

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії  
програмного забезпечення

\_\_\_\_\_ Євген ДАВИДЕНКО

«\_\_» \_\_\_\_\_ 2026 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**  
**ВЕБЗАСТОСУНОК ІНТЕРНЕТ-МАГАЗИНУ КОМП'ЮТЕРНОЇ ТЕХНІКИ**

Спеціальність 121 Інженерія програмного забезпечення  
Освітня програма «Інженерія програмного забезпечення»

**Здобувач**

\_\_\_\_\_

**Богдан АРКУШЕНКО**

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Керівник роботи**

Ст. викладачка

\_\_\_\_\_

**Марина ФАЛЕНКОВА**

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Миколаїв – 2026**

## **Завдання на виконання кваліфікаційної роботи**

Чорноморський національний університет імені Петра Могили

Факультет	Комп'ютерних наук
Кафедра	Інженерії програмного забезпечення
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступінь	Бакалавр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії  
програмного забезпечення

\_\_\_\_\_ Євген ДАВИДЕНКО

«\_\_\_» \_\_\_\_\_ 2026 р.

### **ЗАВДАННЯ**

**на кваліфікаційну бакалаврську роботу здобувача**

**Аркушенко Богдана**

---

1. Тема кваліфікаційної роботи «Вебзастосунок інтернет-магазину комп'ютерної техніки» затверджена наказом ректора ЧНУ ім. Петра Могили № 349 від «26» грудня 2025 р.

2. Строк представлення кваліфікаційної роботи «\_\_\_» \_\_\_\_\_ 2026 р.

3. Очікуваний результат роботи та початкові дані якщо такі потрібні.

Очікуваним результатом є розроблений інтернет-магазин електроніки.

4. Перелік питань, що підлягають розробці:

- дослідження та аналіз існуючих аналогів;
- формування вимог до програмного забезпечення;
- визначення стеку технологій розробки;
- проєктування інтернет-магазину;

- розробка інтернет-магазину;
  - проведення аналізу результатів.
5. Перелік графічних матеріалів:
- презентація кваліфікаційної бакалаврської роботи.
6. Консультанти:

<b>Консультант</b>	<b>Кафедра (організація)</b>	<b>Частина роботи</b>

Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**КАЛЕНДАРНИЙ ПЛАН**  
**виконання кваліфікаційної роботи**

Тема: Вебзастосунок інтернет-магазину комп'ютерної техніки.

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КБР	01.02.2026	05.02.2026	Виконано
2	Огляд літератури за темою роботи	06.02.2026	15.02.2026	Виконано
3	Складання календарного плану КБР	16.02.2026	18.02.2026	Виконано
4	Аналіз предметної області	19.02.2026	28.02.2026	Виконано
5	Розробка проєктних рішень	01.03.2026	10.03.2026	Виконано
6	Моделювання та конструювання ПЗ	11.04.2026	20.04.2026	Виконано
7	Кодування, тестування та апробація ПЗ	21.03.2026	10.05.2026	Виконано
8	Відгук керівника КБР	01.06.2026	05.06.2026	Виконано
9	Оформлення КБР та презентації	20.05.2026	25.05.2026	Виконано
10	Попередній захист	25.05.2026	25.05.2026	Виконано
11	Рецензування	06.06.2026	10.06.2026	Виконано
12	Завершення оформлення КБР	26.05.2026	30.05.2026	Виконано
13	Захист кваліфікаційної роботи			

**Здобувач**

\_\_\_\_\_

**Богдан АРКУШЕНКО**

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Керівник роботи**

Старша викладачка

\_\_\_\_\_

**Марина ФАЛЕНКОВА**

«\_\_» \_\_\_\_\_ 20\_\_ р.

## АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи

«Вебзастосунок інтернет-магазину комп'ютерної техніки»

Здобувач 408 гр.: Аркушенко Богдан

Керівник: старша викладачка Фаленкова Марина

Актуальність теми кваліфікаційної бакалаврської роботи полягає в закритті потреб клієнтів та надання послуг продажу електроніки, в результаті чого є продукт, що використовується для автоматизації процесів управління клієнтами та послугами у сервісних компаніях. Система побудована з використанням трирівневої веб-архітектури, що передбачає розподіл на клієнтський рівень, сервер прикладної логіки та рівень зберігання даних, що забезпечує масштабованість, гнучкість та зручність супроводу програмного продукту. Обумовлено активним розвитком електронної комерції, зростанням потреби користувачів у швидкому пошуку товарів, зручному оформленні замовлень та безпечній взаємодії зонлайн-сервісами.

Об'єкт роботи: процес розробки інтернет-магазину електроніки.

Предмет роботи: технології розробки інтернет-магазину електроніки.

Мета: розробка інтернет-магазину електроніки для надання послуги продажу товару.

Кваліфікаційна бакалаврська робота складається з вступу, чотирьох розділів, висновків та переліку джерел посилання.

У вступі зазначена актуальність теми, мета та завдання, що вирішується, предмет дослідження кваліфікаційної бакалаврської роботи, а також об'єкт самої роботи.

У першому розділі надається аналіз ринку розробки та огляд існуючих аналогів інтернет-магазинів, визначення функціоналу та переваг, план виконання завдань. Формується згідно цього опис специфікації вимог до вебзастосунку, що сприяє виконанню роботи.

У другому розділі є опис процесу розробки по факту специфікації вимог. Моделюється об'єкт та предмет дослідження, що сприяє вирішенню задачі.

У третьому розділі йде детальний опис конструювання та моделювання вебзастосунок за допомогою діаграм, обрання стеку технологій.

У четвертому розділі наявний результат роботи над розробкою інтернет-магазину з аналізом створеного функціоналу.

У висновках зазначений аналіз проведеної роботи з результатами виконання, що є метою розробки інтернет-магазину електроніки.

Кваліфікаційна бакалаврська робота викладена на 61 сторінок, вона містить 4 розділи, 16 ілюстрацій, 8 таблиць, 32 джерел в переліку посилань.

*Ключові слова: інтернет-магазин, електронна комерція, React, JavaScript, Node.js, Express.js, MongoDB, REST API, JWT, вебзастосунок.*

# **ABSTRACT**

for the Bachelor's Thesis

“Web application for an online computer store”

Student of group 408: Arkushenko Bohdan

Supervisor: Senior Lecturer Falenkova Marina

Relevance of the bachelor’s thesis topic lies in meeting customer needs and providing electronics sales services, resulting in a product used to automate customer and service management processes in service companies. The system is built using a three-tier web architecture, which involves separation into a client tier, an application logic server, and a data storage tier, ensuring scalability, flexibility, and ease of software maintenance. This is driven by the active development of e-commerce, the growing user need for rapid product search, convenient checkout, and secure interaction with online services.

Object of the study: the development process of an electronics e-commerce store.

Subject of the study: technologies for developing an electronics e-commerce store.

Objective: to develop an electronics e-commerce store to provide product sales services.

The bachelor’s thesis consists of an introduction, four chapters, conclusions, and a list of references.

The Introduction outlines the relevance of the topic, the objective and tasks to be solved, the subject of the bachelor’s thesis research, as well as the object of the work itself.

Chapter 1 provides a market development analysis and a review of existing analogues of e-commerce stores, identifying their functionality and advantages, along with a task execution plan. Based on this, a requirements specification description for the web application is formed, which facilitates the execution of the work.

Chapter 2 describes the development process based on the requirements specification. The object and subject of the study are modeled, which contributes to solving the problem.

Chapter 3 provides a detailed description of the design and modeling of the web application using diagrams, as well as the choice of the technology stack.

Chapter 4 presents the results of the electronics e-commerce store development along with an analysis of the created functionality.

The Conclusions provide an analysis of the work performed, along with the results achieved, which represents the objective of developing the electronics e-commerce store.

The bachelor's thesis comprises 61 pages, containing 4 chapters, 16 illustrations, 8 tables, and 32 sources in the list of references.

*Keywords: e-commerce store, e-commerce, React, JavaScript, Node.js, Express.js, MongoDB, REST API, JWT, web application.*

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	6
1.1 Огляд сучасних інтернет-магазинів електроніки .....	6
1.2 Аналіз обраної області .....	9
1.3 Технічний аспект проблеми.....	11
Висновки до розділу 1 .....	14
2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ ...	16
2.1 Обґрунтування вибору технологій розробки.....	16
2.2 Розробка моделі предметної області.....	17
2.3 Специфікація вимог до інтернет-магазину електроніки.....	19
Висновки до розділу 2 .....	26
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ .....	28
3.1 Реалізація клієнтської частини вебзастосунку .....	28
3.2.1 Створення REST API.....	33
3.2.2 Реалізація авторизації користувачів .....	34
3.2.3 Організація роботи з базою даних MongoDB .....	35
3.2.4 Реалізація обробки замовлень .....	36
3.2.5 Адміністративний функціонал .....	37
Висновки до розділу 3 .....	37
4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	39
4.1 Планування процесу тестування .....	39
4.2 Розробка тестових сценаріїв .....	40
4.3 Результати тестування системи.....	44
Висновки до розділу 4.....	50
ВИСНОВКИ.....	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	52
ДОДАТОК А .....	55
ДОДАТОК Б.....	57

## ПЕРЕЛІК СКОРОЧЕНЬ

API (Application Programming Interface) – програмний інтерфейс застосунку.

CRUD (Create, Read, Update, Delete) – основні операції створення, читання, оновлення та видалення даних.

CSS (Cascading Style Sheets) – каскадні таблиці стилів.

DB (Database) – база даних.

DOM (Document Object Model) – об’єктна модель документа.

HTML (HyperText Markup Language) – мова розмітки гіпертекстових документів.

HTTP (HyperText Transfer Protocol) – протокол передачі гіпертексту.

HTTPS (HyperText Transfer Protocol Secure) – захищена версія протоколу HTTP.

IDE (Integrated Development Environment) – інтегроване середовище розробки.

JS (JavaScript) – мова програмування JavaScript.

JSON (JavaScript Object Notation) – формат обміну даними.

JWT (JSON Web Token) – токен для автентифікації та авторизації користувачів.

MongoDB – документоорієнтована система керування базами даних.

MVC (Model–View–Controller) – архітектурний шаблон побудови програмного забезпечення.

Node.js – програмна платформа для виконання JavaScript на стороні сервера.

NPM (Node Package Manager) – менеджер пакетів для Node.js.

REST (Representational State Transfer) – архітектурний стиль побудови вебсервісів.

REST API (Representational State Transfer Application Programming Interface) – програмний інтерфейс, побудований відповідно до принципів REST.

SPA (Single Page Application) – односторінковий вебзастосунок.

SQL (Structured Query Language) – мова структурованих запитів до баз даних.

TS (TypeScript) – типізоване розширення мови JavaScript.

UI (User Interface) – користувацький інтерфейс.

URL (Uniform Resource Locator) – уніфікований покажчик ресурсу.

UX (User Experience) – досвід взаємодії користувача із системою.

VCS (Version Control System) – система керування версіями програмного забезпечення.

## ВСТУП

**Актуальність теми** кваліфікаційної бакалаврської роботи полягає в закритті потреб клієнтів та надання послуг продажу електроніки, в результаті чого є продукт, що використовується для автоматизації процесів управління клієнтами та послугами у сервісних компаніях. Система побудована з використанням трирівневої веб-архітектури, що передбачає розподіл на клієнтський рівень, сервер прикладної логіки та рівень зберігання даних, що забезпечує масштабованість, гнучкість та зручність супроводу програмного продукту. Обумовлено активним розвитком електронної комерції, зростанням потреби користувачів у швидкому пошуку товарів, зручному оформленні замовлень та безпечній взаємодії зонлайн-сервісами.

**Об'єкт роботи:** процес розробки інтернет-магазину електроніки.

**Предмет роботи:** технології розробки інтернет-магазину електроніки.

**Мета:** розробка інтернет-магазину електроніки для надання послуги продажу товару.

Відповідно для досягнення визначеної мети необхідно вирішити наступні **завдання:**

- аналіз застосунків-аналогів;
- розробка специфікації вимог до програмного забезпечення;
- проектування інтернет-магазину;
- розробка інтернет-магазину;
- розробка front-end;
- розробка back-end.

**Сфера застосування:** інтернет-магазин електроніки поширений у сфері надання послуг продажу та розвитку торгівельного бізнесу. Інтернет-магазин сприяє активному пошуку потенційних клієнтів магазину.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд сучасних інтернет-магазинів електроніки

Електронна комерція сьогодні належить до найактивніше зростаючих сфер у галузі інформаційних технологій. Завдяки розвитку Інтернету та цифрових каналів взаємодії дедалі більше торговельних процесів відбувається саме в онлайн-просторі. Вагому роль у системі електронної комерції відіграють інтернет-магазини електроніки, що спеціалізуються на реалізації комп'ютерної техніки, мобільних пристроїв, комплектуючих, периферії та іншої технологічної продукції.

Інтернет-магазин у сучасному розумінні — це багатофункціональна інформаційна система, у межах якої поєднано інструменти демонстрації товарів, пошукові механізми, засоби опрацювання замовлень, платіжні сервіси та модулі адміністрування. Головне призначення такого програмного рішення полягає у створенні зручних умов для придбання товарів покупцями та забезпеченні ефективного керування торговельними процесами з боку адміністрації або власників бізнесу.

Серед найпопулярніших інтернет-магазинів електроніки на українському ринку доцільно виокремити Rozetka, Comfy, Foxtrot, Eldorado та Moyo. Зазначені платформи надають користувачам доступ до широкого асортименту товарів, розвиненої системи розподілу продукції за категоріями, інструментів для зіставлення характеристик, а також механізмів оцінювання та залишення відгуків.

Rozetka є одним із наймасштабніших онлайн-майданчиків України у сфері торгівлі. До ключових переваг цього ресурсу належать значний вибір товарів, ефективна пошукова система, різноманітні фільтри за параметрами та спрощений процес оформлення замовлення. Водночас велика кількість елементів інтерфейсу інколи може створювати труднощі для користувачів, які вперше працюють із системою.

Comfy орієнтується насамперед на продаж побутової техніки та електронних товарів. Характерними рисами цього інтернет-магазину є докладний опис продукції, якісне візуальне представлення товарів і поєднання з програмами

лояльності для клієнтів. Платформа також забезпечує зручне оформлення замовлень і доступ до консультаційної підтримки.

Foxtrot також посідає важливе місце серед провідних представників українського ринку електроніки. Цей ресурс дає змогу оперативно знаходити потрібні товари за допомогою системи пошуку та фільтрації, ознайомлюватися з актуальними акційними пропозиціями, а також перевіряти наявність продукції в офлайн-магазинах.

Дослідження наявних рішень свідчить про те, що більшість сучасних інтернет-магазинів мають подібний базовий набір функціональних можливостей. До нього, як правило, входять:

- перегляд каталогу товарів;
- пошук продукції за назвою;
- відбір товарів за визначеними характеристиками;
- упорядкування результатів пошуку;
- перегляд повного опису товару;
- реєстрація та вхід користувачів до системи;
- формування кошика;
- оформлення замовлення;
- доступ до історії придбань;
- адміністрування товарного каталогу.

Однією з ключових частин інтернет-магазину є підсистема керування товарами. Адміністратор повинен мати змогу створювати нові товарні позиції, змінювати наявні записи, вилучати неактуальні товари та здійснювати керування категоріями. Реалізація такого функціоналу потребує відповідної серверної логіки, а також механізмів розмежування прав доступу.

Переважає більшість сучасних вебзастосунків будується на основі клієнт-серверної архітектури. Клієнтська частина забезпечує подання інформації користувачеві та взаємодію з інтерфейсом, тоді як серверна відповідає за оброблення даних, взаємодію з базою даних і реалізацію бізнес-логіки. Такий

підхід сприяє підвищенню масштабованості програмного продукту та полегшує його подальшу підтримку.

Огляд існуючих інтернет-магазинів дає підстави стверджувати, що користувачі найбільше цінують швидкодію системи, зручність навігації, ефективність пошуку товарів і належний рівень захисту персональних даних. Саме тому під час проектування нового програмного продукту необхідно враховувати актуальні вимоги до продуктивності, безпеки та зручності використання вебзастосунків.

Отже, результати проведеного аналізу дають змогу зробити висновок, що для розроблення конкурентоспроможного інтернет-магазину електроніки потрібно передбачити інтуїтивно зрозумілий інтерфейс користувача, систему автентифікації, механізми керування товарами, кошик, засоби оформлення замовлень та адміністративну панель для управління вмістом. Урахування сильних сторін уже наявних аналогів сприятиме створенню ефективного програмного продукту, що відповідатиме сучасним потребам електронної комерції.

Порівняльну характеристику типових аналогів інтернет-магазину електроніки наведено у таблиці 1.1.

Таблиця 1.1 – Типи аналогічних систем

<b>Тип системи</b>	<b>Переваги</b>	<b>Недоліки</b>	<b>Корисні функції</b>	<b>Висновок для проєкту</b>
Маркетплейс	великий каталог, багато продавців	перевантажений інтерфейс	пошук, фільтри, рейтинг	використати пошук і категорії
Спеціалізований магазин	детальні характеристики електроніки	складна адмін-панель	порівняння, фільтрація	додати характеристики товарів
Локальний магазин	простота, швидкість	обмежений функціонал	кошик, оформлення замовлення	зберегти просту навігацію

## 1.2 Аналіз обраної області

Предметною областю цієї кваліфікаційної роботи виступає електронна комерція у сегменті онлайн-продажу електроніки. Електронна комерція охоплює комплекс технологічних рішень, програмних інструментів і бізнес-процесів, за допомогою яких забезпечується здійснення торговельної діяльності в цифровому середовищі. Її ключове призначення полягає в оптимізації взаємодії між продавцем і покупцем, автоматизації опрацювання замовлень та підвищенні загальної ефективності торговельних операцій.

Галузь реалізації електроніки через мережу Інтернет відзначається широким спектром товарних категорій, значною кількістю технічних параметрів продукції та постійними змінами асортименту. До найпоширеніших видів електронних товарів належать смартфони, планшети, ноутбуки, персональні комп'ютери, монітори, комплектуючі, периферійне обладнання, телевізори, навушники, акустичні системи, мережеві пристрої та інші технічні засоби.

Специфікою цієї предметної області є потреба в опрацюванні й зберіганні великого обсягу даних про товари. Кожна одиниця продукції має власний набір характеристик, який визначається її категорією. Наприклад, для ноутбуків суттєвими параметрами є тип і модель процесора, обсяг оперативної пам'яті, різновид накопичувача та розмір екрана. Для смартфонів важливими залишаються операційна система, місткість вбудованої пам'яті, параметри камери та ємність акумулятора.

У межах аналізу предметної області було виокремлено основних учасників, які взаємодіють із системою інтернет-магазину електроніки. До таких учасників належать незареєстровані користувачі, зареєстровані користувачі та адміністратори.

Незареєстрований користувач може переглядати каталог продукції, виконувати пошук товарів, ознайомлюватися з їхніми характеристиками та отримувати загальну інформацію про магазин. Його основна мета полягає в отриманні необхідних відомостей про асортимент і прийнятті рішення щодо можливого придбання товару.

Зареєстрований користувач має розширений перелік можливостей у системі. Після проходження авторизації він може додавати товари до кошика, оформлювати замовлення, переглядати історію власних покупок і редагувати персональні дані. Наявність особистого кабінету сприяє підвищенню зручності взаємодії між клієнтом та інтернет-магазином.

Адміністратор виконує функції підтримки працездатності програмного продукту й керування інформаційним наповненням системи. До його обов'язків належать створення нових товарних записів, редагування наявних позицій, управління категоріями, опрацювання замовлень клієнтів та контроль стабільного функціонування програмного забезпечення.

Дослідження бізнес-процесів предметної області дало змогу виокремити кілька базових сценаріїв роботи системи. Перший із них пов'язаний із пошуком і переглядом продукції. У цьому випадку користувач звертається до каталогу, знаходить потрібний товар за назвою або категорією та аналізує його характеристики.

Наступним важливим процесом є формування кошика покупок. Після вибору відповідної продукції користувач додає її до кошика, а система автоматично обчислює загальну вартість замовлення залежно від кількості вибраних товарів.

Ще одним ключовим бізнес-процесом виступає оформлення замовлення. На цьому етапі користувач вносить контактні дані, указує адресу доставки та підтверджує покупку. Після завершення цієї процедури відомості про замовлення фіксуються в базі даних і стають доступними для подальшого опрацювання адміністратором.

Суттєве значення в межах предметної області має організація структури даних. Для належного функціонування інтернет-магазину необхідно забезпечити зберігання відомостей про користувачів, товари, товарні категорії та замовлення. Сукупність зв'язків між цими сутностями формує основу інформаційної моделі та бази даних майбутньої системи.

З позиції програмної реалізації інтернет-магазин доцільно розглядати як багаторівневу інформаційну систему. Клієнтський рівень забезпечує візуальне

подання інформації та взаємодію користувача з інтерфейсом. Серверний рівень відповідає за оброблення запитів, реалізацію бізнес-логіки та взаємодію з базою даних. Рівень зберігання даних призначений для довготривалого розміщення інформації про товари, користувачів і замовлення.

Окремої уваги потребують питання інформаційної безпеки. Оскільки система працює з персональними даними користувачів, необхідно впровадити механізми автентифікації й авторизації, застосовувати хешування паролів, а також забезпечити розмежування доступу до адміністративного функціоналу.

Проведений аналіз предметної області дав змогу визначити основні функціональні можливості майбутнього програмного продукту. Система має підтримувати роботу з каталогом товарів, реєстрацію та авторизацію користувачів, формування кошика, оформлення замовлень, управління товарами й категоріями, а також перегляд інформації про виконані покупки.

### **1.3 Технічний аспект проблеми**

Інтенсивний розвиток інформаційних технологій і постійне збільшення кількості користувачів мережі Інтернет зумовили зростання потреби у веборієнтованих інформаційних системах. Серед найпоширеніших різновидів таких рішень важливе місце займають інтернет-магазини, призначені для автоматизації продажу товарів і організації взаємодії між продавцем та покупцем. Попри наявність значної кількості готових програмних продуктів, завдання створення продуктивного, захищеного та масштабованого інтернет-магазину й надалі залишається актуальним.

Однією з ключових технічних проблем є забезпечення стабільного та швидкого функціонування системи в умовах одночасної роботи великої кількості користувачів. Під час експлуатації інтернет-магазину програмне забезпечення має обробляти численні запити, що стосуються пошуку товарів, перегляду їх характеристик, додавання позицій до кошика, оформлення замовлень і виконання адміністративних дій.

Не менш важливим аспектом виступає організація зберігання та опрацювання даних. Інтернет-магазин електроніки повинен забезпечувати ефективно накопичення відомостей про товари, категорії, користувачів і замовлення. Водночас структура даних має бути побудована таким чином, щоб підтримувати подальше розширення функціональності без необхідності суттєвого перепроєктування програмного коду. Значна кількість технічних характеристик товарів ускладнює моделювання бази даних і потребує застосування сучасних підходів до керування інформацією.

Окремої уваги потребує проблема створення зручного користувацького інтерфейсу. Користувач повинен мати змогу швидко знаходити потрібну продукцію, ознайомлюватися з її параметрами та здійснювати покупку за мінімальну кількість дій. У разі складної навігації або незручної структури інтерфейсу ефективність системи знижується, що може негативно вплинути на кількість потенційних клієнтів.

Сучасний вебзастосунок також повинен підтримувати адаптивність інтерфейсу для коректного відображення інформації на різних типах пристроїв. Оскільки значна частина користувачів здійснює покупки за допомогою смартфонів і планшетів, програмний продукт має бути однаково зручним як для персональних комп'ютерів, так і для мобільних пристроїв.

Важливим напрямом технічного проектування є забезпечення належного рівня безпеки. Інтернет-магазини працюють із персональними даними користувачів, зокрема контактною інформацією, історією замовлень та обліковими записами. Несанкціонований доступ до таких відомостей може призвести до їх витоку або втрати. Саме тому під час розроблення системи необхідно передбачити сучасні механізми автентифікації та авторизації.

Одним із поширених підходів до захисту вебзастосунків є використання технології JWT (JSON Web Token). Такий механізм дає змогу реалізувати безпечну авторизацію користувачів шляхом передавання спеціального токена між клієнтською та серверною частинами системи. Застосування JWT сприяє контролю

доступу до функцій програмного забезпечення та загальному підвищенню рівня захищеності інформаційної системи.

Для розроблення клієнтської частини інтернет-магазину доцільно застосовувати бібліотеку React, яка є однією з найпоширеніших технологій створення сучасних вебінтерфейсів. Її суттєвою перевагою є компонентний підхід, що дозволяє формувати багаторазово використовувані елементи інтерфейсу та спрощує супровід програмного продукту.

Використання React позитивно впливає на швидкодію вебзастосунку завдяки застосуванню Virtual DOM. Цей механізм дає змогу скоротити кількість безпосередніх оновлень сторінки та забезпечити швидке відображення змін в інтерфейсі користувача.

Для реалізації серверної частини системи доцільно використовувати платформу Node.js у поєднанні з вебфреймворком Express.js. Node.js забезпечує виконання JavaScript-коду на сервері та демонструє високу ефективність при обробленні значної кількості паралельних запитів. У свою чергу Express.js полегшує створення серверної логіки та розроблення REST API.

REST API виконує роль засобу взаємодії між клієнтською та серверною частинами системи. За допомогою HTTP-запитів клієнт отримує дані про товари, передає відомості для авторизації, створює замовлення та ініціює інші операції. Така модель сприяє розмежуванню функціональних обов'язків між компонентами системи та покращує її масштабованість.

Для зберігання інформації доцільно застосовувати систему керування базами даних MongoDB. На відміну від реляційних систем, MongoDB реалізує документоорієнтований підхід, що надає більшу гнучкість у роботі зі структурою даних і спрощує збереження складних об'єктів, характерних для інтернет-магазинів електроніки.

У межах розроблюваної системи передбачається формування кількох основних сутностей бази даних, серед яких користувачі, товари, категорії та замовлення. Кожна з цих сутностей повинна містити набір атрибутів, необхідних для реалізації повного функціоналу програмного продукту.

Додатковим технічним завданням є організація ефективного керування станом клієнтського застосунку. У процесі роботи інтернет-магазину необхідно зберігати інформацію про авторизованого користувача, вміст кошика, результати пошуку та інші проміжні дані. Для централізованого управління станом може бути використана бібліотека Redux Toolkit, яка спрощує оброблення даних у межах вебзастосунку.

## **Висновки до розділу 1**

У першому розділі кваліфікаційної роботи було здійснено дослідження предметної області, пов'язаної з розробленням інтернет-магазину електроніки, а також проаналізовано сучасні тенденції розвитку електронної комерції.

У результаті проведеного аналізу встановлено, що електронна комерція належить до найбільш динамічних напрямів розвитку інформаційних технологій. Сучасні споживачі все частіше обирають онлайн-покупки завдяки зручності використання вебресурсів, широкому вибору продукції та можливості оперативного порівняння характеристик товарів.

У межах розділу було розглянуто наявні інтернет-магазини електроніки та проаналізовано їхні функціональні можливості. Визначено основні переваги сучасних програмних рішень, серед яких широкий асортимент товарів, наявність пошуку та фільтрації, механізми реєстрації користувачів, кошик покупок, засоби оформлення замовлень і адміністративні інструменти керування вмістом. Водночас було виокремлено й окремі недоліки існуючих систем, зокрема складність інтерфейсу, перевантаженість функціональними елементами та підвищені вимоги до адміністрування.

Під час аналізу предметної області було встановлено основних учасників системи, а саме незареєстрованих користувачів, зареєстрованих користувачів та адміністраторів. Для кожної з визначених груп окреслено перелік функціональних можливостей, необхідних для повноцінної та результативної взаємодії з інформаційною системою.

Крім того, було досліджено ключові бізнес-процеси інтернет-магазину електроніки, до яких належать перегляд каталогу товарів, пошук продукції, формування кошика, оформлення замовлень та адміністрування товарного асортименту. Опрацювання цих процесів дало змогу сформувавши перелік основних функціональних вимог до майбутнього програмного забезпечення.

Окрему увагу в межах розділу приділено технічним аспектам розроблення системи. Було визначено базові вимоги до архітектури програмного продукту, структури бази даних, захисту персональних даних користувачів і забезпечення належної продуктивності. Також обґрунтовано доцільність використання сучасних вебтехнологій для реалізації проєкту, зокрема React для створення клієнтської частини, Node.js та Express.js для серверного рівня, MongoDB для зберігання інформації та JWT для реалізації безпечної авторизації користувачів.

## 2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ

### 2.1 Обґрунтування вибору технологій розробки

Для створення клієнтської частини застосунку було обрано React, оскільки ця бібліотека надає зручну компонентну модель, сприяє швидкій розробці та підтримує сучасні підходи до побудови односторінкових вебзастосунків. React-компоненти можуть використовуватися повторно, окремо тестуватися та комбінуватися для формування складних сторінок, що суттєво полегшує подальший супровід програмного забезпечення.

Як основну мову програмування для клієнтського та серверного рівнів використано JavaScript. Застосування єдиної мови для всієї системи дає змогу спростити процес розроблення, зменшити складність підтримки проєкту та забезпечити більш цілісну організацію програмного коду. Для реалізації навігації між сторінками використовується React Router, який забезпечує швидкий перехід між розділами без потреби у повному перезавантаженні сторінки.

Для побудови серверної частини було обрано платформу Node.js, яка дозволяє виконувати JavaScript-код на сервері та ефективно обробляти велику кількість одночасних запитів. У ролі серверного фреймворку застосовується Express.js, що спрощує розроблення REST API, налаштування маршрутів, оброблення HTTP-запитів і підключення додаткових програмних модулів.

Для зберігання даних використовується система керування базами даних MongoDB. Вибір цієї технології пояснюється її гнучкою моделлю даних і зручністю роботи з документами JSON-подібної структури. Для інтернет-магазину електроніки це є особливо важливим, оскільки різні категорії товарів можуть мати відмінний набір характеристик. Наприклад, ноутбук описується такими параметрами, як процесор, обсяг оперативної пам'яті та тип накопичувача, тоді як для навушників більш важливими є тип підключення, тривалість автономної роботи та наявність функції шумозаглушення.

Для реалізації механізму авторизації користувачів використовується технологія JWT. Після успішного входу в систему сервер формує токен доступу,

який зберігається на стороні клієнта та надалі використовується під час надсилання захищених запитів. Такий підхід дає змогу забезпечити надійну взаємодію між клієнтською та серверною частинами застосунку.

Для оформлення інтерфейсу застосовуються CSS і сучасні засоби адаптивної верстки. Це забезпечує коректне відображення сторінок на різних типах пристроїв, зокрема на персональних комп'ютерах, планшетах і смартфонах.

Як методологію розроблення обрано Kanban. Цей підхід дає змогу впорядкувати виконання завдань, контролювати поточний стан проєкту та своєчасно реагувати на зміну вимог. Для навчального проєкту використання Kanban є доцільним завдяки простоті застосування та наочності організації процесу розроблення.

Таблиця 2.1 – Вибір стеку технологій

Технологія	Призначення	Причина вибору
React	клієнтський інтерфейс	компонентність і швидка розробка
Node.js	серверне середовище	єдиний стек JavaScript
Express.js	REST API	простота маршрутизації
MongoDB	база даних	гнучка структура документів
JWT	авторизація	захист маршрутів і масштабованість
Axios	HTTP-запити	зручна робота з API

## 2.2 Розробка моделі предметної області

Модель предметної області відображає ключові сутності інформаційної системи та взаємозв'язки між ними. Для інтернет-магазину електроніки до основних сутностей доцільно віднести користувача, товар, категорію, кошик, замовлення, позицію замовлення та адміністратора.

Сутність користувача включає ідентифікатор, ім'я, адресу електронної пошти, пароль у хешованому вигляді та роль. Саме роль визначає рівень доступу до функціональних можливостей системи: звичайний користувач може переглядати каталог продукції, додавати товари до кошика та оформлювати

замовлення, тоді як адміністратор має права на керування товарним каталогом і оброблення замовлень.

Сутність товару містить назву, опис, вартість, категорію, бренд, зображення, залишок на складі та набір технічних характеристик. Для сфери продажу електроніки важливо забезпечити підтримку додаткових характеристик, оскільки кожна категорія товарів має власні специфічні параметри.

Замовлення пов'язується з конкретним користувачем і включає перелік вибраних товарів, загальну суму, адресу доставки, контактний номер телефону, поточний статус та дату створення. Серед основних статусів замовлення можна виділити такі: «нове», «обробляється», «відправлено», «виконано» та «скасовано».

Кошик являє собою тимчасову структуру даних, у якій зберігаються товари, обрані користувачем до моменту завершення оформлення замовлення. На клієнтському рівні кошик може зберігатися у стані застосунку або в localStorage, а після авторизації користувача — синхронізуватися із серверною частиною системи.

Таблиця 2.2 – Узагальнення варіантів використання інтернет-магазину електроніки

<b>Актор</b>	<b>Варіанти використання</b>
Гість	перегляд каталогу, пошук товарів, реєстрація, вхід
Користувач	додавання до кошика, редагування кошика, оформлення замовлення, перегляд профілю
Адміністратор	додавання товарів, редагування товарів, видалення товарів, перегляд замовлень

На підставі визначених сутностей і встановлених між ними зв'язків формується структура бази даних та загальна архітектура інформаційної системи, що слугує основою для подальшого розроблення програмного забезпечення.

## 2.3 Специфікація вимог до інтернет-магазину електроніки

### СПЕЦИФІКАЦІЯ ВИМОГ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ ЕЛЕКТРОНІКИ

Призначення та межі проєкту

#### 1.1 Призначення системи

Вебзастосунок інтернет-магазину електроніки призначений для автоматизації процесів роздрібної торгівлі цифровою та побутовою технікою через мережу Інтернет. Система забезпечує повний цикл взаємодії покупця з товаром, надаючи користувачам швидкий та коректний доступ до каталогу товарів, технічних характеристик, інструментів пошуку, фільтрації, віртуального кошика, а також забезпечує автоматизацію створення й обробки замовлень.

#### 1.2 Погодження, ухвалені в програмній документації

Специфікація розроблена на основі технічного завдання кваліфікаційної роботи, вимог замовника та стандартів захисту персональних даних (GDPR, Закону України «Про захист персональних даних»);

Ухвалено використання єдиного технологічного стеку на базі мови JavaScript (Node.js + React) для оптимізації швидкості розробки та підтримки системи;

Визначено перелік пріоритетних бізнес-процесів для запуску MVP (мінімально життєздатного продукту), що включає каталог товарів з фільтрацією, авторизацію, кошик, модуль оформлення замовлень та базову адміністративну панель.

#### 1.3 Межі проєкту

Проєкт охоплює розробку клієнтської частини (SPA-інтерфейс для покупців) та серверної частини (REST API + база даних), а також модуля адміністрування для керування контентом і замовленнями;

Система передбачає гнучку документо-орієнтовану структуру бази даних для збереження різнотипних технічних характеристик техніки (смартфони, ноутбуки, периферія тощо);

Передбачена робота над адаптивним дизайном застосунку для коректного відображення на мобільних пристроях, планшетах та ПК;

Проект у межах MVP не охоплює інтеграцію з фізичним обладнанням складів (сканери штрих-кодів) чи пряму інтеграцію з касовими апаратами (РРО/ПРРО).

## 2. Загальний опис

### 2.1 Сфера застосування

Вебзастосунок використовується як комерційна та інформаційна платформа для здійснення онлайн-продажів. Застосунок підтримує функції електронного каталогу, розширеного пошуку, динамічної фільтрації за технічними параметрами, брендами та ціною, управління кошиком, відстеження історії покупок, а також адміністрування товарних позицій і категорій.

### 2.2 Характеристики користувачів

Незареєстрований користувач (Гість) — фізична особа, яка відвідує ресурс з метою ознайомлення. Має доступ до перегляду каталогу, пошуку товарів, вивчення їхніх характеристик та зображень;

Зареєстрований користувач (Покупець) — клієнт магазину, який пройшов процедуру аутентифікації. Може додавати товари до кошика, керувати його вмістом, оформлювати замовлення та переглядати історію власних покупок в особистому кабінеті;

Адміністратор — персонал магазину, який має повний доступ до внутрішньої системи управління для додавання, редагування й видалення товарних позицій, керування категоріями товарів та моніторингу й обробки замовлень.

Рівень підготовки користувачів (гості та покупці) передбачає наявність базових навичок роботи з веббраузерами та інтернет-ресурсами. Адміністратори повинні володіти середнім рівнем технічної підготовки для роботи з інтерфейсами систем управління контентом (CMS).

## 3. Функції системи

### 3.1 Авторизація та аутентифікація

#### 3.1.1 Опис функції

Дозволяє користувачам створювати облікові записи та входити в систему під своїми обліковими даними для отримання доступу до персоналізованого функціоналу (кошик, замовлення, історія) та адміністративних маршрутів.

### 3.1.2 Вхідна і вихідна інформація

Вхідна: реєстраційні дані (ПІБ, email, пароль) або дані для входу (email, пароль);

Вихідна: токен доступу (JWT), статус авторизації, роль користувача (покупець/адміністратор), дані профілю.

### 3.1.3 Функціональні вимоги

Забезпечення унікальності email-адрес при реєстрації користувачів;

Хешування та безпечне зберігання паролів у базі даних (використання алгоритмів шифрування/хешування на стороні сервера);

Захист адміністративних та клієнтських маршрутів (маршрутизація на базі React Router та перевірка JWT-токена у HTTP-заголовках запитів).

## 3.2 Перегляд каталогу, пошук та фільтрація товарів

### 3.2.1 Опис функції

Надає користувачеві можливість зручної навігації за категоріями електроніки, швидкого пошуку за назвою, а також детальної фільтрації асортименту за ціною, брендом та специфічними технічними параметрами.

### 3.2.2 Вхідна і вихідна інформація

Вхідна: ID категорії, пошуковий запит (текстовий рядок), параметри фільтрації (діапазон цін, виробник, технічні характеристики);

Вихідна: динамічний список товарів із зображеннями, цінами та коротким описом, або картка конкретного товару з повним набором характеристик.

### 3.2.3 Функціональні вимоги

Динамічне оновлення списку товарів без повного перезавантаження сторінки;

Підтримка гнучкої структури характеристик (відображення унікальних полів для кожної категорії, наприклад, процесор для ноутбуків або тип підключення для навушників);

Сортування товарів за ціною (від дешевих до дорогих і навпаки) та новизною.

## 3.3 Керування кошиком та оформлення замовлення

### 3.3.1 Опис функції

Забезпечує можливість збирання вибраних товарних позицій у віртуальний кошик, редагування їх кількості та підтвердження наміру купівлі шляхом створення замовлення.

### 3.3.2 Вхідна і вихідна інформація

Вхідна: ID товару, кількість одиниць, контактні дані для доставки, спосіб отримання;

Вихідна: сформоване замовлення з унікальним ID, зміна статусу замовлення, оновлення кількості товарів на складі.

### 3.3.3 Функціональні вимоги

Збереження стану кошика користувача між сесіями;

Автоматичний перерахунок загальної вартості кошика при зміні кількості або видаленні товарів;

Перевірка наявності товару на складі перед фінальним підтвердженням замовлення.

## 3.4 Адміністративне керування (Панель адміністратора)

### 3.4.1 Опис функції

Дозволяє адміністраторам здійснювати повний контроль над контентом інтернет-магазину та керувати процесом виконання замовлень.

### 3.4.2 Вхідна і вихідна інформація

Вхідна: дані нового товару (назва, категорія, ціна, опис, технічні параметри, фото), ID замовлення для зміни статусу;

Вихідна: оновлена база даних товарів, змінений статус замовлення в системі.

### 3.4.3 Функціональні вимоги

Можливість створення, редагування та видалення (CRUD) товарів і категорій;

Перегляд списку всіх замовлень із можливістю фільтрації за статусом (очікує обробки, відправлено, виконано, скасовано).

## 4. Вимоги до інформаційного забезпечення

### 4.1 Джерела і зміст вхідної інформації

Персональні дані покупців (ПІБ, email, телефон, адреса доставки), що вносяться через інтерфейси реєстрації та оформлення замовлення;

Контентне наповнення (описи товарів, зображення, технічні специфікації), що вноситься адміністратором через панель управління;

Системні логи та дані про стан сесій (JWT-токени), що генеруються під час взаємодії користувача із сервером.

#### 4.2 Нормативно-довідкова інформація

Довідник категорій та підкатегорій товарів електроніки;

Схеми метаданих технічних характеристик для різних типів пристроїв;

Класифікатор статусів замовлень (в обробці, оплачено, відправлено, доставлено, скасовано).

#### 4.3 Вимоги до способів організації, збереження та ведення інформації

Дані зберігаються в документо-орієнтованій базі даних MongoDB, що забезпечує гнучкість збереження неструктурованих характеристик товарів;

Організація взаємодії з базою даних здійснюється через серверні моделі даних (наприклад, за допомогою Mongoose);

Передбачено регулярне створення резервних копій (бэкапів) бази даних для запобігання втраті інформації.

#### 5. Вимоги до технічного забезпечення

Сервери — хмарні сервери або хостинг із підтримкою середовища Node.js та бази даних MongoDB;

Мережа — стабільне підключення до мережі Інтернет, підтримка захищеного обміну даними через протокол HTTPS;

Клієнтські пристрої — сучасні ПК, ноутбуки, планшети та смартфони користувачів, обладнані актуальними версіями веббраузерів (Google Chrome, Mozilla Firefox, Safari, Edge).

#### 6. Вимоги до програмного забезпечення

##### 6.1 Архітектура програмної системи

Клієнт-серверна архітектура (SPA + REST API) — чітке розділення фронтенд-частини (React) та бекенд-частини (Node.js/Express);

Компонентна модель — побудова інтерфейсу користувача з незалежних компонентів React, що забезпечує перевикористання коду та простоту розширення системи;

Керування станом — використання сучасних механізмів управління станом (Context API або Redux Toolkit) для збереження даних кошика, авторизації та результатів фільтрації без перезавантаження сторінок.

## 6.2 Системне програмне забезпечення

ОС серверів — операційні системи сімейства Linux/Unix (наприклад, Ubuntu Server) або спеціалізовані хмарні платформи;

Середовище виконання — Node.js (актуальна LTS версія).

## 6.3 Мережне програмне забезпечення

Протокол передачі даних — HTTP/HTTPS для взаємодії клієнта з REST API;  
Захист з'єднання — використання SSL/TLS сертифікатів для шифрування трафіку.

## 6.4 Програмне забезпечення ведення інформаційної бази

СКБД — нереляційна (NoSQL) база даних MongoDB;

Управління моделями — використання Object Data Modeling (ODM) бібліотеки Mongoose для валідації та структурування документів.

## 6.5 Мова і технологія розробки

Frontend: мова JavaScript, бібліотека React, маршрутизатор React Router, Axios для виконання асинхронних HTTP-запитів до API, CSS/Tailwind CSS для адаптивної стилізації;

Backend: мова JavaScript, платформа Node.js, вебфреймворк Express.js;

Авторизація: бібліотека розробки та перевірки JSON Web Tokens (JWT).

## 7. Вимоги до зовнішніх інтерфейсів

### 7.1 Інтерфейс користувача

Адаптивність — гнучка верстка, яка адаптується під будь-які розширення екранів (від смартфонів до широкоформатних моніторів);

Юзабіліті — проста навігація, зрозуміла структура категорій, швидкий доступ до кошика з будь-якої сторінки застосунку.

## 7.2 Апаратний інтерфейс

Спеціалізований апаратний інтерфейс не потрібен. Взаємодія відбувається через стандартні периферійні пристрої (миша, клавіатура, тачскрін).

## 7.3 Програмний інтерфейс

Система реалізує внутрішній програмний інтерфейс REST API, який забезпечує отримання, створення, оновлення та видалення даних між клієнтом та сервером за допомогою стандартних методів (GET, POST, PUT, DELETE).

## 7.4 Комунікаційний протокол

Обмін даними між компонентами системи (фронтенд і бекенд) здійснюється у форматі JSON (JavaScript Object Notation) через захищений протокол HTTPS.

## 8. Властивості програмного забезпечення

### 8.1 Доступність

Вебзастосунок має бути доступний для користувачів у режимі 24/7. Допускаються короткочасні перерви для проведення планових технічних робіт на сервері у години найменшої активності користувачів.

### 8.2 Супроводжуваність

Завдяки компонентній структурі React та модульності Express.js, система дозволяє легко оновлювати або додавати нові функції (наприклад, модуль порівняння товарів чи систему відгуків);

Код повинен бути структурованим та за необхідності прокоментованим для швидкої передачі проекту іншим розробникам.

### 8.3 Переносимість

Програмне забезпечення є кросплатформним на рівні клієнта (працює у будь-якому сучасному браузері) та легко переноситься на будь-яку серверну інфраструктуру, де встановлено Node.js.

### 8.4 Продуктивність

Оптимізація рендерингу компонентів React для забезпечення миттєвого відгуку інтерфейсу на дії користувача;

Асинхронна обробка запитів на сервері Node.js, що дозволяє обслуговувати велику кількість одночасних з'єднань без зниження швидкості роботи застосунку.

## 8.5 Надійність

Коректна обробка помилок (наприклад, відсутність зв'язку з базою даних або невірні HTTP-запити) з виведенням зрозумілих повідомлень для користувача;

Валідація даних на стороні клієнта (перед відправкою форм) та обов'язкова повторна валідація на стороні сервера перед записом у БД.

## 8.6 Безпека

Захист від поширених вебвразливостей;

Зберігання конфіденційних даних (паролів) виключно у хешованому вигляді;

Обмеження доступу до адміністративного функціоналу на рівні перевірки прав доступу (ролей) у JWT-токені.

## 9. Інші вимоги

Конфіденційність — суворе дотримання правил збереження та обробки персональних даних покупців інтернет-магазину;

Масштабованість — архітектурна можливість подальшого розширення бізнес-логіки системи (впровадження онлайн-оплати через платіжні шлюзи типу LiqPay, підключення API логістичних служб, наприклад, Нової Пошти, додавання аналітичних модулів для відстеження обсягів продажів).

## **Висновки до розділу 2**

У другому розділі було здійснено моделювання та проєктування програмного комплексу інтернет-магазину електроніки. На основі результатів аналізу предметної області визначено ключові вимоги до програмного забезпечення та сформовано загальну архітектуру майбутньої інформаційної системи.

У межах розділу обґрунтовано вибір технологічного стеку, використаного для реалізації проєкту. Для побудови клієнтської частини застосунку обрано бібліотеку React, яка забезпечує компонентний підхід до створення користувацького інтерфейсу та підтримує розроблення односторінкових вебзастосунків. Для реалізації серверної частини використано платформу Node.js і фреймворк Express.js, що дають змогу створити REST API та організувати ефективну взаємодію між клієнтською частиною й базою даних. Як систему

зберігання інформації обрано MongoDB, яка характеризується гнучкою структурою даних і є доцільною для роботи з різними категоріями електронної продукції.

У процесі моделювання предметної області було виокремлено основні сутності системи, зокрема користувачів, товари, категорії, кошик і замовлення. На основі встановлених зв'язків між цими сутностями сформовано логічну структуру бази даних, що забезпечує підтримку основних бізнес-процесів інтернет-магазину.

Крім того, у розділі було розроблено специфікацію вимог до програмного забезпечення. Визначено основні функціональні можливості системи, серед яких перегляд каталогу товарів, пошук і фільтрація продукції, реєстрація та авторизація користувачів, керування кошиком, оформлення замовлень та адміністрування каталогу. Також сформовано перелік нефункціональних вимог, які стосуються продуктивності, безпеки, масштабованості, адаптивності та надійності функціонування системи.

Отримані результати створюють необхідне підґрунтя для практичної реалізації програмного забезпечення та переходу до наступного етапу роботи, пов'язаного з розробленням клієнтської та серверної частин інтернет-магазину електроніки.

## 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ

### 3.1 Реалізація клієнтської частини вебзастосунку

Клієнтську частину інтернет-магазину електроніки реалізовано з використанням бібліотеки React. Такий підхід дає змогу створити сучасний односторінковий вебзастосунок, який забезпечує швидку взаємодію користувача із системою без необхідності повного перезавантаження сторінок.

Однією з ключових переваг React є компонентний принцип побудови інтерфейсу. Користувацький інтерфейс системи формується з окремих незалежних компонентів, кожен із яких відповідає за реалізацію певної функціональної частини. Це спрощує супровід програмного коду та дозволяє повторно використовувати окремі елементи інтерфейсу в різних модулях застосунку.

Структуру клієнтської частини організовано таким чином, щоб забезпечити зручну навігацію між сторінками та швидкий доступ до основних можливостей системи. Функціональність користувацького рівня реалізовано через набір взаємопов'язаних сторінок і компонентів.

Головна сторінка призначена для ознайомлення користувачів із асортиментом продукції, популярними товарами та основними категоріями інтернет-магазину. На рисунку 3.1 представлено головну сторінку інтернет-магазину електроніки.

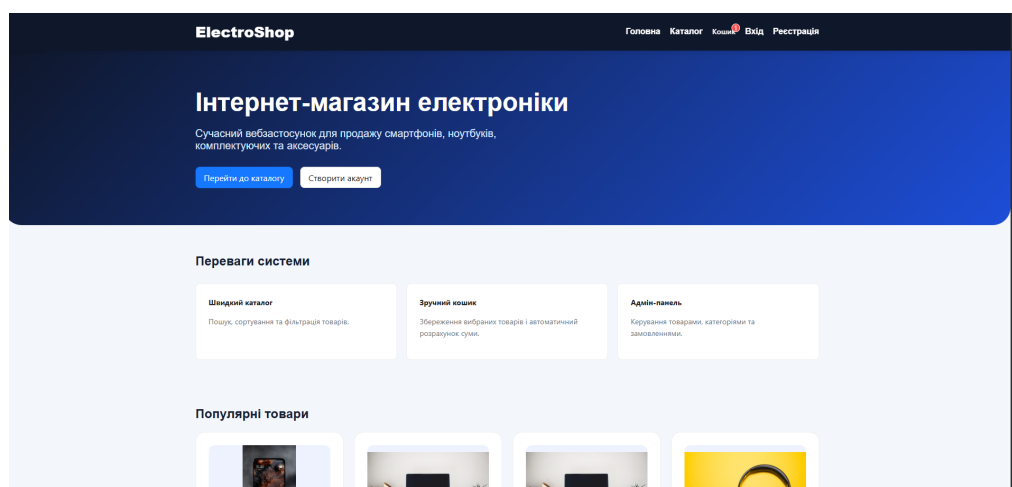


Рисунок 3.1 – Головна сторінка інтернет-магазину електроніки

Каталог товарів забезпечує перегляд усього асортименту продукції з можливістю виконання пошуку, сортування та фільтрації за різними критеріями. На рисунку 3.2 представлено каталог товарів із можливістю пошуку та фільтрації продукції.

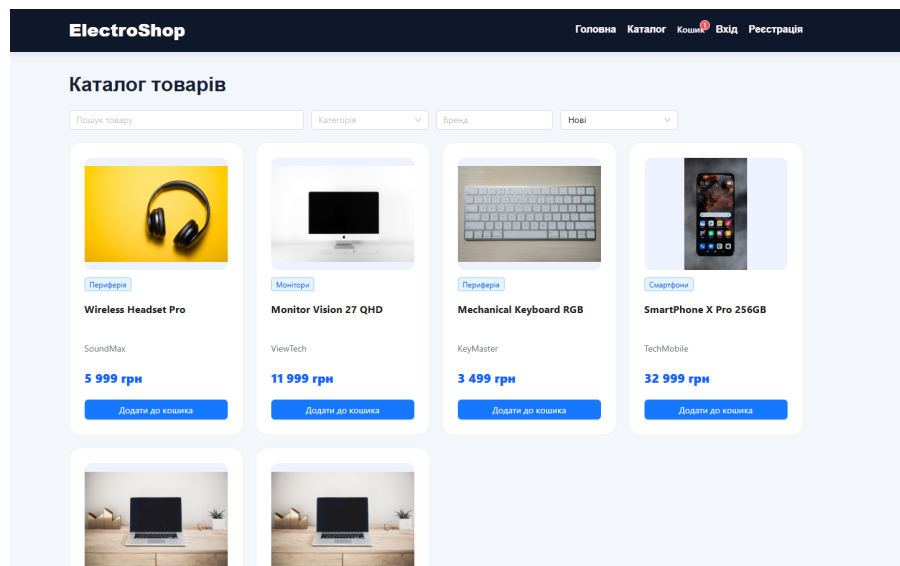


Рисунок 3.2 – Каталог товарів

Для детального ознайомлення з конкретною продукцією передбачено окрему сторінку товару, на якій відображаються технічні характеристики, опис, зображення, ціна та інформація про наявність на складі. На рисунку 3.3 показано сторінку детального перегляду товару, де відображаються його характеристики та ціна.

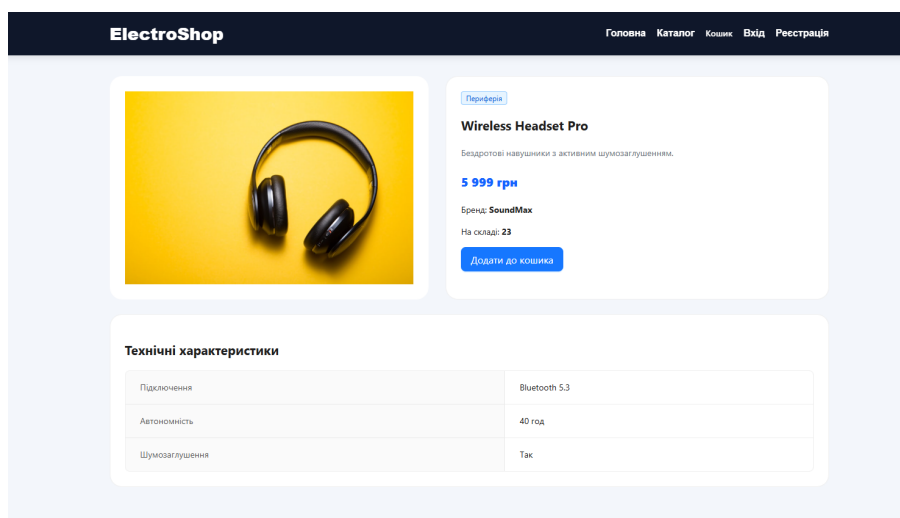
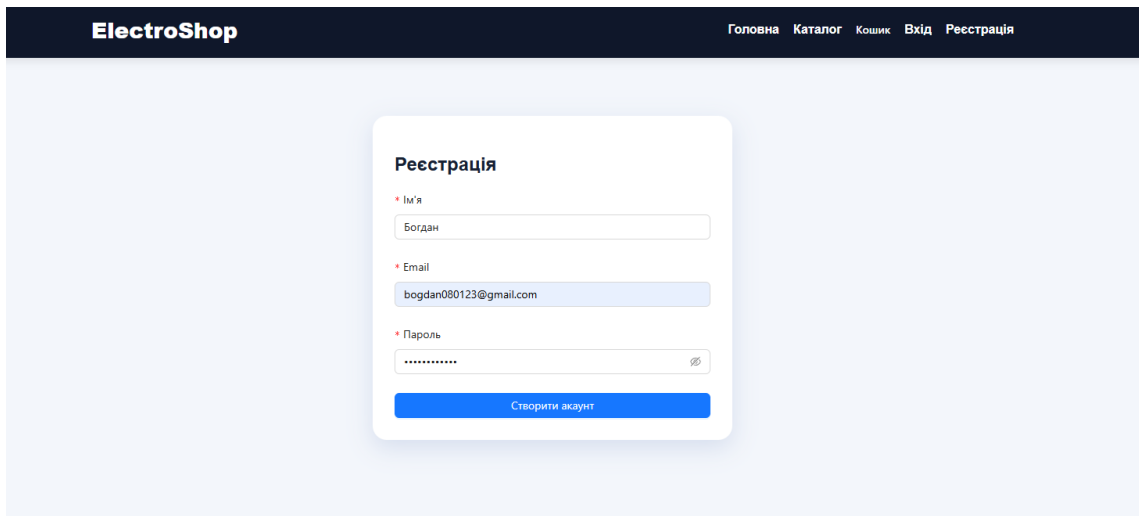


Рисунок 3.3 – Сторінка перегляду товару

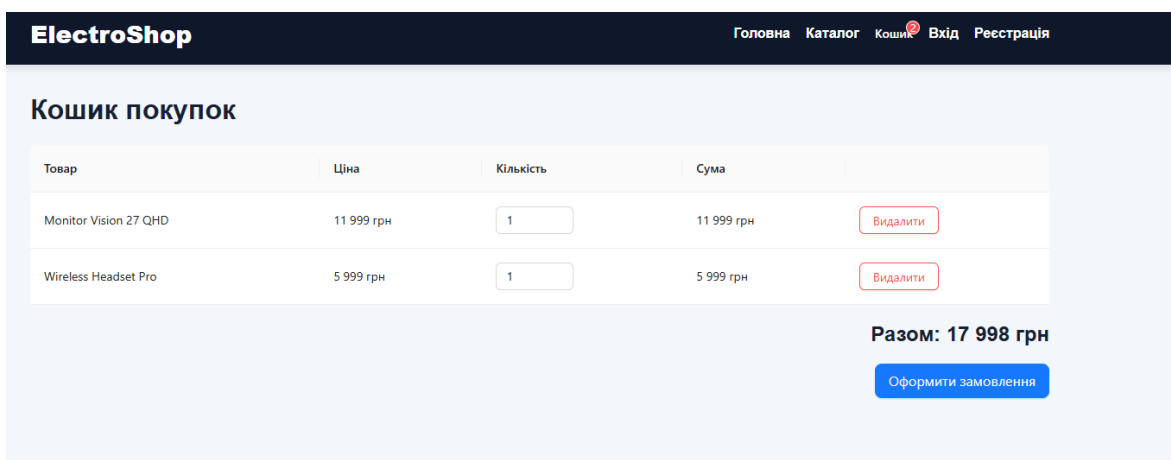
Для взаємодії із системою користувачеві доступні сторінки реєстрації та авторизації, які забезпечують створення облікового запису та вхід до системи. На рисунку 3.4 представлено форму реєстрації нового користувача.



The screenshot shows the registration page of the ElectroShop website. The header includes the site name 'ElectroShop' and navigation links: 'Головна', 'Каталог', 'Кошик', 'Вхід', and 'Реєстрація'. The main content area features a white registration form with the title 'Реєстрація'. The form contains three input fields: 'Ім'я' (Name) with the value 'Богдан', 'Email' with the value 'bogdan080123@gmail.com', and 'Пароль' (Password) which is masked with dots. A blue button labeled 'Створити акаунт' (Create account) is positioned at the bottom of the form.

Рисунок 3.4 – Сторінка авторизації користувача

Важливою складовою інтерфейсу є кошик покупок, який дає змогу зберігати вибрані товари, змінювати їх кількість і переглядати загальну вартість замовлення. Після формування кошика користувач переходить до сторінки оформлення замовлення, де вказує контактні дані та адресу доставки. На рисунку 3.5 представлено кошик покупок із переліком обраних товарів. На рисунку 3.6 показано процес оформлення замовлення.



The screenshot displays the shopping cart page on the ElectroShop website. The header is identical to the previous page, with the 'Кошик' (Cart) link highlighted. The main section is titled 'Кошик покупок' and contains a table with the following items:

Товар	Ціна	Кількість	Сума	
Monitor Vision 27 QHD	11 999 грн	<input type="text" value="1"/>	11 999 грн	<a href="#">Видалити</a>
Wireless Headset Pro	5 999 грн	<input type="text" value="1"/>	5 999 грн	<a href="#">Видалити</a>

Below the table, the total amount is displayed as 'Разом: 17 998 грн' and a blue button labeled 'Оформити замовлення' (Place order) is visible.

Рисунок 3.5 – Кошик покупок

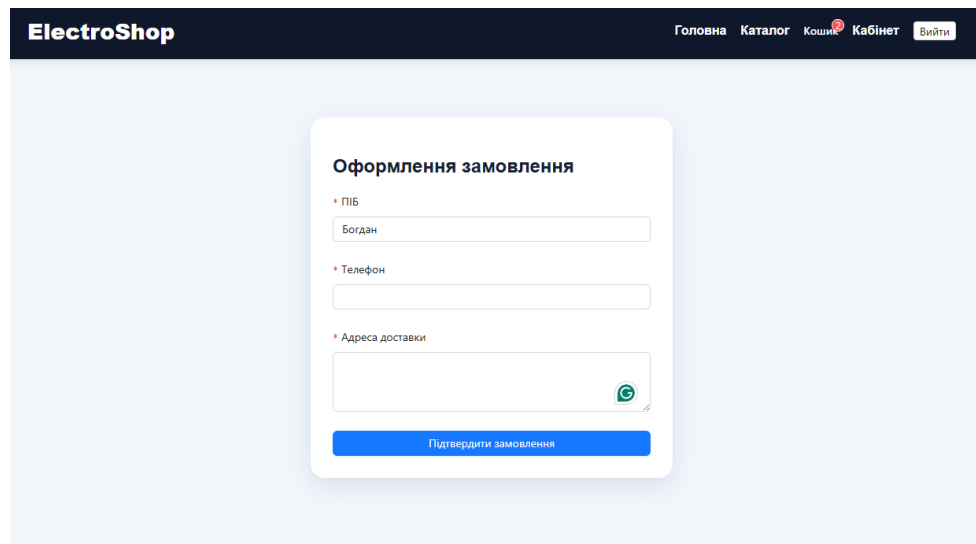


Рисунок 3.6 – Сторінка оформлення замовлення

Для зареєстрованих користувачів реалізовано особистий кабінет, у межах якого доступна інформація про профіль та історію оформлених замовлень. На рисунку 3. показано особистий кабінет користувача.

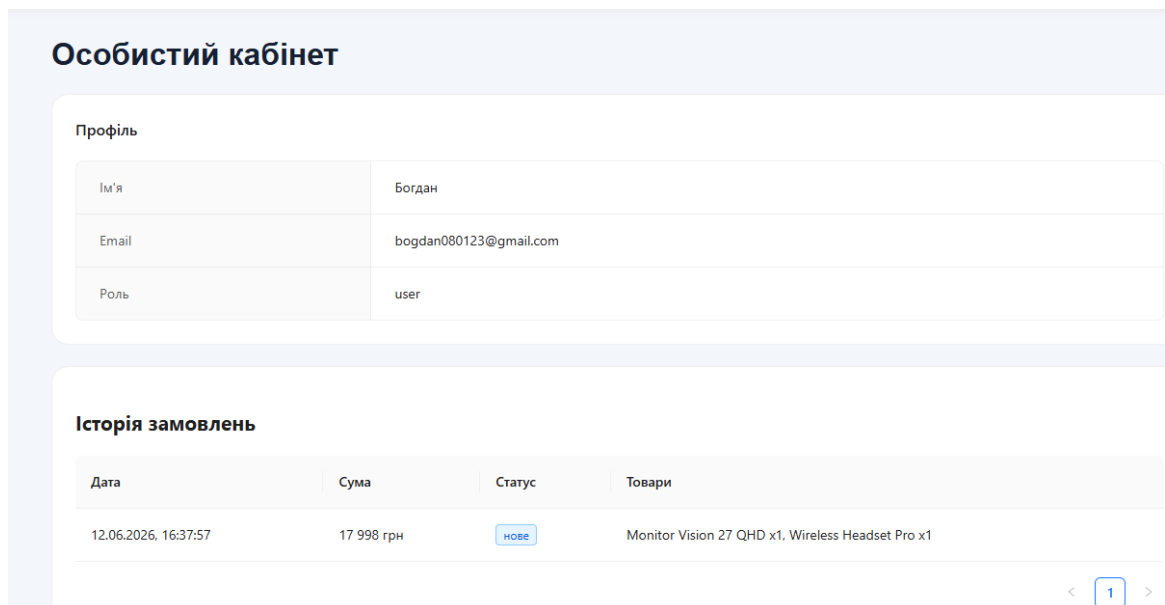


Рисунок 3.7 – Особистий кабінет користувача

Окрім цього, у системі передбачено адміністративну панель, яка надає можливість керувати товарами, категоріями та замовленнями, а також контролювати функціонування інтернет-магазину в цілому. На рисунку 3.8 представлено адміністративну панель системи.

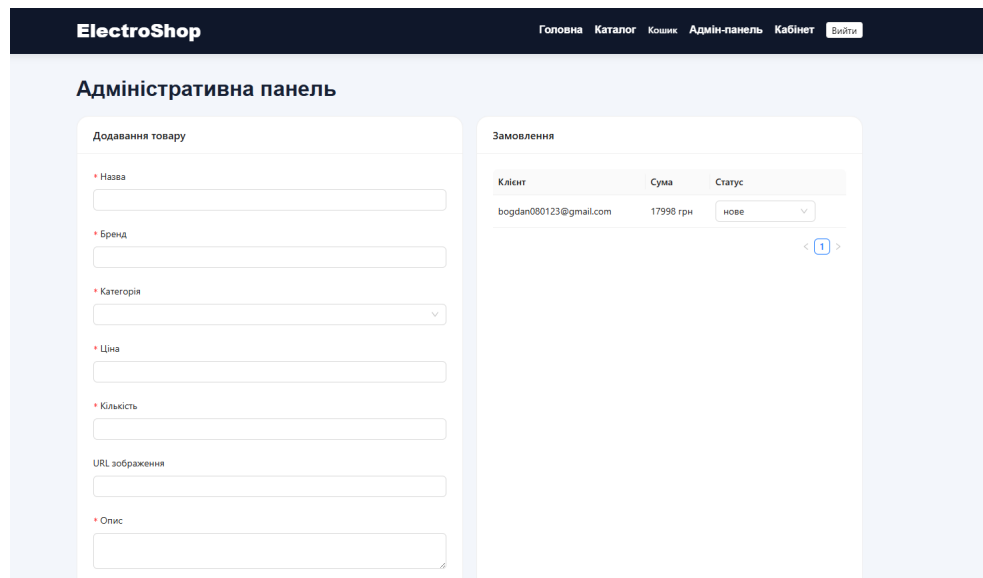


Рисунок 3.8 – Адміністративна панель

Для реалізації навігації між сторінками застосовується бібліотека React Router. Вона забезпечує маршрутизацію без повторного завантаження вебзастосунку. Кожна сторінка має власний маршрут, що формує логічну структуру інтерфейсу та покращує зручність взаємодії користувача із системою.

Для керування станом клієнтського застосунку використовується Redux Toolkit. Цей інструмент забезпечує централізоване зберігання інформації про користувача, товари, кошик та результати запитів до серверної частини. Використання Redux Toolkit дає змогу уникнути надмірної передачі даних між компонентами та покращує організацію програмного коду.

Процес авторизації реалізовано через окремі форми реєстрації та входу до системи. Після успішної автентифікації користувач отримує доступ до особистого кабінету, де може переглядати відомості про власний профіль і історію оформлених замовлень.

Для обміну даними із серверною частиною використовуються HTTP-запити. Отримана від сервера інформація обробляється та відображається в інтерфейсі користувача. Передавання даних здійснюється у форматі JSON, що забезпечує ефективну взаємодію між клієнтським і серверним рівнями системи.

Окрема увага під час розроблення приділялася адаптивності інтерфейсу. Усі сторінки коректно відображаються на персональних комп'ютерах, планшетах і

мобільних пристроях. Для реалізації адаптивного дизайну застосовано сучасні засоби CSS, які забезпечують автоматичне налаштування розташування елементів залежно від розміру екрана.

Отже, розроблена клієнтська частина забезпечує зручну взаємодію користувача з інтернет-магазином, надає можливість переглядати каталог товарів, здійснювати пошук продукції, оформлювати замовлення та керувати власним обліковим записом. Реалізований інтерфейс формує основу для ефективного функціонування всієї інформаційної системи.

### 3.2 Реалізація серверної частини вебзастосунок

Серверну частину інтернет-магазину електроніки реалізовано з використанням платформи Node.js та вебфреймворку Express.js. Застосування цих технологій дає змогу створити масштабований серверний застосунок, який забезпечує обробку запитів користувачів, взаємодію з базою даних і реалізацію бізнес-логіки системи.

Основне призначення серверної частини полягає в організації обміну даними між клієнтським застосунком і базою даних. З цією метою реалізовано REST API, через яке виконуються операції отримання, створення, оновлення та видалення даних.

#### 3.2.1 Створення REST API

Для забезпечення взаємодії між клієнтською та серверною частинами застосовано архітектурний стиль REST. Сервер надає набір API-маршрутів, які забезпечують доступ до функціональних можливостей системи.

Основні API-маршрути поділяються на такі категорії:

- маршрути авторизації та реєстрації користувачів;
- маршрути роботи з товарами;
- маршрути роботи з категоріями;
- маршрути оформлення замовлень;
- адміністративні маршрути.

Після надходження HTTP-запиту сервер здійснює перевірку вхідних даних, обробляє запит і повертає результат у форматі JSON. Такий підхід забезпечує

зручну інтеграцію з клієнтською частиною вебзастосунку та створює передумови для подальшого масштабування системи.

Для отримання списку товарів використовується GET-запит, який повертає інформацію про доступні позиції. Для додавання нового товару адміністратор застосовує POST-запит. Оновлення інформації виконується за допомогою PUT-запиту, а видалення записів реалізується через DELETE-запити.

### **3.2.2 Реалізація авторизації користувачів**

Однією з ключових складових системи є механізм авторизації та автентифікації користувачів. Для забезпечення безпечного доступу до ресурсів використовується технологія JWT (JSON Web Token).

Під час реєстрації користувач вводить персональні дані, електронну адресу та пароль. Перед збереженням у базі даних пароль проходить процедуру хешування, що підвищує рівень захисту облікових записів від несанкціонованого доступу.

Після успішної авторизації сервер генерує JWT-токен і передає його клієнтському застосунку. Надалі токен використовується для підтвердження особи користувача під час виконання захищених операцій.

У системі передбачено два основні типи користувачів:

- звичайний користувач;
- адміністратор.

Залежно від ролі користувача система надає відповідний рівень доступу до функціональних можливостей вебзастосунку. На рисунку 3.9 наведено схему процесу авторизації користувача.

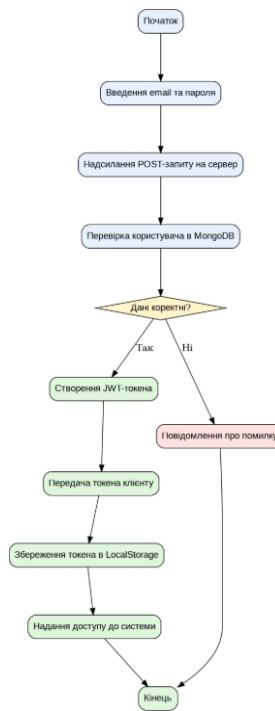


Рисунок 3.9 – Схема процесу авторизації користувача

### 3.2.3 Організація роботи з базою даних MongoDB

Для зберігання інформації використовується документоорієнтована база даних MongoDB. Вибір цієї системи керування базами даних зумовлений її гнучкою структурою та високою продуктивністю під час роботи з вебзастосунками.

Для взаємодії з базою даних застосовується бібліотека Mongoose, яка забезпечує створення моделей даних і виконання CRUD-операцій.

У системі реалізовано кілька основних колекцій:

- Users;
- Products;
- Categories;
- Orders.

Колекція Users містить відомості про зареєстрованих користувачів системи. Для кожного користувача зберігаються ім'я, електронна адреса, пароль і роль.

Колекція Products містить інформацію про товари, представлені в каталозі інтернет-магазину. Для кожного товару зберігаються назва, опис, ціна, бренд,

категорія, зображення та технічні характеристики. Колекція Categories використовується для групування товарів за категоріями та спрощення навігації каталогом. Колекція Orders містить інформацію про оформлені замовлення, зокрема список товарів, загальну суму, дату створення та статус замовлення. На рисунку 3.10 наведено структуру бази даних системи.

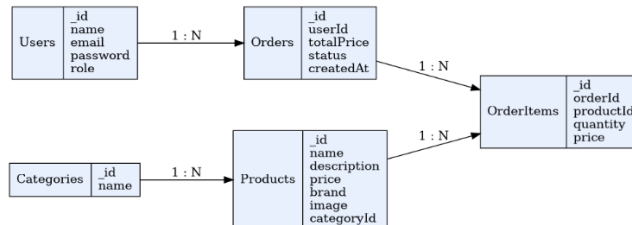


Рисунок 3.10 – Структура бази даних MongoDB

### 3.2.4 Реалізація обробки замовлень

Процес оформлення замовлення розпочинається з додавання товарів до кошика. Інформація про вибрані товари зберігається у стані клієнтського застосунку та передається на сервер після підтвердження замовлення користувачем.

Сервер отримує список товарів, перевіряє коректність вхідних даних і формує новий запис у колекції Orders. Після успішного створення замовлення користувач отримує повідомлення про успішне завершення операції.

Для кожного замовлення зберігається поточний статус, який може змінюватися адміністратором. У системі передбачено такі стани замовлення:

- нове;
- обробляється;
- відправлено;
- виконано;
- скасовано.

Такий підхід дає змогу контролювати процес виконання замовлень і забезпечує користувачам доступ до актуальної інформації про стан покупки.

### **3.2.5 Адміністративний функціонал**

Для керування роботою інтернет-магазину реалізовано адміністративну панель. Доступ до неї мають лише користувачі з роллю адміністратора.

Адміністратор може виконувати такі операції:

- додавання нових товарів;
- редагування інформації про товари;
- видалення товарів із каталогу;
- перегляд замовлень користувачів;
- зміна статусів замовлень;
- керування категоріями товарів.

Для захисту адміністративного функціоналу реалізовано перевірку прав доступу на стороні сервера. Перед виконанням адміністративних операцій система перевіряє роль користувача та наявність дійсного JWT-токена.

Отже, серверна частина забезпечує стабільне функціонування інтернет-магазину електроніки, реалізує взаємодію з базою даних, обробку замовлень, авторизацію користувачів і адміністративне керування системою.

### **Висновки до розділу 3**

У третьому розділі розглянуто процес реалізації програмного забезпечення інтернет-магазину електроніки. У ході виконання роботи було створено клієнтську та серверну частини вебзастосунку, які забезпечують повноцінне функціонування системи та взаємодію між користувачами і базою даних.

Під час реалізації клієнтської частини використано бібліотеку React, що дало змогу побудувати сучасний односторінковий вебзастосунок із компонентною архітектурою. Для організації маршрутизації застосовано React Router, а для централізованого керування станом — Redux Toolkit. Реалізовано основні сторінки системи, зокрема головну сторінку, каталог товарів, сторінку перегляду товару, кошик покупок, форми реєстрації та авторизації користувачів, сторінку оформлення замовлення, а також адміністративну панель.

Серверну частину системи розроблено з використанням Node.js та Express.js. Для зберігання інформації застосовано базу даних MongoDB, яка забезпечує гнучке зберігання даних про користувачів, товари, категорії та замовлення. Взаємодію між клієнтською та серверною частинами реалізовано за допомогою REST API та обміну даними у форматі JSON.

Для забезпечення безпеки реалізовано механізм автентифікації та авторизації користувачів на основі JWT-токенів. Це дає змогу обмежувати доступ до захищених ресурсів системи та забезпечувати розмежування прав між звичайними користувачами й адміністраторами.

У результаті виконання розділу було створено повноцінний програмний комплекс інтернет-магазину електроніки, який забезпечує перегляд каталогу товарів, оформлення замовлень, керування кошиком покупок, авторизацію користувачів та адміністрування системи. Отримані результати підтверджують доцільність використання сучасних вебтехнологій для розробки ефективних інформаційних систем електронної комерції.

## 4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Планування процесу тестування

Тестування є одним із найважливіших етапів розробки програмного забезпечення, оскільки дає змогу перевірити коректність функціонування системи, виявити можливі помилки та встановити відповідність реалізованого функціоналу поставленим вимогам. Для інтернет-магазину електроніки процес тестування спрямовано на перевірку основних функціональних можливостей системи, стабільності її роботи та зручності використання.

Метою тестування є підтвердження працездатності вебзастосунку, перевірка правильності обробки даних користувачів, коректності роботи механізму авторизації, функціонування кошика покупок і процесу оформлення замовлень.

Під час тестування особливу увагу приділено таким компонентам системи:

- модулю реєстрації користувачів;
- модулю авторизації користувачів;
- каталогу товарів;
- пошуку та фільтрації продукції;
- сторінці перегляду товару;
- кошику покупок;
- модулю оформлення замовлення;
- адміністративній панелі;
- взаємодії з базою даних MongoDB;
- REST API серверної частини.

Для перевірки працездатності системи використано функціональне тестування типу «чорна скринька» (Black Box Testing). Такий підхід дає змогу оцінити роботу програмного забезпечення без аналізу внутрішньої структури програмного коду.

У межах роботи було сформовано набір тестових сценаріїв, які охоплюють основні варіанти використання системи. Для кожного тестового сценарію визначено вхідні дані, очікуваний і фактичний результати виконання.

План тестування передбачає перевірку таких функцій:

1. Реєстрація нового користувача.
2. Авторизація користувача.
3. Перегляд каталогу товарів.
4. Пошук товарів за назвою.
5. Фільтрація товарів за категоріями.
6. Перегляд детальної інформації про товар.
7. Додавання товару до кошика.
8. Видалення товару з кошика.
9. Оформлення замовлення.
10. Перегляд адміністративної панелі.
11. Додавання нового товару адміністратором.
12. Редагування інформації про товар.
13. Видалення товару з каталогу.

Окрім функціонального тестування, проводиться також перевірка коректності роботи інтерфейсу користувача. Особливу увагу приділено адаптивності вебзастосунку та коректному відображенню сторінок на різних типах пристроїв.

Тестування виконувалося в середовищі операційної системи Windows 11 із використанням браузера Google Chrome. Серверна частина запускалася на платформі Node.js, а база даних функціонувала під керуванням MongoDB.

Отже, сформований план тестування дає змогу комплексно перевірити функціональність інтернет-магазину електроніки та підтвердити відповідність реалізованої системи встановленим вимогам.

## **4.2 Розробка тестових сценаріїв**

Для перевірки працездатності інтернет-магазину електроніки було розроблено набір тестових сценаріїв, що охоплюють основні функціональні можливості системи. Кожен тестовий сценарій містить опис дій користувача, вхідні дані, очікуваний результат і фактичний результат виконання.

Тестування проводилося методом функціонального тестування типу «чорна скринька», що дає змогу оцінити правильність роботи програмного забезпечення без аналізу його внутрішньої структури.

Таблиця 4.1 – Тестування функцій реєстрації та авторизації

№	Тестовий сценарій	Вхідні дані	Очікуваний результат	Результат
1	Реєстрація нового користувача	Коректні дані	Створення облікового запису	Успішно
2	Реєстрація з порожніми полями	Порожні значення	Відображення повідомлення про помилку	Успішно
3	Реєстрація з некоректною електронною адресою	test@test	Відображення повідомлення про помилку валідації	Успішно
4	Авторизація користувача	Правильний логін і пароль	Успішний вхід до системи	Успішно
5	Авторизація з неправильним паролем	Невірний пароль	Відображення повідомлення про помилку	Успішно

Таблиця 4.2 – Тестування каталогу товарів

<b>№</b>	<b>Тестовий сценарій</b>	<b>Вхідні дані</b>	<b>Очікуваний результат</b>	<b>Результат</b>
1	Відкриття каталогу товарів	Відсутні	Відображення списку товарів	Успішно
2	Пошук товару	Назва товару	Відображення результатів пошуку	Успішно
3	Пошук неіснуючого товару	Довільний текст	Відображення повідомлення про відсутність результатів	Успішно
4	Фільтрація товарів	Категорія товарів	Відображення товарів обраної категорії	Успішно
5	Перехід на сторінку товару	Натискання кнопки «Деталі»	Відображення сторінки товару	Успішно

Таблиця 4.3 – Тестування кошика покупок

№	Тестовий сценарій	Вхідні дані	Очікуваний результат	Результат
1	Додавання товару до кошика	Натискання кнопки «Додати до кошика»	Додавання товару до кошика	Успішно
2	Видалення товару з кошика	Натискання кнопки видалення	Видалення товару з кошика	Успішно
3	Зміна кількості товару	Нова кількість	Перерахунок загальної вартості	Успішно
4	Очищення кошика	Видалення всіх товарів	Кошик стає порожнім	Успішно

Таблиця 4.4 – Тестування оформлення замовлення

№	Тестовий сценарій	Вхідні дані	Очікуваний результат	Результат
1	Створення замовлення	Коректні дані	Успішне створення замовлення	Успішно
2	Створення замовлення без авторизації	Неавторизований користувач	Відмова в доступі	Успішно
3	Введення порожньої адреси	Порожнє поле	Відображення повідомлення про помилку	Успішно
4	Введення некоректного номера телефону	Невірний формат	Відображення повідомлення про помилку	Успішно

Таблиця 4.5 – Тестування адміністративної панелі

№	Тестовий сценарій	Вхідні дані	Очікуваний результат	Результат
1	Вхід адміністратора	Дані адміністратора	Відображення адміністративної панелі	Успішно
2	Додавання нового товару	Коректні дані	Успішне створення товару	Успішно
3	Редагування товару	Нові дані	Оновлення інформації про товар	Успішно
4	Видалення товару	Існуючий товар	Успішне видалення товару	Успішно
5	Перегляд замовлень	Відсутні	Відображення списку замовлень	Успішно

Розроблені тестові сценарії дали змогу перевірити основні функціональні можливості системи та підтвердити коректність роботи інтернет-магазину електроніки. Усі виконані тести завершилися успішно, що свідчить про відповідність програмного забезпечення поставленим функціональним вимогам.

### 4.3 Результати тестування системи

Після завершення розробки інтернет-магазину електроніки було проведено комплексне тестування програмного забезпечення. Основною метою тестування стала перевірка працездатності всіх функціональних модулів системи, виявлення можливих помилок і підтвердження відповідності реалізованого функціоналу встановленим вимогам.

Тестування виконувалося в середовищі операційної системи Windows 11 з використанням веббраузера Google Chrome. Серверна частина функціонувала на платформі Node.js із застосуванням вебфреймворку Express.js, а зберігання даних забезпечувалося системою керування базами даних MongoDB.

У процесі тестування було перевірено роботу всіх основних компонентів системи:

- реєстрацію користувачів;
- авторизацію користувачів;
- перегляд каталогу товарів;
- пошук товарів;
- фільтрацію товарів;
- перегляд інформації про товар;
- роботу кошика покупок;
- оформлення замовлень;
- функціонування адміністративної панелі;
- взаємодію з базою даних;
- обмін даними між клієнтською та серверною частинами.

Результати тестування засвідчили, що система коректно обробляє запити користувачів і забезпечує виконання всіх передбачених функцій. Під час проведення тестування критичних помилок, які могли б призвести до втрати даних або унеможливити використання системи, виявлено не було. На рисунку 4.1 наведено результат роботи головної сторінки вебзастосунку.

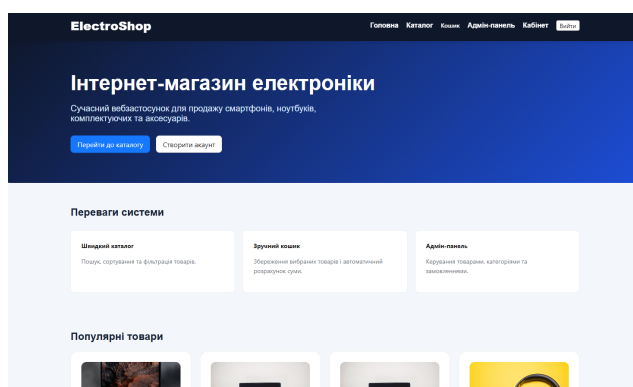


Рисунок 4.1 – Результат тестування головної сторінки

На рисунку 4.2 наведено результат роботи каталогу товарів.

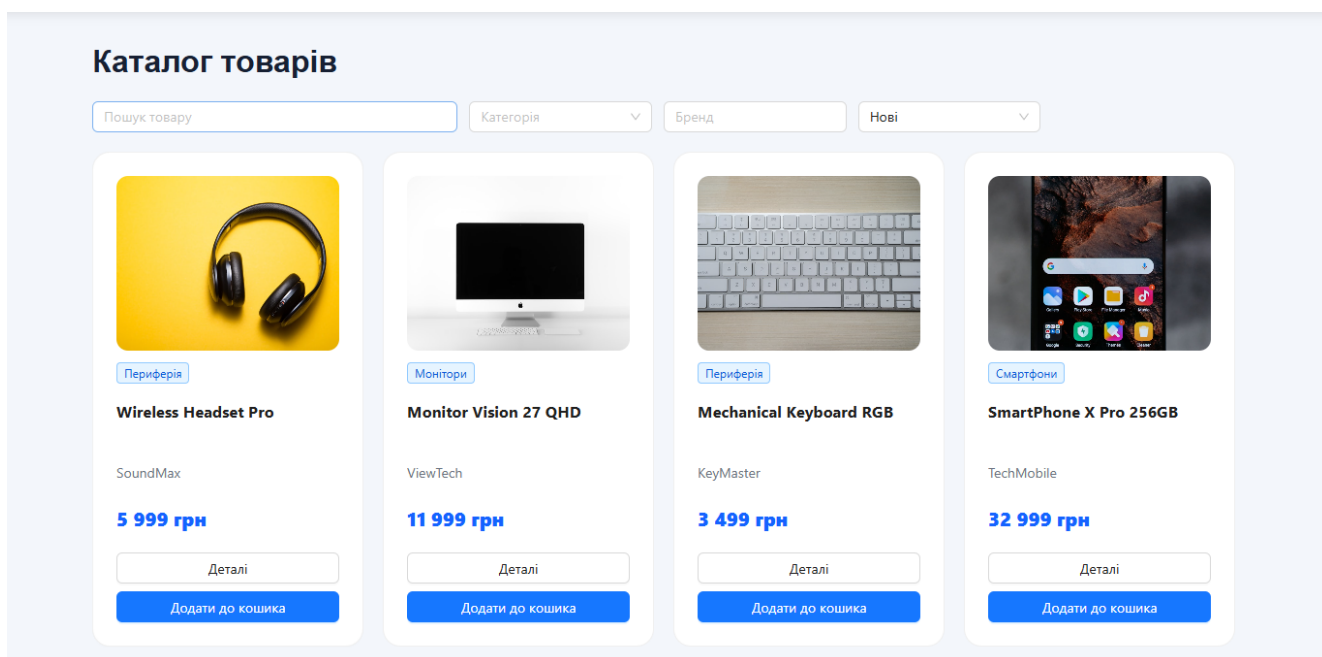


Рисунок 4.2 – Результат тестування каталогу товарів

На рисунку 4.3 наведено результат тестування сторінки перегляду товару.

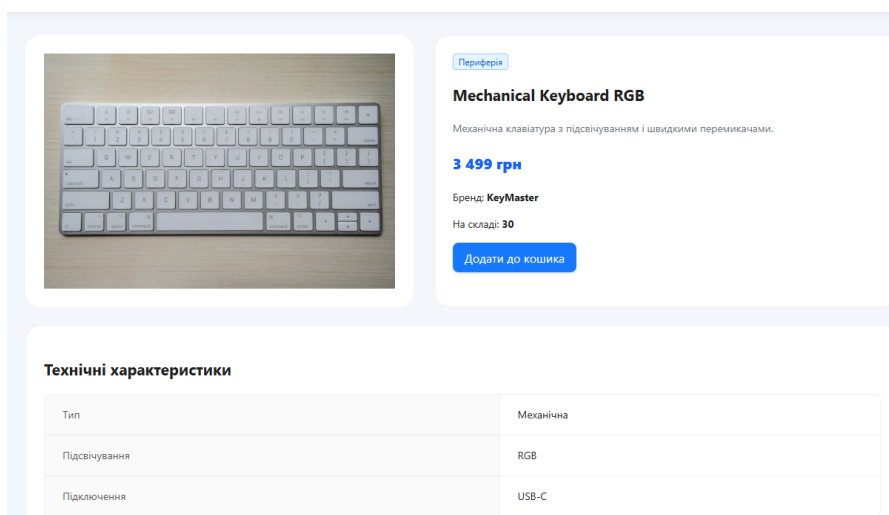


Рисунок 4.3 – Результат тестування сторінки товару

На рисунку 4.4 наведено результат тестування процесу авторизації користувача.

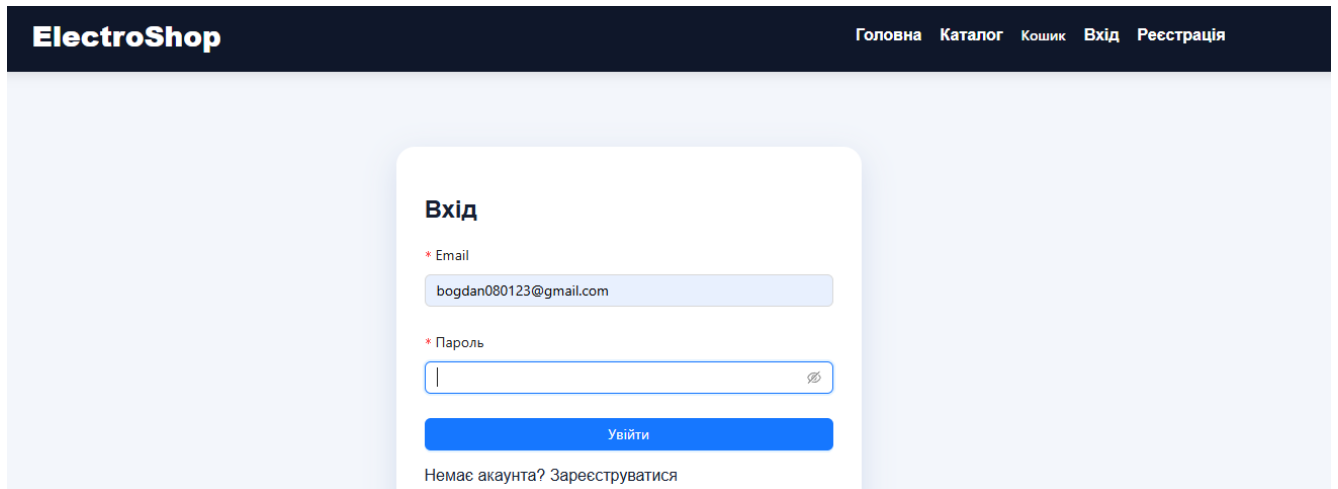


Рисунок 4.4 – Результат тестування авторизації

На рисунку 4.5 наведено результат тестування кошика покупок.

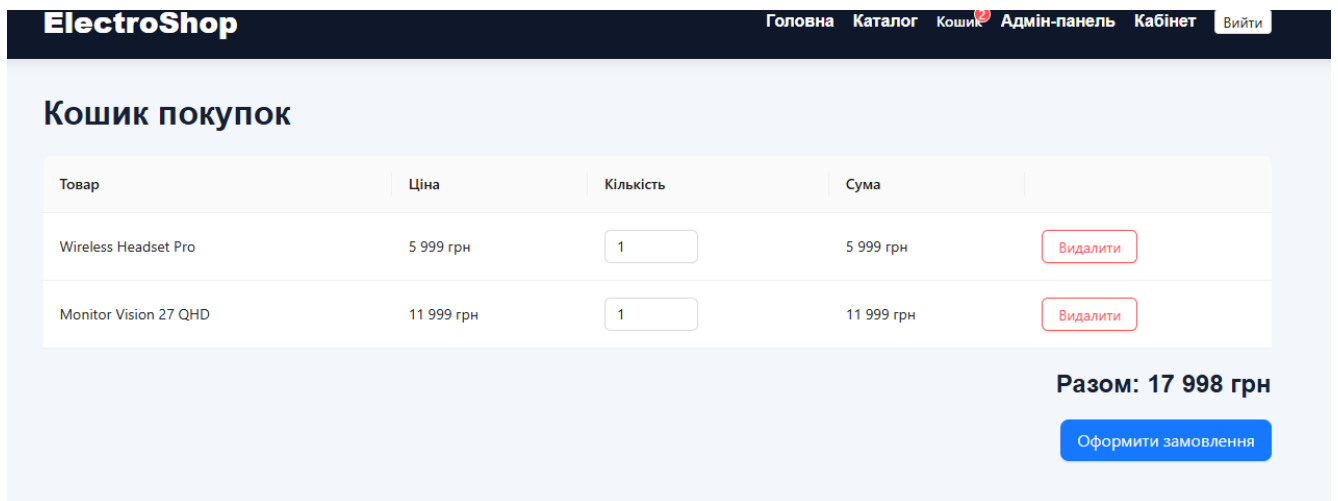


Рисунок 4.5 – Результат тестування кошика покупок

На рисунку 4.6 наведено результат тестування адміністративної панелі.

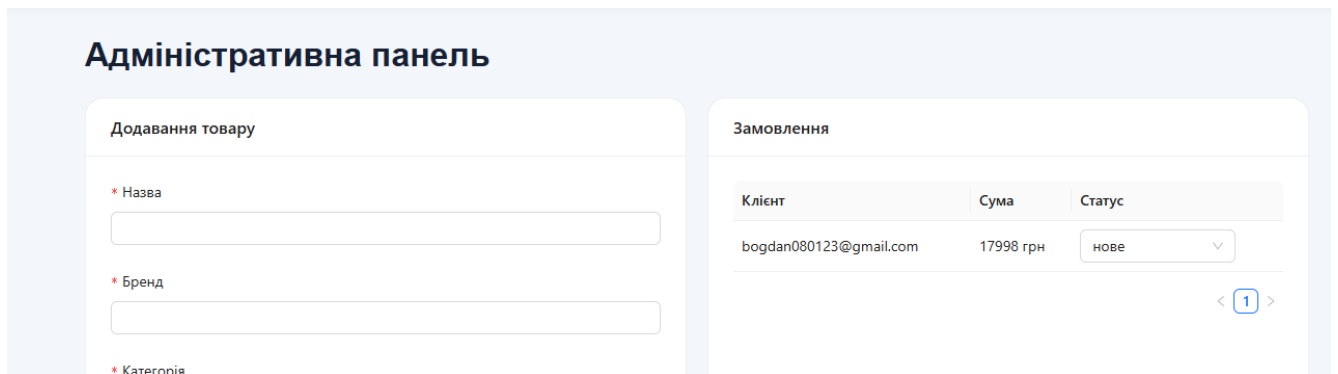


Рисунок 4.6 – Результат тестування адміністративної панелі

Під час перевірки функціоналу реєстрації та авторизації встановлено, що система коректно виконує створення нових облікових записів, перевіряє введені користувачем дані та забезпечує захищений доступ до особистого кабінету за допомогою JWT-токенів.

Перевірка каталогу товарів підтвердила правильність відображення інформації про продукцію, а також коректність роботи механізмів пошуку та фільтрації. Сторінка перегляду товару забезпечує відображення технічних характеристик, вартості та опису продукції.

Тестування кошика покупок показало, що система правильно виконує додавання товарів, зміну кількості позицій та автоматичний перерахунок загальної вартості замовлення.

У процесі перевірки оформлення замовлень було встановлено, що інформація про замовлення успішно передається на сервер і зберігається в базі даних MongoDB. Користувач при цьому отримує підтвердження про успішне створення замовлення.

Під час тестування адміністративної панелі перевірено можливість додавання нових товарів, редагування наявних записів, видалення продукції та перегляду замовлень користувачів. Усі операції виконувалися коректно та належним чином відображалися в базі даних.

Отже, результати тестування підтвердили працездатність розробленого інтернет-магазину електроніки та його готовність до експлуатації. Реалізована система забезпечує виконання всіх поставлених завдань, відповідає

функціональним вимогам і може бути використана як основа для подальшого розвитку та вдосконалення програмного продукту.

#### **Висновки до розділу 4**

У четвертому розділі проведено тестування розробленого інтернет-магазину електроніки та здійснено аналіз отриманих результатів. Для перевірки працездатності системи було сформовано набір тестових сценаріїв, які охоплюють основні функціональні можливості вебзастосунку.

У процесі тестування перевірено роботу модулів реєстрації та авторизації користувачів, каталогу товарів, сторінки перегляду товару, кошика покупок, оформлення замовлень і адміністративної панелі. Окрему увагу приділено перевірці взаємодії клієнтської частини із сервером і базою даних MongoDB.

Результати тестування засвідчили, що система коректно обробляє запити користувачів, забезпечує надійне зберігання даних і виконує всі функції відповідно до встановлених вимог. Усі розроблені тестові сценарії було успішно виконано, що підтверджує працездатність програмного забезпечення.

Проведене тестування дало змогу переконатися в правильності реалізації механізмів авторизації, роботи кошика покупок, обробки замовлень і функціонування адміністративного інтерфейсу. Помилки, виявлені під час розробки, було усунено, що сприяло підвищенню стабільності та надійності роботи системи.

Отже, результати тестування підтвердили відповідність розробленого інтернет-магазину електроніки поставленим вимогам і готовність програмного забезпечення до практичного використання.

## ВИСНОВКИ

У кваліфікаційній роботі розроблено вебзастосунок «Інтернет-магазин електроніки», який забезпечує автоматизацію процесів перегляду товарів, оформлення замовлень та адміністрування каталогу продукції.

У ході виконання роботи проведено аналіз предметної області електронної комерції та сучасних інтернет-магазинів електроніки. Результати аналізу дали змогу визначити основні функціональні вимоги до системи та сформувавши перелік завдань, необхідних для створення програмного продукту.

Під час проєктування було розроблено модель предметної області, визначено основні сутності системи та встановлено зв'язки між ними. Також обґрунтовано вибір технологій розробки, зокрема React для реалізації клієнтської частини, Node.js та Express.js — для серверної частини, MongoDB — для зберігання даних, а JWT — для організації механізму авторизації користувачів.

У результаті виконання практичної частини створено повноцінний вебзастосунок інтернет-магазину електроніки. Реалізовано каталог товарів із можливістю пошуку та перегляду продукції, сторінки реєстрації та авторизації користувачів, кошик покупок, механізм оформлення замовлень, а також адміністративну панель для керування товарами й замовленнями.

Для забезпечення зручності використання застосовано сучасний підхід до побудови користувацького інтерфейсу на основі компонентної архітектури React. Це дало змогу створити адаптивний інтерфейс, який коректно функціонує на різних типах пристроїв і забезпечує комфортну взаємодію користувача із системою.

Серверна частина забезпечує взаємодію з базою даних MongoDB через REST API, обробку клієнтських запитів і виконання бізнес-логіки системи. Для підвищення рівня безпеки реалізовано механізм автентифікації та авторизації на основі JWT-токенів.

Проведене тестування підтвердило працездатність усіх функціональних модулів системи. Отримані результати засвідчили коректність роботи інтернет-магазину та відповідність реалізованого програмного забезпечення встановленим вимогам.

Отже, поставленої мети кваліфікаційної роботи досягнуто, а всі визначені завдання виконано в повному обсязі. Розроблений інтернет-магазин електроніки може бути використаний як основа для створення повноцінної комерційної платформи та має перспективи подальшого розвитку шляхом розширення функціональних можливостей і впровадження додаткових сервісів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Pressman R. S. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2019. 880 p.
2. Sommerville I. Software Engineering. 10th ed. Boston : Pearson, 2016. 816 p.
3. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1995. 395 p.
4. Martin R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Upper Saddle River : Prentice Hall, 2008. 464 p.
5. Fowler M. Patterns of Enterprise Application Architecture. Boston : Addison-Wesley, 2002. 560 p.
6. React Documentation. URL: <https://react.dev> (last excess: 26.05.2026).
7. Redux Toolkit Documentation. URL: <https://redux-toolkit.js.org> (last excess: 26.05.2026).
8. React Router Documentation. URL: <https://reactrouter.com> (last excess: 26.05.2026).
9. Node.js Documentation. URL: <https://nodejs.org> (last excess: 26.05.2026).
10. Express.js Documentation. URL: <https://expressjs.com> (last excess: 26.05.2026).
11. MongoDB Documentation. URL: <https://www.mongodb.com/docs> (last excess: 26.05.2026).
12. Mongoose Documentation. URL: <https://mongoosejs.com> (last excess: 26.05.2026).
13. JSON Web Token Introduction. URL: <https://jwt.io> (last excess: 26.05.2026).
14. Mozilla Developer Network (MDN) Web Docs. URL: <https://developer.mozilla.org> (last excess: 26.05.2026).
15. ECMAScript Language Specification. URL: <https://tc39.es> (last excess: 26.05.2026).

16. HTML Living Standard. URL: <https://html.spec.whatwg.org> (last excess: 26.05.2026).
17. CSS Specifications. URL: <https://www.w3.org/Style/CSS> (last excess: 26.05.2026).
18. Ant Design Documentation. URL: <https://ant.design> (last excess: 26.05.2026).
19. Styled Components Documentation. URL: <https://styled-components.com> (last excess: 26.05.2026).
20. Vite Documentation. URL: <https://vitejs.dev> (last excess: 26.05.2026).
21. MongoDB University. Introduction to MongoDB. URL: <https://learn.mongodb.com> (last excess: 26.05.2026).
22. Richardson L., Ruby S. RESTful Web Services. Sebastopol : O'Reilly Media, 2007. 448 p.
23. Fielding R. Architectural Styles and the Design of Network-based Software Architectures. Irvine : University of California, 2000.
24. Chacon S., Straub B. Pro Git. New York : Apress, 2014. 520 p.
25. Duckett J. JavaScript and JQuery: Interactive Front-End Web Development. Indianapolis : Wiley, 2014. 640 p.
26. Duckett J. HTML and CSS: Design and Build Websites. Indianapolis : Wiley, 2011. 490 p.
27. Flanagan D. JavaScript: The Definitive Guide. 7th ed. Sebastopol : O'Reilly Media, 2020. 704 p.
28. Freeman E., Robson E. Head First Design Patterns. Sebastopol : O'Reilly Media, 2020. 694 p.
29. Google Chrome Developer Tools Documentation. URL: <https://developer.chrome.com/docs/devtools> (last excess: 26.05.2026).
30. Visual Studio Code Documentation. URL: <https://code.visualstudio.com/docs> (last excess: 26.05.2026).
31. MongoDB Compass Documentation. URL: <https://www.mongodb.com/products/compass> (last excess: 26.05.2026).

32. Git Documentation. URL: <https://git-scm.com/doc> (last excess: 26.05.2026).
33. Docker Documentation. URL: <https://docs.docker.com> (last excess: 26.05.2026).
34. OpenAPI Specification. URL: <https://swagger.io/specification> (last excess: 26.05.2026).
35. Axios Documentation. URL: <https://axios-http.com> (last excess: 26.05.2026).

## ДОДАТОК А

### ЛІСТИНГИ ОСНОВНИХ ПРОГРАМНИХ МОДУЛІВ

#### Лістинг А.1 – Головний файл клієнтської частини (main.jsx)

```
ReactDOM.createRoot(document.getElementById('root')).render(  
  <Provider store={store}>  
    <BrowserRouter>  
      <App />  
    </BrowserRouter>  
  </Provider>  
);
```

#### Лістинг А.2 – Компонент відображення товару (ProductCard.jsx)

```
export default function ProductCard({ product }) {  
  return (  
    <Card hoverable>  
      <Link to={` /product/${product._id}` }>  
        <img src={product.image} alt={product.name} />  
      </Link>  
    </Card>  
  );  
}
```

#### Лістинг А.3 – Каталог товарів (Catalog.jsx)

```
useEffect(() => {  
  api.get('/products', {  
    params: { search, sort }  
  }).then(r => setProducts(r.data));  
}, [search, sort]);
```

#### Лістинг А.4 – Сторінка перегляду товару (ProductDetails.jsx)

```
const { id } = useParams();  
  
useEffect(() => {
```

```
api.get(`/products/${id}`)  
  .then(r => setProduct(r.data));  
}, [id]);
```

#### **Лістинг А.5 – Серверна частина (server.js)**

```
app.use('/api/auth', authRoutes);  
app.use('/api/products', productRoutes);  
app.use('/api/orders', orderRoutes);
```

#### **Лістинг А.6 – Модель користувача (User.js)**

```
const UserSchema = new mongoose.Schema({  
  name: String,  
  email: String,  
  password: String,  
  role: String  
});
```

#### **Лістинг А.7 – Модель товару (Product.js)**

```
const ProductSchema = new mongoose.Schema({  
  name: String,  
  price: Number,  
  brand: String,  
  image: String  
});
```

#### **Лістинг А.8 – Авторизація користувача**

```
const token = jwt.sign(  
  { id: user._id, role: user.role },  
  process.env.JWT_SECRET  
);
```

## ДОДАТОК Б

### ІНСТРУКЦІЯ КОРИСТУВАЧА ВЕБЗАСТОСУНКУ

У додатку наведено коротку інструкцію користувача для роботи з вебзастосунком «Інтернет-магазин електроніки».

#### **Б.1 Запуск програмного забезпечення**

Для запуску інтернет-магазину необхідно встановити Node.js, MongoDB та завантажити проєкт на локальний комп'ютер.

Після цього потрібно відкрити термінал у кореневій папці проєкту та виконати команди:

```
npm run install:all
```

```
npm run seed
```

```
npm run dev
```

Після успішного запуску клієнтська частина буде доступна за адресою:

```
http://localhost:5173
```

Серверна частина працює за адресою:

```
http://localhost:5000
```

#### **Б.2 Робота з головною сторінкою**

Після відкриття вебзастосунку користувач потрапляє на головну сторінку інтернет-магазину. На ній розміщено загальну інформацію про магазин, навігаційне меню та посилання для переходу до каталогу товарів.

#### **Б.3 Перегляд каталогу товарів**

Для перегляду товарів користувач переходить до розділу «Каталог». У цьому розділі відображаються доступні товари із зазначенням назви, бренду, ціни та зображення.

Користувач може скористатися пошуком для швидкого знаходження потрібного товару. Також передбачено сортування товарів за новизною та ціною.

#### **Б.4 Перегляд детальної інформації про товар**

Для перегляду повної інформації про товар потрібно натиснути кнопку «Деталі». Після цього відкривається сторінка товару, на якій відображаються опис, технічні характеристики, ціна та зображення продукції.

## **Б.5 Додавання товару до кошика**

Щоб додати товар до кошика, користувач натискає кнопку «Додати до кошика». Після цього товар зберігається у кошику покупок. У кошику користувач може переглянути список вибраних товарів і загальну вартість замовлення.

## **Б.6 Реєстрація користувача**

Для створення облікового запису користувач переходить на сторінку реєстрації та вводить необхідні дані: ім'я, адресу електронної пошти та пароль. Після успішної реєстрації користувач може виконати вхід до системи.

## **Б.7 Авторизація користувача**

Для входу до системи користувач переходить на сторінку авторизації та вводить електронну пошту й пароль. Після успішного входу користувач отримує доступ до оформлення замовлень та особистого кабінету.

## **Б.8 Оформлення замовлення**

Після додавання товарів до кошика користувач переходить до сторінки оформлення замовлення. На цій сторінці потрібно ввести контактні дані та адресу доставки. Після підтвердження замовлення інформація передається на сервер і зберігається в базі даних.

## **Б.9 Робота адміністратора**

Для входу до адміністративної панелі потрібно авторизуватися під обліковим записом адміністратора:

Email: admin@shop.com

Пароль: admin123

Після входу адміністратор отримує доступ до адміністративної панелі, де може додавати нові товари, редагувати інформацію про наявні товари, видаляти записи та переглядати замовлення користувачів.

## **Б.10 Завершення роботи**

Для завершення роботи користувач може вийти з облікового запису за допомогою відповідної кнопки в інтерфейсі. Після виходу доступ до захищених функцій системи буде обмежена