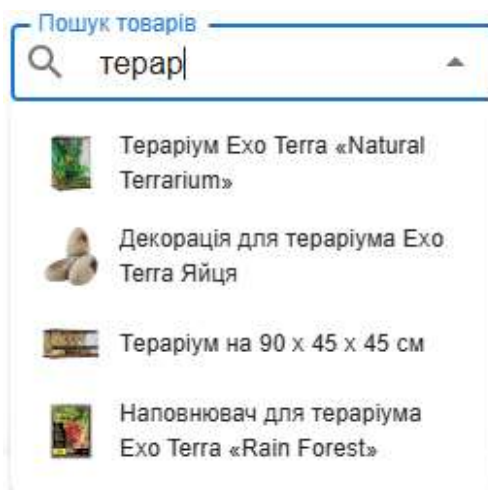


Рисунок 3.11 – Пошук товарів інтернет-магазину Dolittle

Для реалізації випадаючого списку для іконки користувача, який з'являється тільки після входу користувача в свій акаунт, слід виконати кілька кроків (рис. 3.12). Перш за все, необхідно налаштувати навігаційний бар з використанням компонентів Bootstrap для React.

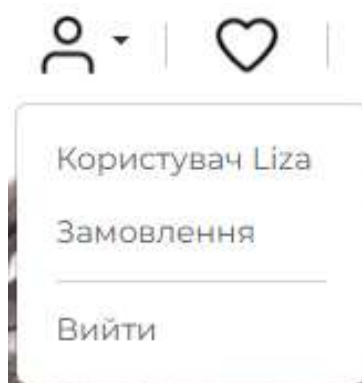


Рисунок 3.12 – Профіль користувача інтернет-магазину Dolittle

Реалізація профілю користувача включає налаштування компонентів навігаційної панелі, додавання логіки перевірки стану входу користувача та

динамічне відображення відповідних елементів меню для забезпечення доступу до профілю користувача та історії замовлень.

### 3.3 Наповнення інформацією про товари бази даних інтернет-магазину

Після успішного завершення етапів розробки backend і frontend для інтернет-магазину Dolittle, настає ключовий етап – наповнення даними. Розгортання є важливим процесом, який забезпечує інтернет-магазин необхідною інформацією для зручного користування покупцями та ефективного продажу товарів.

Для початку необхідно створити структуру каталогу товарів. Важливо визначити основні категорії, щоб покупцям легко знаходити потрібні товари (рис. 3.13).

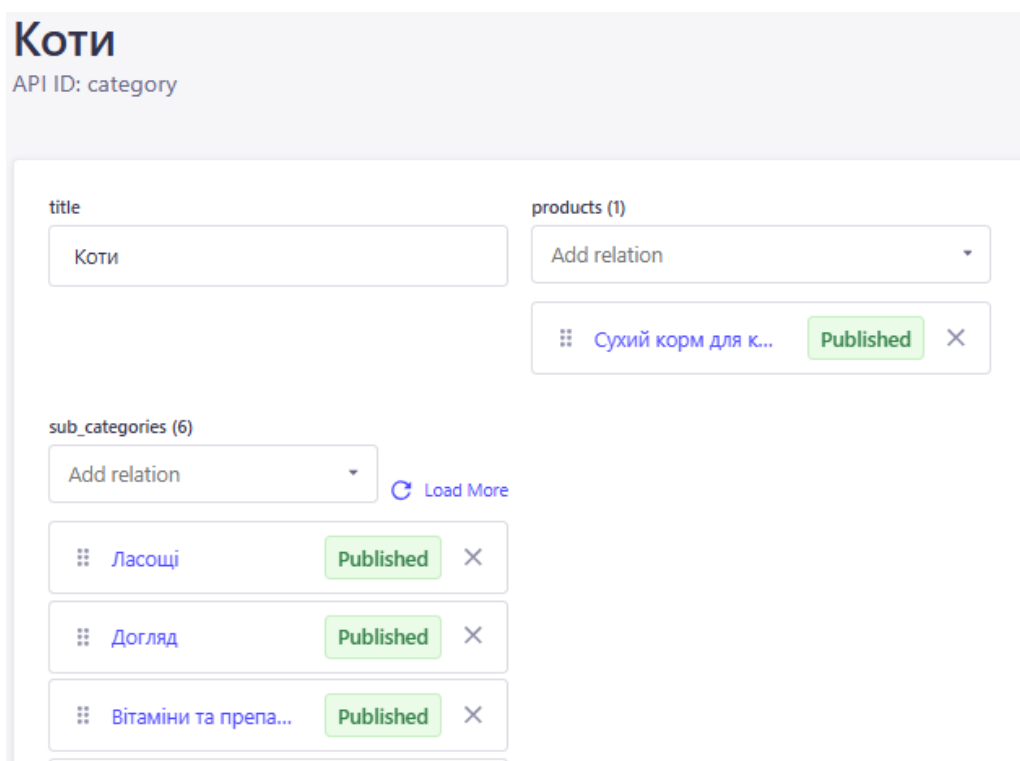


Рисунок 3.13 – Приклад наповнення категорії «Коти» інтернет-магазину

Під кожною основною категорією слід створити підкатегорії за видами товарів (рис. 3.14). Наприклад, в категорії «Собаки» можуть мати підкатегорії «Корм», «Вітаміни та препарати», «Догляд», «Ласощі», «Акcesуари».

**Корм**  
API ID: sub-category

title: Корм

products (1): Add relation

Сухий корм для к... Published X


Рисунок 3.14 – Приклад наповнення підкатегорії «Корм» інтернет-магазину


Після наповнення даними категорій та підкатегорій можна приступити до наповнення даними таблиці товарів, що є одним із найважливіших етапів для успішної роботи інтернет-магазину Dolittle (рис. 3.15).

**Сухий корм для котів Royal Canin Hair & Skin**  
API ID: product

title: Сухий корм для котів Royal Canin Hair & Ski

isNew: FALSE TRUE

img:   
sukhiy-korm-dlya-kotiv-royal-canin-hair-skin-16...

img2:   
73673030454626.webp

price:

weight:

Рисунок 3.15 – Заповнення таблиці товарів інтернет-магазину

Таблиця товару включає в себе багато інформації, яка допомагає уточнити деталі про товари та забезпечує покупцям повну інформацію для прийняття рішення щодо покупки. Вона містить такі елементи, як назва товару, його опис, характеристики, склад, ціна, наявність на складі та відгуки покупців. Такий детальний підхід дозволяє клієнтам проаналізувати повну інформацію для прийняття рішення щодо покупки. Загальний процес наповнення даними інтернет-магазину Dolittle є критично важливим для забезпечення зручності користувачів та ефективного продажу товарів. Створення основних категорій та підкатегорій сприяє зручному пошуку товарів, а наповнення таблиці товарів забезпечує покупцям необхідну інформацію для виваженого вибору. Цей процес є ключовим для успішного функціонування інтернет-магазину.

### 3.4 Розгортання програмної інфраструктури інтернет-магазину

Заключним етапом в розробці інтернет-магазину є його розгортання. Для розгортання серверної частини обрано два сервіси: `render.com` та `Cloudinary`. А для клієнтської частини – `InfinityFree` [27]. Служби `Render` та `Cloudinary` надають безкоштовні облікові записи та мають можливість вибору безкоштовного плану для розгортання. Спочатку створюється на `render.com` нова база даних PostgreSQL (рис. 3.16). На інформаційній панелі `render.com` потрібно натиснути «Бази даних» у меню ліворуч, а потім «Створити базу даних». Обрати тип БД, дати назву базі даних і натиснути «Створити базу даних» [28].

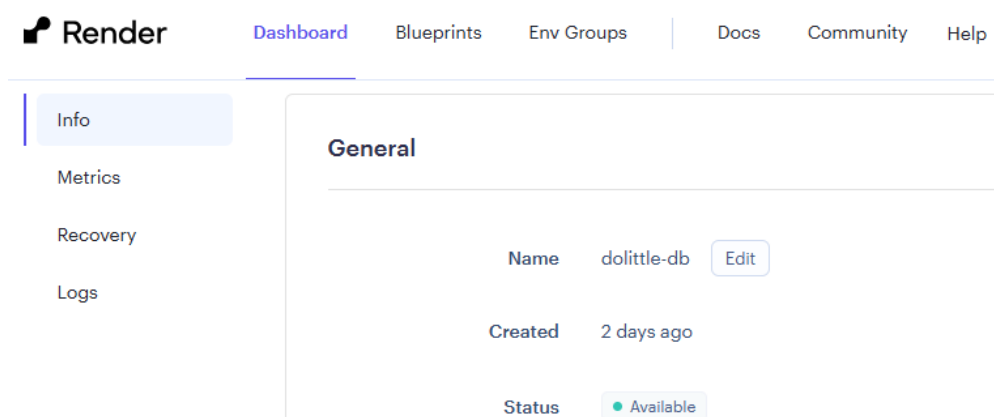


Рисунок 3.16 – БД PostgreSQL на `render.com`

Спочатку на Cloudinary створюється новий проєкт і встановлюється плагін Strapi Cloudinary [29]. Скопійовані ключ і API необхідно додати до змінних середовища інтернет-магазину на платформі Render. Після створення нового вебсервісу на Render і налаштування відповідних змінних середовища, Render надає доступ до програми через унікальну URL-адресу (рис. 3.17).

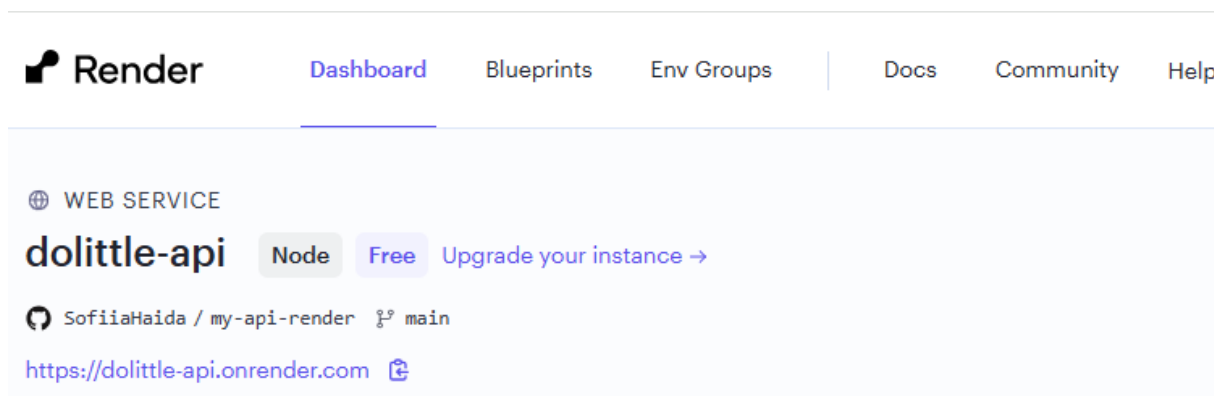


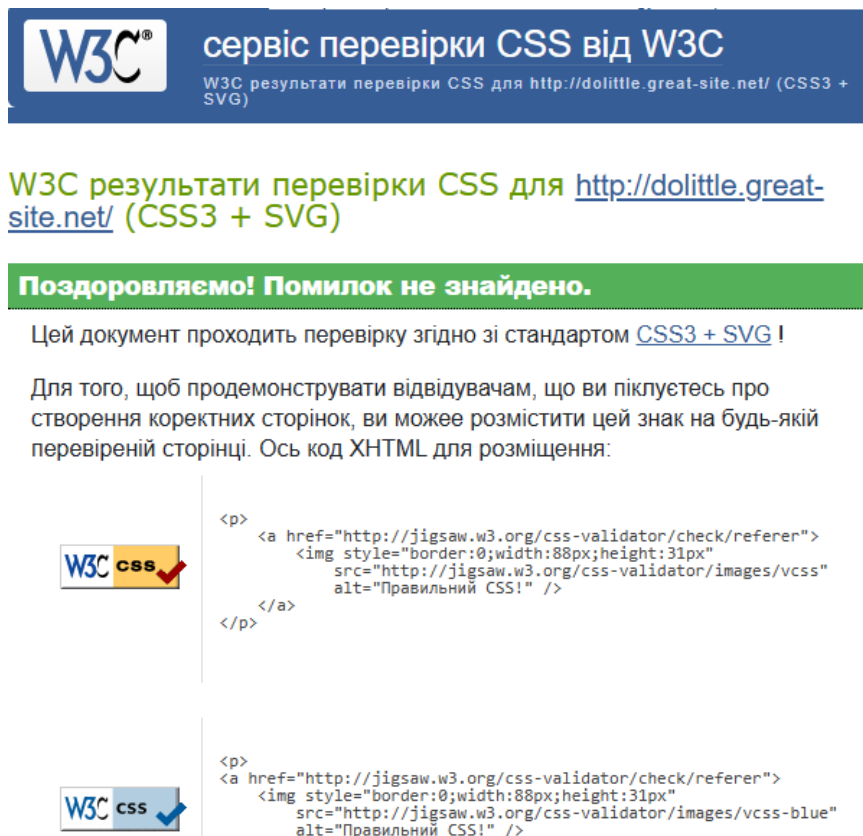
Рисунок 3.17 – Доступ до панелі адміністратора інтернет-магазину

Серверна частина була успішно розгорнута на Render з використанням Cloudinary для оптимізації медіа-контенту. Render забезпечив надійний сервер і доступ до панелі через унікальну URL-адресу, що полегшує управління вмістом інтернет-магазину. Це забезпечує швидке завантаження зображень і відео та високу продуктивність системи. Cloudinary, як SaaS-рішення, дозволяє інтегрувати керування медіа без необхідності самостійної розробки цього функціоналу. Для дослідження більшої інформації про моделі хмарних послуг взято наукову статтю, яка має назву «Огляд моделей хмарних послуг» [30].

### **3.5 Валідація розмітки інтерфейсу користувача та тестування інтернет-магазину**

Валідація CSS-коду є важливим процесом в розробці інтернет-магазину, оскільки вона гарантує, що стилі сайту відповідають сучасним стандартам та рекомендаціям. Процес валідації допомагає виявити помилки в коді, які можуть

призвести до некоректного відображення сайту на різних пристроях і браузерах (рис. 3.18). Це також сприяє кращій продуктивності сайту і забезпечує його доступність для всіх користувачів.



**W3C** сервіс перевірки CSS від W3C  
W3C результати перевірки CSS для <http://dolittle.great-site.net/> (CSS3 + SVG)

**Поздоровляємо! Помилки не знайдено.**

Цей документ проходить перевірку згідно зі стандартом [CSS3 + SVG](#) !

Для того, щоб продемонструвати відвідувачам, що ви піклуєтесь про створення коректних сторінок, ви можете розмістити цей знак на будь-якій перевірених сторінці. Ось код XHTML для розміщення:

```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```

```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```

Рисунок 3.18 – Успішна валідація CSS-коду інтернет-магазину

Перевірка CSS-коду через W3C Validator була успішною. Це означає, що всі стилі, які використовуються в інтернет-магазині, відповідають стандартам і не містять критичних помилок [31]. Така відповідність забезпечує коректне відображення сторінок на різних пристроях і браузерах, підвищуючи якість користувацького досвіду. Окрім того, це свідчить про професійний підхід до розробки та підтримки сайту, що позитивно впливає на репутацію інтернет-магазину.

Тестування інтернет-магазину є критичним етапом у процесі розробки, оскільки воно забезпечує якість, стабільність і функціональність сайту. Регулярне та ретельне тестування допомагає виявити та виправити помилки, покращити користувацький досвід та забезпечити відповідність сайту сучасним

стандартам і вимогам. Без належного тестування, інтернет-магазин може зіткнутися з проблемами продуктивності, безпеки та доступності, що може негативно вплинути на довіру користувачів і призвести до втрати продажів.

Для тестування інтернет-магазину був використаний Lighthouse – потужний інструмент від Google. Lighthouse проводить всебічний аналіз сайту, оцінюючи його продуктивність, доступність, прогресивність та SEO [32]. Інструмент генерує звіти з детальними рекомендаціями щодо покращення кожного з аспектів, допомагаючи оптимізувати сайт для кращої роботи (рис. 3.19).

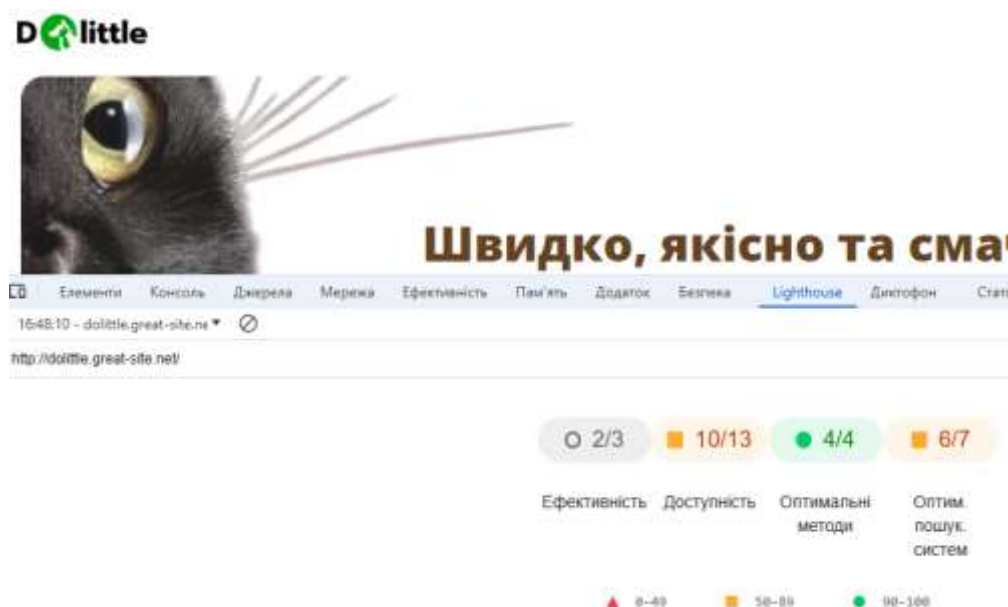


Рисунок 3.19 – Тестування інтернет-магазину за допомогою Lighthouse

Результати тестування інтернет-магазину за допомогою Lighthouse відображають кілька ключових аспектів роботи інтернет-магазину.

Показник ефективності на рівні 2 з 3 вказує на те, що сайт працює досить добре, але є місця для покращення. Основні проблеми можуть стосуватися швидкості завантаження сторінок та загальної продуктивності. Це може бути наслідком великого обсягу ресурсів, таких як зображення або скрипти, які потребують оптимізації для швидшого завантаження.



Рівень доступності на рівні 10 з 13 показує, що сайт в цілому доступний для користувачів, включаючи тих, хто має обмежені можливості. Однак, є кілька аспектів, які потребують покращення, щоб сайт відповідав всім стандартам доступності. Це можуть бути питання кольорового контрасту, альтернативного тексту для зображень або навігаційних елементів.

Показник оптимальних методів на рівні 4 з 4 свідчить про те, що сайт дотримується всіх найкращих практик веброзробки. Це включає безпечне використання ресурсів, коректну роботу скриптів та загальну оптимізацію медіа-контенту. Показник SEO на рівні 6 з 7 говорить про те, що сайт добре оптимізований для пошукових систем, але ще є простір для вдосконалення.

Тестування дозволяє виявити ключові проблеми, які можуть впливати на швидкість завантаження сторінок, доступність контенту для користувачів з обмеженими можливостями, а також на пошукову оптимізацію. Для більш детальної інформації про процес тестування та розгортання можна ознайомитися з науковою статтею [33]. У ній йдеться про важливість підтримки якості програмного коду та програмного продукту в цілому для успішності технологічного процесу розробки ПЗ. Ця задача вирішується шляхом застосування методології CI/CD, яка складається з двох основних етапів.

### **3.6 Висновок до третього розділу**

У 3 розділі розглянуто процес створення інтернет-магазину Dolittle. Розроблено серверну частину на базі CMS Strapi для забезпечення зручного управління контентом та клієнтську частину з використанням фреймворка React для створення інтерактивного інтерфейсу. Крім того, проведено наповнення даними магазину, виконано валідацію та тестування ПЗ. Також детально описано процес розгортання інтернет-магазину з використанням хмарних сервісів та хостингів. У цілому, реалізація цих кроків сприяла успішному створенню інтернет-магазину, підвищуючи його функціональність та ефективність, забезпечуючи зручність як для адміністратора, так і для кінцевих користувачів.

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Долікарська допомога при ураженні електричним струмом

Ураження електричним струмом може мати серйозні наслідки для здоров'я, включаючи зупинку серця, опіки, пошкодження внутрішніх органів та м'язів. Негайне надання першої допомоги може врятувати життя постраждалого. Долікарська допомога включає кілька важливих етапів, які слід виконувати в чіткій послідовності.

Перше, що потрібно зробити – це зупинити подачу електричного струму. Якщо є можливість, вимкнути живлення за допомогою вимикача, розетки або вимикача на щитку. Якщо безпечне вимкнення електрики неможливе, потрібно використати неелектропровідні предмети, такі як дерев'яна палиця або пластиковий предмет, щоб відсунути постраждалого від джерела струму. Важливо не торкатися постраждалого голими руками, поки він знаходиться під впливом струму, щоб не постраждати самому.

Одразу після зупинення подачі струму необхідно викликати швидку допомогу. Якщо постраждалий здається в свідомості та відчувається відносно добре, ураження електричним струмом може мати відстрочені наслідки, такі як аритмії або інші проблеми з серцем, які можуть проявитися пізніше. Під час виклику медиків важливо чітко пояснити ситуацію і вказати, що сталося ураження електричним струмом.

Після виклику швидкої допомоги необхідно оцінити стан постраждалого. Перевірити чи є у нього свідомість, чи дихає він і чи наявний пульс. Якщо постраждалий у свідомості, необхідно заспокоїти його і попросити не рухатися, щоб уникнути можливих подальших травм. Якщо постраждалий не дихає або не має пульсу, слід негайно розпочати серцево-легеневу реанімацію (СЛР). Вона є критично важливою при зупинці серця або дихання [34].

Для виконання СЛР покладіть постраждалого на тверду поверхню, розпочніть компресії грудної клітини зі швидкістю 100-120 компресій за

хвилину, глибиною приблизно 5-6 см, у співвідношенні 30 компресій на 2 вдихи, якщо ви маєте відповідні навички. Якщо людина не впевнена у своїх навичках, продовжуйте робити тільки компресії грудної клітини, поки не приїде швидка допомога (рис. 4.1).

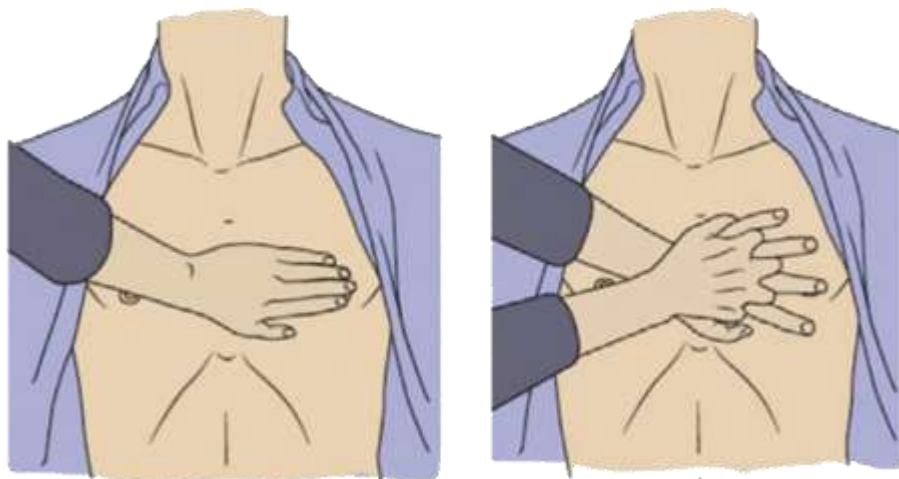


Рисунок 4.1 – Виконання компресій грудної клітини

Якщо у постраждалого є видимі опіки, необхідно їх охолодити. Використати прохолодну (але не холодну) воду для охолодження уражених ділянок протягом 10-15 хвилин. Не використовувати лід, оскільки це може пошкодити тканини ще більше. Після охолодження опіків накрити їх стерильною пов'язкою або чистою тканиною, щоб запобігти інфекції.

До приїзду швидкої допомоги потрібно зберегти постраждалого в стані спокою [35]. Покласти його в горизонтальне положення, підняти ноги для покращення кровообігу, якщо він не має травм, що унеможливають це. Накрити постраждалого ковдрою або одягом, щоб запобігти переохолодженню, особливо якщо він знаходиться в шоковому стані.

Постійно потрібно контролювати стан постраждалого до прибуття медиків. Перевіряти дихання, пульс і свідомість, бути готовим знову розпочати СЛР, якщо це буде необхідно. Своєчасна та правильна перша допомога при ураженні електричним струмом може суттєво підвищити шанси на виживання та мінімізувати наслідки травми для постраждалого.

## 4.2 Психофізіологічне розвантаження для працівників.

Психофізіологічне розвантаження є важливим аспектом підтримання здоров'я та продуктивності працівників. Воно включає різні методи і техніки, спрямовані на зниження стресу, зменшення втоми та покращення загального самопочуття. Це питання набуває особливої актуальності в умовах сучасного робочого середовища, де працівники часто стикаються з високими навантаженнями, тиском та емоційним вигоранням.

Одним із ключових методів психофізіологічного розвантаження є релаксація [36]. Релаксаційні техніки, такі як медитація, глибоке дихання, йога та прогресивна м'язова релаксація, допомагають зменшити напругу та стрес, покращують концентрацію та сприяють відновленню енергії. Регулярна практика цих технік може значно покращити психологічний стан працівників, зробити їх стійкими до стресових ситуацій та підвищити рівень задоволення роботою.

Фізична активність також відіграє важливу роль у психофізіологічному розвантаженні. Регулярні фізичні вправи, такі як кардіотренування, силові тренування, а також простіші форми активності, наприклад, ходьба або розтяжка, допомагають знизити рівень стресу, покращують настрій та загальну фізичну форму [37]. Вони стимулюють вироблення ендорфінів – «гормонів щастя», які природним чином знижують рівень стресу і тривоги.

Аутогенне тренування – це метод психофізіологічного розвантаження, який базується на самогіпнозі. Цей метод був розроблений німецьким психіатром Йоганном Генріхом Шульцем у кінці 19-го століття. Основна ідея полягає в тому, щоб за допомогою спеціальних вправ і концентрації уваги змінити стан свідомості, викликати фізіологічні реакції та досягти глибокого фізичного та психологічного розслаблення. Цей тип тренувань поділяють на три основні етапи, які відображають різні стадії відновлюваного процесу.

Процес аутогенного тренування розділяється на три етапи. Першим етапом тренувань є абстрагування. Воно сприяє відволіканню та розслабленню

працівників від робочого середовища. Працівники слухають спокійну музику, можливо звуки природи. Займають зручну позу, що допомагає адаптуватися до наступних етапів.

Наступним етапом тренувань є затишок. Він сприяє регенерації та відновленню енергії. Під час нього на екрані демонструються заспокійливі фотографії та відео, а у навушниках звучить повільна музика, супроводжувана формулою аутогенного тренування. Ця формула складається з трьох коротких речень [38]:

- «Все моє тіло розслаблене, я відчуваю спокій».
- «Моє дихання рівномірне і спокійне».
- «Моє тіло важке і тепле, я повністю розслаблений, мій лоб прохолодний, а голова легка».

Під час сеансу горить зелене світло, але під кінець гасне на кілька хвилин. Зелений колір освітлення може служити як світло, яке допомагає людям залишатися у свідомому стані під час тренування, сприяючи їх концентрації та увазі.

Останній етап включає активізацію. На початку цього етапу вимикається світло, а потім з'являється яскрава пляма червоного кольору, супроводжуючись веселою музикою та формулами аутогенного тренування, що стимулюють відчуття бадьорості та енергії. Важливою частиною цього етапу є можливість вимовляти позитивні фрази для підсилення позитивного настрою та самопочуття. Ці фрази можуть включати:

- «Відчуваю бадьорість та енергію, маю чудовий настрій».
- «Я готовий до дій».

Аутогенне тренування сприяє підвищенню рівня концентрації та уваги. Це дозволяє працівникам краще зосередитися на завданнях та зберігати чіткість думок, що в свою чергу підвищує їх продуктивність та ефективність роботи. Регулярна практика цього методу може поліпшити якість сну, знизити кров'яний тиск та полегшити відчуття втоми. В результаті, працівники відчують більшу

енергію та витривалість протягом дня, що сприяє їх загальному самопочуттю та задоволенню від роботи.

Для ефективного психофізіологічного розвантаження необхідно мотивувати персонал, використовуючи нематеріальні засоби. Найуспішніші компанії впроваджують інноваційні підходи, щоб забезпечити своїм працівникам комфортне та натхненне робоче середовище [39].

Наприклад, Google створює для своїх співробітників простори, які сприяють творчості та зняттю стресу. Офіси компанії обладнані яскравими інтер'єрами, зонами відпочинку, кафе з безкоштовною їжею та тренажерними залами. Google також пропонує різноманітні навчальні програми та курси для розвитку професійних навичок. Ці заходи не тільки підвищують рівень задоволеності працівників, але й стимулюють їх до нових звершень та інновацій.

Apple також приділяє велику увагу нематеріальному мотивуванню. Компанія створює середовище для балансування роботи та особистого життя. Працівники мають доступ до зручних робочих місць, де вони можуть відпочити або провести час у спокійній атмосфері. Apple забезпечує своїх співробітників безкоштовними обідами, доступом до фітнес-центрів та можливостями для навчання і професійного зростання. Такий підхід сприяє створенню гармонійного робочого середовища, де працівники відчують себе цінними та мотивованими.

Facebook активно застосовує нематеріальні методи мотивації. В офісах компанії є зони для відпочинку, де працівники можуть зняти стрес і відновити енергію. Компанія надає можливість займатися спортом, відвідувати тренінги та майстер-класи, а також бере участь у благодійних проєктах. Це допомагає працівникам відчувати свою причетність до важливих справ, що позитивно впливає на їх моральний дух і мотивацію.

Таким чином, нематеріальне мотивування, реалізоване для створення комфортних умов праці, а також можливості для професійного зростання, має значний позитивний вплив на психофізіологічний стан працівників, їх продуктивність та загальне задоволення роботою.

### 4.3 Висновок до четвертого розділу

У цьому розділі проаналізовано питання долікарської допомоги при ураженні електричним струмом. Інструкції з надання допомоги, які надані у розділі, включають в себе кроки зі зупинення подачі струму, виклику медичної допомоги та надання допомоги постраждалому до прибуття медиків. Ця інформація може допомогти врятувати життя та запобігти серйозним ушкодженням постраждалого, забезпечуючи своєчасне та правильне надання першої допомоги до прибуття професійних медиків.

Розглянуто психофізіологічне розвантаження працівників, яке є критичним елементом збереження здоров'я та підвищення продуктивності. У робочому середовищі, яке насичене стресом та навантаженнями, методи психофізіологічного розвантаження, такі як релаксація, фізична активність та аутогенне тренування, набувають особливої важливості.

Одним із ключових методів психофізіологічного розвантаження є релаксація. Вона включає в себе різноманітні техніки, такі як медитація, глибоке дихання та йога, які сприяють зниженню напруги та стресу, а також покращенню енергетики. Крім того, фізична активність, така як кардіотренування та регулярні вправи, допомагає зберегти здоров'я та знизити рівень стресу.

Аутогенне тренування, яке базується на самогіпнозі, також є ефективним методом психофізіологічного розвантаження. Цей метод розділений на три етапи: абстрагування, затишок та активізацію, дозволяє досягти глибокого фізичного та психологічного розслаблення, підвищуючи загальний рівень концентрації та уваги працівників.

Додатково до цих методів, мотивація персоналу є важливим елементом підтримки психофізіологічного благополуччя. Компанії, такі як Google, Apple та Facebook вирішують це завдання, створюючи комфортне та творче робоче середовище, а також надаючи працівникам доступ до різноманітних можливостей для професійного розвитку.

## ВИСНОВКИ

Результатом виконання кваліфікаційної роботи є проєкт інтернет-магазин Dolittle, розроблений згідно із сучасними патернами проєктування та вимогами до розробки. Інтернет-магазин є ефективним інструментом для продажу товарів, який відповідає потребам сучасних користувачів та забезпечує зручний і швидкий доступ до продукції. Результати дослідження теми кваліфікаційної роботи поділено на три основні розділи: аналіз вимог та постановка задачі, проєктування інтернет-магазину, реалізація інтернет-магазину.

У першому розділі проведено аналіз предметної області, сформовано вимоги інтернет-магазину, побудовано та досліджено діаграму варіантів використання, обрано середовище розробки, проаналізовано використовуваний стек технологій та розглянуто структуру інтернет-магазину. Цей розділ закладає фундамент для подальших етапів розробки, визначаючи основні параметри та технології, які використовуватимуться.

У другому розділі спроектовано архітектуру інтернет-магазину, описано функціональні моделі, змодельовано схеми інтернет-магазину, відображено структуру бази даних, досліджено основні залежності та файлову структуру інтернет-магазину. Також обґрунтовано вибір кольорової схеми та шаблонів для інтерфейсу користувача. Цей розділ детально описує архітектурні рішення та візуальні аспекти, які забезпечують зручність використання та естетичну привабливість інтернет-магазину.

У третьому розділі представлено реалізацію інтернет-магазину Dolittle, зокрема розробку серверної частини на базі CMS Strapi та клієнтської частини на базі фреймворка React. Проведено наповнення даними бази даних інтернет-магазину, описано процес розгортання інтернет-магазину, проведено валідацію та тестування. Цей розділ демонструє практичну реалізацію інтернет-магазину Dolittle, включаючи технічні аспекти та перевірку працездатності системи.

У розділі «Безпека життєдіяльності, основи охорони праці» досліджено основний алгоритм долікарської допомоги при ураженні електричним струмом



та надано рекомендації щодо надання допомоги постраждалому відповідно до його стану. Також висвітлено проблему психофізіологічного розвантаження працівників та описано методи нематеріальної мотивації працівників у глобальних компаніях, таких як Google, Apple, Facebook. Цей розділ акцентує увагу на важливості здоров'я та добробуту працівників, пропонуючи ефективні рішення для зниження стресу та підвищення мотивації.

## ПЕРЕЛІК ДЖЕРЕЛ

1. QArea – eCommerce Website Development: Process, Tips, and More. [Електронний ресурс] – Режим доступу до ресурсу: <https://qarea.com/blog/the-ultimate-guide-to-ecommerce-web-development>
2. WebDev&SEO – Переваги інтернет-магазину над звичайною торговою точкою. [Електронний ресурс] – Режим доступу до ресурсу: <https://webdevandseo.com/online-store-benefits/>.
3. Purchase Commerce – 10 Essential Ecommerce Website Requirements and Best-in-class Features. [Електронний ресурс] – Режим доступу до ресурсу: <https://purchasecommerce.com/blog/10-essential-ecommerce-requirement-best-in-class-feature>.
4. IlionDigital – Правильна структура інтернет-магазину. [Електронний ресурс] – Режим доступу до ресурсу: <https://ilion.digital/ua/pravilna-struktura-internet-magazinu/>.
5. LucidChard – UML Use Case Diagram Tutorial. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lucidchart.com/pages/uml-use-case-diagram>.
6. Visual Studio Code – Documentation. [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/docs>.
7. React – Початок роботи. [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.legacy.reactjs.org/docs/getting-started.html>.
8. DigitalOcean – SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>.
9. Strapi – Welcome to the Strapi Developer Docs!. [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.strapi.io/dev-docs/intro>.
10. Е. Фрімен. Патерни проєктування / Е. Фрімен, Е. Робсон, Б. Бейтс, К. Сьєрра: пер. А. Якубовська. – Харків: Фабула, 2020. – 672 с.

11. AWS – What is PostgreSQL? [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.strapi.io/dev-docs/intro>.

12. IBM – The three tiers in detail. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/topics/three-tier-architecture>.

13. Готович, В. А., & Ралік, І. Р. (2022). Програмне забезпечення на основі клієнт-серверної архітектури для обліку реалізації товарів в торгівлі. Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 126-126.

14. EdrawMax – Everything about Functional Block Diagrams. [Електронний ресурс] – Режим доступу до ресурсу: <https://edrawmax.wondershare.com/diagram-tips/function-block-diagram.html>.

15. VisualParadigm – All You Need to Know about State Diagrams. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/about-state-diagrams/>.

16. GeeksforGeeks – Sequence Diagrams | Unified Modeling Language (UML). [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>.

17. IBM – The UML 2 class diagram. [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.ibm.com/articles/the-class-diagram/>.

18. Lucidchart – What is an Entity Relationship Diagram (ERD)? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lucidchart.com/pages/er-diagrams>.

19. GeeksforGeeks – Axios in React: A Guide for Beginners. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/axios-in-react-a-guide-for-beginners/>.

20. Strapi – Users & Permissions plugin. [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.strapi.io/dev-docs/plugins/users-permissions>.

21. Studio1Design – How Important is Color in Website Design? What do colors mean? [Електронний ресурс] – Режим доступу: <https://studio1design.com/how-important-is-color-in-website-design/>.

22. Internetdevels – Build an emotional web design to give your website much more. What is emotional web design? [Електронний ресурс] – Режим доступу: <https://internetdevels.com/blog/emotional-web-design>.

23. Publitas – The Benefits of Using Online Product Catalogs. [Електронний ресурс] – Режим доступу: <https://www.publitas.com/blog/the-benefits-of-using-online-product-catalogs-and-whos-getting-it-right/>.

24. Restack – Strapi develop vs start comparison. [Електронний ресурс] – Режим доступу: <https://www.restack.io/docs/strapi-knowledge-strapi-develop-vs-start>.

25. DigitalOcean – How To Use Axios with React. [Електронний ресурс] – Режим доступу: <https://www.restack.io/docs/strapi-knowledge-strapi-develop-vs-start>.

26. Tutorialspoint – How to use the handleChange() function in react? [Електронний ресурс] – Режим доступу: <https://www.tutorialspoint.com/how-to-use-the-handlechange-function-in-react-component>.

27. InfinityFree – Free Website Hosting. [Електронний ресурс] – Режим доступу: <https://www.infinityfree.com/>.

28. Render – Cloud Application Hosting for Developers. [Електронний ресурс] – Режим доступу: <https://render.com/>.

29. Cloudinary – Image and Video API Platform. Visualize Engaging Experiences. [Електронний ресурс] – Режим доступу: <https://cloudinary.com/>.

30. Шевченко, Н. А., Валігула, М. В., Маєвський, Т. О., & Шимчук, Г. В. (2020). Огляд моделей хмарних послуг. Матеріали міжнародної наукової конференції „Іван Пулюй: життя в ім’я науки та України “(до 175-ліття від дня народження), 109-110.

31. W3C – Перевірка каскадних таблиць стилів (CSS) та документів (X)HTML з таблицями стилів. [Електронний ресурс] – Режим доступу: <https://jigsaw.w3.org/css-validator/>.

32. Chrome for Developers. Lighthouse Overview. [Електронний ресурс] – Режим доступу: <https://developer.chrome.com/docs/lighthouse/overview>.

33. Готович, В. А., & Мачужак, А. В. (2022). Застосування методології CI/CD для автоматизації процесів тестування та розгортання програмного забезпечення. Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 131-132.

34. Бережи себе – Серцево-легенева реанімація: навички, які можуть врятувати людину. [Електронний ресурс] – Режим доступу: <https://berezhysebe.com/navychky-sertsevo-lehenevoi-reanimatsii/>.

35. Gosta - Правила релаксації: 5 технік швидкої та дієвої релаксації. [Електронний ресурс] – Режим доступу: <https://gosta.media/psyholohiya/pravyla-relaksatsii-5-tehnik-shvydkoi-ta-diievoi-relaksatsii/>.

36. Охорона праці та пожежна безпека – Психофізіологічне перевантаження на роботі – як чинник виникнення травматизму. [Електронний ресурс] – Режим доступу: <https://oppb.com.ua/news/psyhofiziologichne-perevantazhennya-na-roboti-yak-chynnyk-vynyknennya-travmatyzmu>.

37. Gov.ua – Перша допомога при ураженні електричним струмом. [Електронний ресурс] – Режим доступу: <https://bozhedarivskaselrada.gov.ua/news/1576497483/>.

38. Курдибаха О. М. Вплив аутогенного тренування на психічні процеси. [Електронний ресурс] – Режим доступу до ресурсу: <https://rehab.kyiv.ua/vpliv-autogenного-trenuvannya-na-psihichni-protsesi/>.

39. Школа бізнесу. Нова Пошта – Нематеріальна мотивація: перелік лієвих способів підвищити залученість персоналу. [Електронний ресурс] – Режим доступу: <https://online.novaposhta.education/blog/nematerialna-motivatsiya-perelik-diyevih-sposobiv-pidvischiti-zaluchenist-personalu>.

# ДОДАТКИ

## Лістинг розмітки та скриптів для сторінки каталогу інтернет-магазину

## Dolittle

```

const Products = () => {
  const catId = parseInt(useParams().id)
  const [maxPrice, setMaxPrice] = useState(50000)
  const [selectedSubCategories, setSelectedSubCategories] =
  useState([])
  const {data, loading, error} = useFetch(`/sub-
categories?populate=*`)
  const handleChange = (e) =>{
    const value = e.target.value;
    const isChecked = e.target.checked;
    setSelectedSubCategories(
      isChecked
        ? [...selectedSubCategories, value]
        : selectedSubCategories.filter(item=>item !==
value));};
  const [selectedFilter, setSelectedFilter] = useState('');
  const handleFilterChange = (selectedOption) => {
    setSelectedFilter(selectedOption.value);};
  const filterOptions = [
    { value: 'new', label: 'Нові' },
    { value: 'asc', label: 'За зростанням' },
    { value: 'desc', label: 'За спаданням' },
    { value: 'none', label: 'Без сортування' },,];
  return (
    <div className='products'>
      <div className='menu font-motserratt'>
        <p><Link to="/"
className='link'>Головна</Link></p>
        <span>&gt;</span>
        <p style={{color: '#9C9C9C'}}>Каталог</p>
      </div>
      <h2 className='font-opensans'>КАТАЛОГ</h2>
      <div className="category">
        <div className="left">
          <div className="filter font-motserratt">
            <div className="filter1">
              <p>&raquo;</p>
              <h5>Ціна</h5>
            </div>
            <div className="inputItem">
              <div className="top">
                <span>10</span>
                <span>{maxPrice}</span>
              </div>
              <input type="range" min={10}
max={50000} onChange={ (e) =>setMaxPrice(e.target.value) }/>
            </div>
            <div className="filter1">

```

```

                <p>&raquo;</p>
                <h5>Категорії</h5>
            </div>
            <Link to="/products/0" className='link
decorate'><p>Усі категорії</p></Link>
            <Link to="/products/3" className='link
decorate'><p>Собаки</p></Link>
            <Link to="/products/2" className='link
decorate'><p>Коти</p></Link>
            <Link to="/products/1" className='link
decorate'><p>Птахи</p></Link>
            <Link to="/products/4" className='link
decorate'><p>Гризуні</p></Link>
            <Link to="/products/5" className='link
decorate'><p>Рептилії</p></Link>
            <Link to="/products/6" className='link
decorate'><p>Риби</p></Link>
            <div className="filter1">
                <p>&raquo;</p>
                <h5>Підкатегорії</h5>
            </div>
            {data?.map(item=>(
                <div
                    className='subcategories'>
                        <input
                            type="checkbox"
                            id={item?.id} value={item?.id} onChange={handleChange}/>
                        <label
                            htmlFor={item?.id}>{item?.attributes?.title}</label>
                    </div>
                )
            )
            </div>
            <div className="right">
                <div className="sort font-motserrat">
                    <p></p>
                    <div className="select">
                        <Select
                            options={filterOptions}
                            className='select font-motserrat'
                            onChange={handleFilterChange}
                            value={filterOptions.find((option) => option.value ===
                                selectedFilter)} />
                    <div class="select_arrow"></div>
                </div>
                <List
                    catId={catId}
                    maxPrice={maxPrice}
                    subCategories={selectedSubCategories}
                    selectedFilter={selectedFilter}/>
            </div>
        </div>
    )
}

export default Products

```



## Лістинг розмітки та скриптів для сторінки товару інтернет-магазину

## Dolittle

```

const Product = () => {
  const id = useParams().id;
  const [selectedImg, setSelectedImg] = useState("img");
  const [quantity, setQuantity] = useState(1);

  const { data, loading, error } =
useFetch(`/products/${id}?populate=*`);
  const dispatch = useDispatch();

  return (
    <div className="product font-motserrat">
      {loading ? (
        "loading"
      ) : (
        <>
          <div className='menu font-motserrat'>
            <p><Link to="/" className='link'>Головна</Link></p>
            <span>&gt;</span>
            <p><Link
              to="/products/0"
              className='link'>Каталог</Link></p>
            <span>&gt;</span>
            <p
              style={{color:
'#9C9C9C'}}>{data?.attributes.title}</p>
          </div>
          <div className="block">
            <div className="left">
              <div className='pictures'>
                <div className="images">
                  <div className="container">
                    <div
                      className="size"><img
src={`_${data?.attributes?.img?.data?.attributes?.url}` } alt=""
onClick={e=>setSelectedImg("img")} /></div>
                  </div>
                  <div className="container">
                    <div
                      className="size"><img
src={`_${data?.attributes?.img2?.data?.attributes?.url}` } alt=""
onClick={e=>setSelectedImg("img2")} /></div>
                  </div>
                </div>
                <div className="mainImg">
                  <img
src={`_${data?.attributes[selectedImg]?.data?.attributes?.url}` }
alt="" />
                </div>
              </div>
            </div>
            <div className='section'>

```

```

        <h3>Характеристики</h3>
        <div className="characteristic">
            <div className="cont1">
                <p>Бренд:</p>
                <p>Призначення:</p>
                <p>Вага:</p>
                <p>Виробник:</p>
            </div>
            <div className="cont1">
                <p style={{color:
'#9C9C9C'}}>{data?.attributes.brend}</p>
                <p>{data?.attributes.purpose}</p>
                <p>{data?.attributes.weight}</p>
            </div>
            <p>{data?.attributes.producer}</p>
        </div>
    </div>
    <div className='section'>
        <h3>Склад</h3>
        <p
className='text'>{data?.attributes.composition}</p>
    </div>
    <div className='section'>
        <h3>Опис</h3>
        <p className='text'>
            {data?.attributes.desc}
        </p>
    </div>
</div>
<div className="right">
    <h1>{data?.attributes.title}</h1>
    <label>В наявності</label><br/>
    <p className='price'>{data?.attributes.price}
грн</p>
    <div className='item'>
        <p className='weight' font-
opensans'>{data?.attributes.weight}</p>
        <p>Кількість:</p>
        <div className="quantity">
            <button onClick={() =>
setQuantity(prev => prev === 1 ? 1 : prev-1)}>-</button>
            <div className='cont'>{quantity}</div>
            <button onClick={() =>
setQuantity(prev => prev+1)}>+</button>
        </div>
    </div>
    <div className="item1">
        <button className='font-opensans'
onClick={() =>
dispatch(
    addtocart({
        id: data.id,

```

```

        title: data.attributes.title,
        desc: data.attributes.desc,
        price: data.attributes.price,
        img:
data.attributes.img.data.attributes.url,
        quantity,
    })
    )
    }>Додати в кошик</button>
<Link onClick={() =>
    dispatch(
        addtolike({
            id: data.id,
            title: data.attributes.title,
            weight: data.attributes.weight,
            price: data.attributes.price,
            img:
data.attributes.img.data.attributes.url,
            quantity,
        })
    )
    }></Link>
</div>
<h3>Доставка</h3>
<div className="section1">
    <div className="cont1">
        
        
        <div className='item2'>
            
            <p>Самовивіз</p>
        </div>
    </div>
    <div>
        <p><span style={{color:
'#00A60E'}}>Безкоштовно</span> <br/>Від 500 грн</p>
        <p><span style={{color:
'#00A60E'}}>Безкоштовно</span> <br/>Від 500 грн</p>
        <p><span style={{color:
'#00A60E'}}>Безкоштовно</span> <br/>Резерв товару на два дні</p>
    </div>
    </div>
    <h3>Оплата</h3>
    <div className='section2'>
        
        <p>Онлайн картою Visa/Mastercard</p>
    </div>
    <h3>Повернення</h3>

```

```
        <div className='section3'>
            <p>У нашому магазині існують такі <span
style={{color: '#00A60E', fontWeight: '500'}}>умови обміну</span> та
<span
            style={{color:
                '#00A60E',
                fontWeight:
                    '500'}}>повернення</span>:</p>
            <ul>
                <li>Сертифікати не підлягають
поверненню.</li>
                <li>Перевіряйте товар відразу на
відділенні або при доставці кур'ером.</li>
                <li>Якщо підписано документ про отримання,
то претензії на пошкодження товару не приймаються.</li>
            </ul>
        </div>
```

```
        <div className='section4'>
            <h5>Повернення товару здійснюється
відповідно до Закону України «Про захист прав споживачів».</h5>
            <p>Згідно постанови Кабінету міністрів
№172 від 19 березня 1994 року про реалізацію окремих положень закону
України "Про захист прав споживачів" не підлягають поверненню:</p>
            <ul>
                <li>продовольчі товари;</li>
                <li>товари медичного призначення:
лікарські препарати та прилади для лікування тварин, медичний
одяг;</li>
                <li>попони, бандажі та тп;</li>
                <li>предмети гігієни;</li>
                <li>м'які або надувні іграшки;</li>
                <li>товари для цуценят і кошенят (пелюшки,
соски, пляшечки для годування, поїльника і т.д.);</li>
                <li>парфумерно-косметичні та товари в
аерозольній упаковці;</li>
                <li>рушники, покривала, лежаки,
будиночки;</li>
                <li>зубні щітки, гребінці, щітки,
інструменти для грумінгу;</li>
                <li>панчішно-шкарпеткові вироби.</li>
            </ul>
        </div>
```

```
    </div>
</div>
</>
})
</div>
)
}

export default Product
```

## Лістинг розмітки та скриптів для корзини інтернет-магазину Dolittle

```
import React from "react";
import "./Cart.scss";
import { useDispatch, useSelector } from "react-redux";
import { removeItem, resetCart } from "../../redux/CartReducer";
import { Link } from "react-router-dom";

const Cart = () => {
  const products = useSelector((state) => state.cart.products);
  const dispatch = useDispatch();

  const totalPrice = () => {
    let total = 0;
    products.forEach((item) => {
      total += item.quantity * item.price;
    });
  };
  return total.toFixed(2);};

return (
  <div className="cart">
    <h1 className="font-opensans">Товари у вашій корзині</h1>
    {products?.map(item => (
      <div className="item font-motserrrat" key={item.id}>
        <div className="image">
          <img src={`/${item.img}`} alt="" />
        </div>
        <div className="details">
          <h2>{item.title?.substring(0, 35)}...</h2>
          <p>{item.desc?.substring(0, 100)}...</p>
          <div className="price">
            {item.quantity} x {item.price} грн
          </div>
        </div>
         dispatch(removeItem(item.id))} />
      </div>
    ))}
    <div className="total font-motserrrat">
      <span>Сума:</span>
      <span style={{color: 'red'}}>{totalPrice()} грн</span>
    </div>
    <Link to={"/checkout"} className="link"><button
      className="font-opensans">Купити</button></Link>
    <span className="reset font-opensans" onClick={() =>
      dispatch(resetCart())}>Скинути кошик</span>
    </div>
  )};};
export default Cart;
```

## Лістинг розмітки та скриптів для сторінки списку вподобаних товарів інтернет-магазину Dolittle

```

const LikePage = () => {
  const productslike = useSelector((state) =>
state.like.productslike);
  const dispatch = useDispatch();
  return (
    <div className="like">
      <h2 className="font-opensans">Список вподобаних
товарів</h2>
      <p className="reset font-opensans" onClick={() =>
dispatch(resetLike())}>Скинути вподобайки</p>
      <Link to={"/products/0"} className="link font-
motserrat"><div className="back">Вернутися до каталогу</div></Link>
      <div className="items">
        {productslike?.map(item =>(
          <div className="cards font-motserrat" key={item.id}>
            <div className="image">
              <div className="size">
                <img src={`\${item.img}`} alt="" />
              </div>
            </div>
            <h3>{item?.title}</h3>
            <div className="price">
              <div className="item"><p className="font-
opensans">{item.weight}</p></div>
              <h4>{item?.price} грн</h4>
            </div>
            <div className="item1">
               dispatch(removeLike(item.id))} />
              <div className="item2" ><p className="font-
opensans">Купити</p></div>
            </div>
          )))
        </div>
      </div>
    );
  };
}

export default LikePage;

```

## Лістинг розмітки та скриптів для функції пошуку товарів інтернет-магазину Dolittle

```

const SearchDropdown = () => {
  const [searchTerm, setSearchTerm] = useState('');
  const [selectedProduct, setSelectedProduct] = useState(null);
  const { data, loading, error } =
useFetch('/products?populate=*');
  const handleSearchChange = (event, value) => {
    setSearchTerm(value);};
  const handleProductSelect = (event, value) => {
    setSelectedProduct(value);};
  return (
    <div className="search-bar">
      <Autocomplete
        options={data || []}
        getOptionLabel={(option) => option.attributes.title}
        loading={loading}
        onChange={handleSearchChange}
        onInputChange={handleProductSelect}
        renderInput={(params) => (
          <TextField
            {...params}
            label="Пошук товарів"
            variant="outlined"
            size="small"
            InputProps={{
              ...params.InputProps,
              startAdornment: (
                <InputAdornment position="start">
                  <SearchIcon />
                </InputAdornment>
              ),}}
          />)}
        renderOption={(props, option) => (
          <li {...props}>
            <Link href={ `/product/${option.id}` }
              className="option-link">
              <img
                src={option.attributes.img.data.attributes.url}
                alt={option.attributes.title}
                className="option-img"
              />
              {option.attributes.title}
            </Link>
          </li>)}
        />
      </div>);};
export default SearchDropdown;

```

# Головна сторінка інтернет-магазину Dolittle

**Dolittle** Пошук товарів 👤 📍 🛒

## Швидко, якісно та смачно!

Швидка доставка, якісне обслуговування та смачний корм для вашого улюбленця.

[Каталог](#) [Про нас](#)



### Продукція

Ми пропонуємо широкий асортиментний набір товарів для тварин, представлений більш ніж 70 найкращими світовими та вітчизняними брендами.

[Детальніше](#)

- Реально підібрені товари
- Проста політика повернення
- Унікальний асортимент
- Етичне виробництво

### Купуй за категоріями


- Лежанки для собак та котів
- Кішки для гризунів та пташок
- Інструменти для утримання
- Товари для тваринців та акваріумів

### Блог

- 2023-07-22 **Виробництво сухого корму**
- 2023-10-04 **5 міфів про змії**
- 2023-01-28 **Здорове харчування котів**



# Сторінка каталогу інтернет-магазину Dolittle



👤
📌
🏠

🏠 Домашня > 📖 Каталог

## КАТАЛОГ

» Ціна

10 50000

» Категорії

Усі категорії

Собаки

Коти

Птахи

Гризуни

Рептилії

Риби

» Підкатегорії

Корні

Листя

Дрова

Шпатель та приладдя





















Аксесуари

Посуд




Котки

Антисептики

Товаруни та фігурки

 <p>Сухий корм для собак Royal Canin 6.3kg</p> <p>4.90 € 390 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Коричневий диван</p> <p>300 € 3000 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Платівка для підлоги</p> <p>1.80 € 300 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Траву для собак</p> <p>25.00 € 2500 грн</p> <p><span>📌</span> <span>👍</span></p>
 <p>Кормовий автомат</p> <p>2.50 € 475 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Сухий корм Brit</p> <p>1.40 € 280 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Коробка для гризунів</p> <p>12.00 € 1200 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Корм для собак</p> <p>8.00 € 160 грн</p> <p><span>📌</span> <span>👍</span></p>
 <p>Коричневий диван</p> <p>1.80 € 270 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Платівка для підлоги</p> <p>0.20 € 40 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Деталь для ремонту</p> <p>1.00 € 100 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Сухий корм</p> <p>8.00 € 160 грн</p> <p><span>📌</span> <span>👍</span></p>
 <p>Листя для пташок</p> <p>2.00 € 40 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Платівка для підлоги</p> <p>0.60 € 120 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Платівка для підлоги</p> <p>0.60 € 120 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Корм для собак</p> <p>2.00 € 40 грн</p> <p><span>📌</span> <span>👍</span></p>
 <p>Платівка для підлоги</p> <p>1.80 € 360 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Коричневий диван</p> <p>1.40 € 280 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Миска для гризунів</p> <p>0.80 € 160 грн</p> <p><span>📌</span> <span>👍</span></p>	 <p>Платівка для підлоги</p> <p>1.80 € 360 грн</p> <p><span>📌</span> <span>👍</span></p>

🏠 📖 👤 📌 🏠

Про нас

Оплата та доставка

INFO

Графік роботи: пн-пт  
10:00 до 19:00

📍 вул. Рухана, 30, Тернопіль, Тернопільська область

📧 dolittle@dolittle.com

☎ 096-399-0000

## Сторінка товару інтернет-магазину Dolittle

Головна > Каталог > Ласощі для гризунів Special One 2 шт. (овочевий мікс)

### Ласощі для гризунів Special One 2 шт. (овочевий мікс)

В наявності

80 грн

90 + Кількість: - 1 +

[Додати в кошик](#)

#### Доставка

	Безкоштовно Від 500 грн
	Безкоштовно Від 500 грн
Самовивіз	Безкоштовно Резерв товару на два дні

#### Оплата

Онлайн картою Visa/Mastercard

#### Повернення

У нашому магазині існують такі умови обміну та повернення:

- Сертифікати не підлягають поверненню.
- Перевіряйте товар відразу на відділенні або при доставці кур'єром.
- Якщо підписано документ про отримання, то претензії на пошкодження товару не приймаються.

**Повернення товару здійснюється відповідно до Закону України «Про захист прав споживачів».**

Згідно постанови Кабінету міністрів №172 від 19 березня 1994 року про реалізацію окремих положень закону України «Про захист прав споживачів» на підлягають поверненню:

- продовольчі товари;
- товари медичного призначення: лікарські препарати та прилади для лікування, товари медичний одяг;
- поліони, бандані та т.п.
- предмети гігієни;
- м'які або надувні іграшки;
- товари для шудженя і кошикет (пелюшки, сосни, пелюшки для годування, польована і т.д.);
- парфумерно-косметичні та товари в аерозольній упаковці;
- рюкзачки, покривала, лежанки, будиночки;
- зубні щітки, гребінці, щітки, інструменти для грунінгу;
- лан-шно-шарпетові вироби.

#### Характеристики

Бренд	Special One
Призначення	для шиншил
Вага	80 г
Виробник	Україна

#### Склад

Цибуля-порой. Покращує травлення та нормалізує мікрофлору кишочника. Гарбуз: Багатий на каротин, який покращує зір та позитивно впливає на стан шкіри і шерсті. Паприка: Джерело вітаміну С, вітамінів групи В, РР, Е. Поліпщує склад крові та колір шерсті. Мирра: Позитивно впливає на травлення, роботу серця та зір. Містить вітаміни групи В, D, Е, РР, С, К. Пастернак: Джерело вітаміну С та вітамінів групи В. Містить ефірну олію, корисну для ендокринної системи.

#### Опис

Рецептура ласощів розроблена спеціально для декоративних гризунів, які утримуються в домашніх умовах. Їхня відповідь: найкращим світовим стандартам годування і виключає продукти, багаті на поживні речовини, мікро- та макроелементи. Відбірні інгредієнти збагачують рацион вітамінами та дарують улюбленцям нові смакові відчуття, недоступні при звичайному домашньому годуванні. Предназначення улюблених ссаків та висока поживна цінність робить ці ласощі найкращим вибором для всіх пухлястиків.

[Про нас](#) [Оплата та доставка](#) [INFO](#)

Графік роботи: пн-пт  
10:00 до 17:00

Вул. Руська, 56 Тернопіль,  
Тернопільська область

Dolittle@gmail.com

098 000-0000