

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри )

## Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СТВОРЕННЯ ПРОГРАМНОЇ ВЕБСИСТЕМИ ЕЛЕКТРОННОЇ  
КОМЕРЦІЇ

DESIGN OF A SOFTWARE WEB SYSTEM  
FOR E-COMMERCE

Виконав(ла): студент(ка) 4 курсу, групи КНТ-222

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ГОРДІЄНКО А.В.

(ПРІЗВИЩЕ та ініціали)

Керівник ПОДКОВАЛІХІНА О.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**Національний університет «Запорізька політехніка»**  
 (повне найменування закладу вищої освіти)

Факультет КНТ  
 Кафедра програмних засобів  
 Ступінь вищої освіти бакалавр  
 Спеціальність 122 Комп'ютерні науки  
(код і найменування)  
 Освітня програма (спеціалізація) Комп'ютерні науки  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ПЗ, д.т.н, проф.  
Сергій СУББОТІН  
 “ ” 2026 року

**З А В Д А Н Н Я**

**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

ГОРДІЄНКА Артема Володимировича  
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення програмної вебсистеми електронної комерції.  
Design of a Software Web System for E-Commerce  
 керівник проєкту (роботи) к.ф.-м.н., доцент, ПОДКОВАЛІХІНА Олена Олександрівна,  
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)
- затверджені наказом закладу вищої освіти від “ 07 ” квітня 2026 року № 139
2. Строк подання студентом проєкту (роботи) 25 травня 2026 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання, технології розробки – Laravel, PostgreSQL.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області та постановка завдання. 2. Вибір та обґрунтування технологій розробки. 3. Проєктування та програмна реалізація вебсистеми. 4. Тестування вебсистеми та аналіз результатів.
5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)  
Слайди презентації

## 6. Консультанти розділів проєкту (роботи)

| Розділ              | ПРИЗВИЩЕ, ініціали та посада консультанта | Підпис, дата   |                           |
|---------------------|-------------------------------------------|----------------|---------------------------|
|                     |                                           | завдання видав | прийняв виконане завдання |
| 1-4 Основна частина | ПОДКОВАЛІХІНА О.О., доцент                |                |                           |
| Нормоконтроль       | БЄЛОВА А.В, асистент                      |                |                           |

7. Дата видачі завдання « 14 » квітня 2026 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                     | Строк виконання етапів проєкту ( роботи ) | Примітка     |
|-------|--------------------------------------------------------------|-------------------------------------------|--------------|
| 1     | Постановка завдання роботи.                                  | 1 тиждень                                 | Завдання, ТЗ |
| 2     | Аналіз предметної області та постановка завдання.            | 1 тиждень                                 | Розділ 1     |
| 3     | Вибір та обґрунтування технологій розробки.                  | 2 тиждень                                 | Розділ 2     |
| 4     | Проектування архітектури вебсистеми та структури бази даних. | 2 тиждень                                 | Розділ 3     |
| 5     | Програмна реалізація вебсистеми.                             | 3-4 тижні                                 | Розділ 3     |
| 6     | Розробка тестової документації та проведення тестування.     | 5 тиждень                                 | Розділ 4     |
| 7     | Оформлення пояснювальної записки та документів до неї.       | 6 тиждень                                 | Додатки      |
| 8     | Нормоконтроль та рецензування.                               | 7 тиждень                                 |              |
| 9     | Захист роботи.                                               | 8 тиждень                                 |              |

Студент(ка)

\_\_\_\_\_ Артем ГОРДІСНКО  
( підпис ) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

\_\_\_\_\_ Олена ПОДКОВАЛІХІНА  
( підпис ) (Ім'я ПРИЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 109 с., 2 табл., 41 рис., 3 дод., 41 джерело.

ІНТЕРНЕТ-МАГАЗИН, ВЕБСИСТЕМА, ЕЛЕКТРОННА КОМЕРЦІЯ, LARAVEL, POSTGRESQL, ТЕСТУВАННЯ ПЗ, АРХІТЕКТУРА MVC.

Об'єкт дослідження – процеси моделювання, розробки та забезпечення якості масштабованих програмних вебсистем у галузі електронної комерції.

Предмет дослідження – методи, алгоритми оптимізації обробки даних та програмні засоби розробки і забезпечення якості вебсистеми інтернет-магазину на основі фреймворку Laravel.

Мета роботи – оптимізація часових показників обробки даних та підвищення надійності вебсистеми на основі застосування сучасних методів розробки та методик комплексного тестування.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мови програмування PHP та JavaScript, фреймворк Laravel, система керування базами даних PostgreSQL, архітектурний патерн MVC, методи функціонального та нефункціонального тестування, мови розмітки та стилізації HTML5/CSS3, персональний комп'ютер з процесором Celeron Dual-Core T3500 під управлінням операційної системи Microsoft Windows 10.

Результати. Створено функціональну вебсистему інтернет-магазину, що включає основні модулі: каталог товарів, систему пошуку, модуль авторизації, кошик та систему оформлення замовлень. Виконано оцінку продуктивності системи, яка підтвердила скорочення часу обробки запитів завдяки використанню оптимізованих конфігураційних структур даних. Проведено тестування системи: розроблено чекліст, тест-кейси та зафіксовано помилки у баг-репортах. Проведено аналіз виявлених дефектів

та здійснено їх усунення, що підтвердило надійність функціонування системи.

Висновки. У результаті виконання роботи досягнуто поставленої мети: впровадження архітектурних рішень фреймворку Laravel та оптимізація структури бази даних PostgreSQL дозволили зменшити час обробки користувацьких запитів. Проведене комплексне тестування забезпечило підвищення надійності функціонування вебсистеми шляхом ідентифікації та усунення дефектів на етапі розробки, що підтверджує готовність масштабованого програмного продукту до експлуатації.

Галузь використання – сфера роздрібної торгівлі комп'ютерними комплектуючими, периферією та цифровими продуктами для геймерів.

## ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 109 pages, 2 tables, 41 figures, 3 appendixes, 41 sources.

ONLINE STORE, WEB SYSTEM, E-COMMERCE, LARAVEL, POSTGRESQL, SOFTWARE TESTING, MVC ARCHITECTURE.

The object of research is the processes of modeling, development, and quality assurance of scalable software web systems in the field of e-commerce.

The subject of the research is methods, algorithms for optimizing data processing, and software tools for developing and ensuring the quality of an online store web system based on the Laravel framework.

The purpose of the work is to optimize data processing time and increase the reliability of the web system based on the use of modern development methods and comprehensive testing techniques.

Materials, methods and technical means: object-oriented programming, PHP and JavaScript programming languages, Laravel framework, PostgreSQL database management system, MVC architectural pattern, functional and non-functional testing methods, HTML5/CSS3 markup and styling languages, personal computer with Celeron Dual-Core T3500 processor running Microsoft Windows 10 operating system.

Results. A functional web system for an online store was created, including the main modules: a product catalog, a search system, an authorization module, a shopping cart, and an order processing system. A performance assessment of the system was performed, which confirmed the reduction in request processing time due to the use of optimized configuration data structures. The system was tested: a checklist, test cases were developed, and errors were recorded in bug reports. The identified defects were analyzed and their elimination was carried out, which confirmed the reliability of the system.

Conclusions. As a result of the work, the set goal was achieved: the implementation of architectural solutions of the Laravel framework and optimization of the PostgreSQL database structure allowed to reduce the processing time of user requests. The comprehensive testing provided an increase in the reliability of the web system by identifying and eliminating defects at the development stage, which confirms the readiness of the scalable software product for operation.

The industry of use is the retail trade of gaming goods, peripherals, and digital products for gamers.

## ЗМІСТ

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
|                                                                                | С.  |
| Перелік скорочень та умовних познач .....                                      | 9   |
| Вступ .....                                                                    | 10  |
| 1 Аналіз предметної області та постановка завдання .....                       | 12  |
| 1.1 Огляд сучасного стану ринку електронної комерції .....                     | 12  |
| 1.2 Аналіз аналогів та обґрунтування необхідності розробки .....               | 14  |
| 1.3 Формування вимог до вебсистеми інтернет-магазину .....                     | 16  |
| 1.3.1 Функціональні вимоги.....                                                | 17  |
| 1.3.2 Нефункціональні вимоги.....                                              | 18  |
| 1.3.3 Архітектурні особливості системи .....                                   | 18  |
| 2 Вибір та обґрунтування технологій розробки .....                             | 20  |
| 2.1 Аналіз та вибір засобів розробки вебсистеми.....                           | 20  |
| 2.2 Обґрунтування вибору мови програмування PHP та фреймворку<br>Laravel ..... | 24  |
| 2.3 Вибір системи керування базами даних та додаткових технологій .....        | 26  |
| 3 Проєктування та програмна реалізація вебсистеми .....                        | 30  |
| 3.1 Проєктування архітектури системи та структури бази даних .....             | 30  |
| 3.2 Програмна реалізація функціоналу вебсистеми .....                          | 35  |
| 3.3 Опис інструкції користувача.....                                           | 38  |
| 4 Тестування вебсистеми та аналіз результатів .....                            | 44  |
| 4.1 Розробка стратегії тестування, чеклістів та тест-кейсів.....               | 44  |
| 4.2 Проведення тестування та фіксація дефектів у баг-репортах.....             | 48  |
| 4.3 Аналіз результатів тестування та оцінка якості продукту .....              | 51  |
| Висновки.....                                                                  | 54  |
| Перелік джерел посилання .....                                                 | 55  |
| Додаток А Результати виконання чеклісту тестування вебсистеми .....            | 60  |
| Додаток Б Текст програми.....                                                  | 65  |
| Додаток В Слайди презентації .....                                             | 101 |

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

|      |                                                        |
|------|--------------------------------------------------------|
| API  | – Application Programming Interface;                   |
| BR   | – Bug Report;                                          |
| CSS  | – Cascading Style Sheets;                              |
| HTML | – HyperText Markup Language;                           |
| JS   | – JavaScript;                                          |
| MVC  | – Model-View-Controller;                               |
| SQA  | – Software Quality Assurance;                          |
| QSOS | – Qualification and Selection of Open Source Software; |
| TC   | – Test Case;                                           |
| UI   | – User Interface;                                      |
| БД   | – база даних;                                          |
| ЕК   | – електронна комерція;                                 |
| ПЗ   | – програмне забезпечення;                              |
| ПК   | – персональний комп'ютер;                              |
| СКБД | – система керування базами даних;                      |
| ТЗ   | – технічне завдання.                                   |

## ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій електронна комерція займає провідне місце у сфері торгівлі. З кожним роком зростає кількість користувачів, які здійснюють покупки через мережу Інтернет, що обумовлює підвищений попит на зручні, функціональні та безпечні вебсистеми інтернет-магазинів. Особливо актуальним є створення спеціалізованих платформ, орієнтованих на певну цільову аудиторію, зокрема геймерів, які потребують швидкого доступу до цифрових товарів, ігрової атрибутики та супутніх послуг.

Актуальність теми дипломної роботи полягає у необхідності розробки ефективної вебсистеми електронної комерції, яка забезпечує зручний інтерфейс користувача та високу якість функціонування. Важливим аспектом є також проведення комплексного тестування розробленого програмного продукту з метою виявлення помилок, підвищення стабільності роботи та покращення користувацького досвіду.

Метою дипломної роботи є зменшення часу обробки користувацьких запитів та підвищення надійності функціонування вебсистеми інтернет-магазину за рахунок застосування сучасних методів розробки та комплексного тестування програмного забезпечення.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати предметну область та визначити основні вимоги до вебсистем електронної комерції;
- обґрунтувати вибір технологічного стеку та інструментів розробки;
- спроектувати архітектуру системи та структуру бази даних;
- розробити основний функціонал інтернет-магазину;
- сформувати тестову документацію та провести комплексне ручне тестування системи з фіксацією результатів.

Практичне значення отриманих результатів полягає у створенні готової до експлуатації вебсистеми, яка може бути використана як основа для

реального комерційного проєкту. Розроблена тестова документація забезпечує можливість подальшого супроводу та розвитку системи з мінімальними ризиками виникнення критичних помилок.

# **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ**

## **1.1 Огляд сучасного стану ринку електронної комерції**

Електронна комерція є одним із найбільш динамічних напрямів розвитку сучасної цифрової економіки. Завдяки стрімкому розвитку інформаційних технологій та зростанню рівня проникнення мережі Інтернет електронна торгівля набуває все більшого поширення як у світі, так і в Україні. Використання цифрових платформ дозволяє підприємствам значно розширювати ринки збуту, оптимізувати бізнес-процеси та підвищувати конкурентоспроможність.

Під електронною комерцією розуміють комерційну діяльність, пов'язану з купівлею та продажем товарів і послуг із використанням інформаційно-телекомунікаційних систем. Вона охоплює різні види взаємодії, зокрема бізнес-до-споживача (B2C), бізнес-до-бізнесу (B2B), споживач-до-споживача (C2C), а також мобільну комерцію (m-commerce) [1]. Варто зауважити, що електронна комерція є лише частиною ширшого поняття електронного бізнесу, який включає повну цифровізацію всіх процесів компанії, а не тільки самі продажі [2]. Така різноманітність дозволяє електронній комерції адаптуватися до різних умов ринку та задовольняти потреби широкого кола користувачів.

Історично розвиток електронної комерції розпочався ще у 1960-х роках із впровадження систем електронного обміну даними, однак значний поштовх вона отримала з появою мережі Інтернет у 1990-х роках, коли почали з'являтися перші інтернет-магазини. З того часу електронна комерція активно розвивається, а її роль у світовій економіці постійно зростає [1].



Рисунок 1.1 – Гістограма розвитку ринку e-commerce України (2020–2025)  
[3]-[4]

Як можна побачити на рисунку 1.1, сучасний стан ринку електронної комерції в Україні характеризується високими темпами зростання. Збільшення кількості користувачів Інтернету, розвиток платіжних систем та логістичної інфраструктури сприяють активному переходу споживачів до онлайн-покупок. Незважаючи на те, що частка електронної торгівлі у загальному обсязі роздрібного товарообігу поки що залишається відносно невеликою, ринок демонструє стабільну позитивну динаміку та значний потенціал для подальшого розвитку [1].

Важливим фактором розвитку ЕК є також законодавче регулювання. В Україні прийняття закону «Про електронну комерцію» сприяло впорядкуванню правових відносин у цій сфері, визначенню прав та обов'язків учасників електронних угод, а також підвищенню довіри споживачів до онлайн-покупок [1].

Окрім цього, розвиток інтернет-торгівлі супроводжується активним використанням різних каналів продажу, таких як власні інтернет-магазини,

маркетплейси, соціальні мережі та електронні дошки оголошень. Кожен із цих каналів має свої особливості, однак найбільш функціональним і перспективним інструментом є саме власний інтернет-магазин, який забезпечує повний контроль над бізнес-процесами, дизайном і взаємодією з користувачами [5]. При створенні такого магазину важливо забезпечити високу якість обслуговування та позитивний досвід для клієнтів, щоб вони хотіли повертатися за покупками знову [2].

Тематикою вебзастосунку було обрано інтернет-магазин геймерського асортименту, оскільки це один із найбільш популярних та активних напрямків сучасної онлайн-торгівлі. Популярність цієї сфери зумовлена стрімким розвитком ігрової індустрії та постійною появою нових технологічних рішень, що забезпечує стабільно високий попит на сучасне обладнання та комплектуючі серед активної аудиторії користувачів.

Таким чином, сучасний ринок електронної комерції характеризується високими темпами розвитку, постійним зростанням кількості користувачів та розширенням функціональних можливостей вебсистем. В Україні електронна комерція перебуває на етапі активного зростання, що створює сприятливі умови для розробки нових вебзастосунків, зокрема спеціалізованих інтернет-магазинів.

## **1.2 Аналіз аналогів та обґрунтування необхідності розробки**

На сучасному ринку електронної комерції України представлено велику кількість інтернет-магазинів, що пропонують широкий спектр товарів і послуг. Серед найбільш відомих та популярних платформ можна виділити такі інтернет-магазини, як ROZETKA, COMFY та TELEMART. Зазначені вебресурси є прикладами успішної реалізації електронної комерції та демонструють сучасні підходи до організації онлайн-продажів.

ROZETKA – найбільший інтернет-магазин і маркетплейс України. Його перевагами є величезний асортимент, висока довіра користувачів та

відпрацьована логістика. Однак для вузькоспеціалізованого покупця цей ресурс характеризується надмірною універсальністю, де складно знайти детальні технічні характеристики спеціалізованого апаратного забезпечення або отримати експертну підтримку саме в ігровій сфері.

COMFY – приклад успішної омніканальної роздрібної торгівлі. Магазин має високий рівень юзабіліті та якісну адаптивну верстку, проте він орієнтований на масового споживача побутової техніки та електроніки, що робить геймерський асортимент лише супутньою категорією.

TELEMART – є найбільш близьким аналогом, оскільки спеціалізується саме на комп'ютерних комплектуючих та ігровому обладнанні. Він пропонує унікальні сервіси, як-от «Конфігуратор ПК». Це підтверджує, що вузька спеціалізація має сталий попит серед цільової аудиторії, яка потребує експертного підходу.

Для більш детального аналізу аналогів було складено таблицю 1.1.

Таблиця 1.1 – Порівняльна характеристика аналогів інтернет-магазину

| Критерій порівняння                        | ROZETKA | COMFY    | TELEMART  |
|--------------------------------------------|---------|----------|-----------|
| Швидкість завантаження та зворотній відгук | Висока  | Середня  | Середня   |
| Фокус на цільову товарну нішу              | Низький | Середній | Високий   |
| Зручність інтерфейсу                       | Висока  | Висока   | Середня   |
| Спеціалізована фільтрація                  | Базова  | Базова   | Просунута |
| Наявність особистого кабінету              | Так     | Так      | Так       |

Аналіз існуючих рішень показує, що кожен із розглянутих інтернет-магазинів має свої сильні сторони, однак жоден із них не забезпечує повного поєднання зручності, спеціалізації та функціональності для конкретної цільової аудиторії, зокрема геймерів. Це створює передумови для розробки нової вебсистеми, яка враховуватиме потреби користувачів та поєднуватиме переваги існуючих аналогів.

Крім того, як зазначається у дослідженнях, створення інтернет-магазину є лише початковим етапом розвитку електронної торгівлі. Для досягнення високих показників ефективності необхідно постійно вдосконалювати структуру вебресурсу, покращувати його зручність та застосовувати сучасні методи інтернет-маркетингу [5]. Це свідчить про те, що навіть існуючі успішні платформи потребують подальшого розвитку та оптимізації.

Таким чином, проведений аналіз аналогів показав, що, незважаючи на наявність великої кількості інтернет-магазинів, існує потреба у створенні спеціалізованих вебсистем, орієнтованих на конкретні категорії користувачів. Зокрема, розробка інтернет-магазину для геймерів дозволить врахувати специфічні потреби даної аудиторії, забезпечити більш зручний інтерфейс, покращену навігацію та функціональні можливості, що в цілому підвищить ефективність взаємодії користувачів із системою. При цьому сучасні інтернет-магазини реалізуються у вигляді вебдодатків, які, на відміну від звичайних сайтів, забезпечують високу інтерактивність, не потребують встановлення та автоматично оновлюються для всіх користувачів одночасно [6].

### **1.3 Формування вимог до вебсистеми інтернет-магазину**

На основі проведеного аналізу предметної області та існуючих аналогів, необхідно сформулювати перелік вимог до проектованої вебсистеми. Визначення вимог є важливим етапом розробки, оскільки саме вони

визначають функціональні можливості системи, рівень її зручності для користувача та ефективність взаємодії з нею.

При формуванні вимог важливо враховувати психологічні аспекти поведінки користувачів. Процес прийняття рішення про покупку в електронній комерції значною мірою залежить від емоційного стану користувача: позитивні емоції підвищують ймовірність здійснення покупок і рівень довіри до бренду, тоді як негативні можуть стримувати користувача від взаємодії з вебресурсом [7]. Це означає, що інтерфейс вебсистеми повинен бути інтуїтивно зрозумілим, естетично привабливим та не викликати у користувача труднощів під час використання. Тому вимоги до магазину варто формувати так, щоб кожна функція мотивувала людину до покупки і робила процес замовлення максимально легким, адже саме це змушує клієнтів повертатися знову [8].

Враховуючи вищезазначене, вимоги до системи поділяються на дві основні групи: функціональні та нефункціональні. Варто зазначити, що ці вимоги є базою для майбутнього тестування: функціональні – допомагають перевірити правильність результатів кожної дії, а нефункціональні – допомагають оцінити загальну якість та швидкість роботи системи [9].

### **1.3.1 Функціональні вимоги**

Вони описують конкретні дії, які користувач зможе виконувати на сайті:

а) модуль керування обліковим записом:

- 1) реєстрація та авторизація користувачів;
- 2) особистий кабінет із можливістю редагування персональних даних;
- 3) безпечна зміна пароля та можливість повного видалення акаунту користувачем;
- 4) функція безпечного виходу з облікового запису;

б) модуль каталогу та пошуку:

- 1) відображення списку товарів у каталозі;
- 2) детальна картка з технічними характеристиками, фото та описом товару;
- 3) система пошуку товарів за назвою для швидкої навігації;

в) модуль кошика та оформлення замовлення:

- 1) додавання товарів до кошика, зміна їхньої кількості та видалення;
- 2) оформлення замовлення через спеціалізовану форму.

### **1.3.2 Нефункціональні вимоги**

Ці вимоги відповідають за те, наскільки якісно працює сайт:

- надійність та безпека: Захист персональних даних користувачів та збереження конфіденційності під час авторизації.
- продуктивність: Система повинна забезпечувати швидке завантаження сторінок.
- адаптивність та кросбраузерність: Коректне відображення інтерфейсу на різних типах пристроїв та у сучасних веббраузерах.
- зручність використання(Usability): Інтерфейс повинен бути простим та інтуїтивно зрозумілим.

Таким чином, сформовані вимоги дозволяють розробити вузькоспеціалізовану систему, яка мінімізує негативні психологічні фактори та забезпечує ефективну взаємодію користувача з інтернет-магазином.

### **1.3.3 Архітектурні особливості системи**

Керування товарним асортиментом (додавання, редагування, видалення) буде реалізовано на рівні вихідного коду через окремі файли для кожного товару. Це дозволить відмовитися від розробки окремої панелі

адміністратора, спростити архітектуру керування даними та забезпечити максимальну швидкість обробки запитів на стороні клієнта.

## **2 ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ**

Розробка сучасної вебсистеми потребує обґрунтованого вибору технологій, інструментів та програмних засобів, що забезпечують її ефективність, надійність і масштабованість. Від правильного підбору мови програмування, фреймворку та системи керування базами даних залежить продуктивність системи, зручність її супроводу та подальшого розвитку. У цьому розділі буде проведено аналіз існуючих засобів розробки та обґрунтовано вибір технологій, які доцільно використати для реалізації вебсистеми інтернет-магазину.

### **2.1 Аналіз та вибір засобів розробки вебсистеми**

Розробка вебсистеми інтернет-магазину передбачає використання комплексу сучасних програмних засобів і технологій, які забезпечують створення функціонального, надійного та зручного для користувача продукту. Вибір відповідних інструментів розробки є важливим етапом, оскільки він впливає на продуктивність системи, її масштабованість, безпеку та зручність супроводу.

Для створення вебсистеми необхідно враховувати як клієнтську, так і серверну частини застосунку.

Клієнтська частина відповідає за відображення інтерфейсу користувача та взаємодію з ним. Для її реалізації доцільно використовувати мови розмітки та стилізації, такі як HTML та CSS, а також мову програмування JavaScript, яка забезпечує динамічну поведінку вебсторінок.

Серверна частина відповідає за обробку запитів користувачів, взаємодію з базою даних та реалізацію бізнес-логіки системи. Для її реалізації можуть використовуватися різні мови програмування. Вибір конкретної мови залежить від вимог до системи, складності проєкту, швидкості розробки та доступності ресурсів.

На етапі вибору мови програмування для серверної частини було розглянуто кілька популярних рішень, що широко застосовуються у веброзробці: Java, Ruby, Python та PHP. Кожна з цих мов має свої технічні особливості [10]:

- Java є стандартом для корпоративних систем завдяки високій об'єктно-орієнтованій абстракції та кросплатформеності, проте вона вимагає значних апаратних ресурсів та має високий поріг входження;

- Ruby є динамічною мовою програмування, що використовується у веброзробці. Відома своєю простотою та швидкістю написання коду, але часто поступається іншим мовам у продуктивності при високих навантаженнях;

- Python є універсальною мовою з лаконічним синтаксисом, проте її використання у веброзробці іноді вимагає додаткових налаштувань для досягнення максимальної швидкодії порівняно з мовами, що були створені спеціально для вебу;

- PHP є однією з найпопулярніших серверних мов програмування для створення вебзастосунків. Вона спеціально розроблена для генерації динамічного вебконтенту, легко інтегрується з HTML, підтримує роботу з більшістю систем керування базами даних та є кросплатформеною. Завдяки своїй простоті, швидкості виконання та широкій підтримці, PHP використовується для створення великої кількості сучасних вебсистем.

Серед наведених мов програмування можна зробити висновок, що кожна з них має свої переваги та недоліки, і вибір залежить від специфіки проєкту. Для розробки інтернет-магазину доцільно обрати мову PHP, оскільки ця мова є оптимальною з точки зору швидкості розробки, простоти використання, продуктивності та широкої підтримки вебтехнологій.

Проте сучасна розробка на «чистому» PHP є неефективною через велику кількість однотипного коду. Використання PHP-фреймворків дозволяє значно прискорити процес створення продукту, забезпечуючи стабільність та дотримання стандартів кодування [11]. Для вибору

оптимального фреймворку було проаналізовано показники трьох лідерів ринку: Laravel, Symfony та CodeIgniter.

Згідно з тестуванням продуктивності, за методологією QSOS, фреймворк Laravel демонструє суттєві переваги над конкурентами за ключовими параметрами [11]:

- швидкість обробки запитів (див. рис. 2.1): Laravel здатний обробляти до 3000 запитів на секунду, що значно перевищує показники аналогів;

- час відгуку системи (див. рис. 2.2): Середній час відгуку Laravel складає всього 4.46 мс, тоді як у CodeIgniter цей показник становить 7.2 мс, а у Symfony – 12 мс;

- використання пам'яті (див. рис. 2.3): Laravel виявився найбільш енергоефективним, споживаючи менше оперативної пам'яті для відображення сторінки.

Результати тестування продуктивності представлені на рисунках 2.1-2.3 [11].

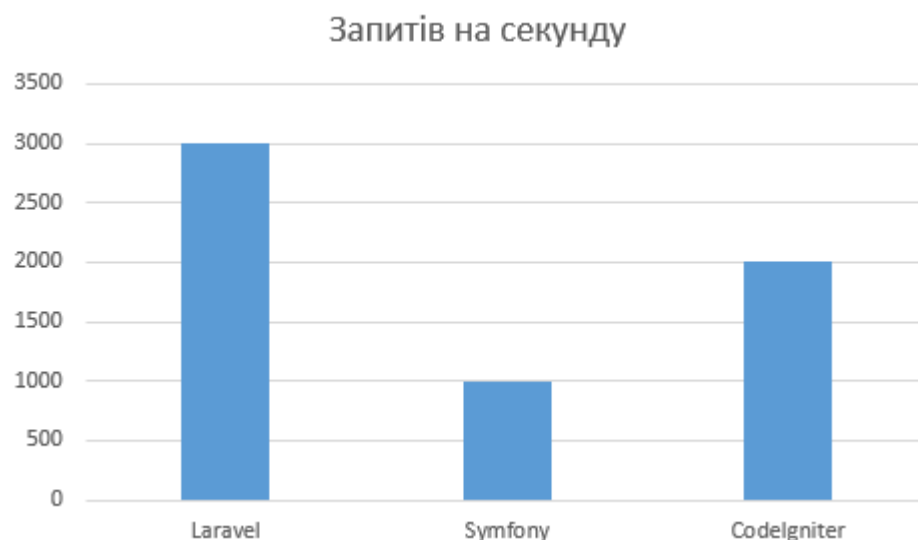


Рисунок 2.1 – Порівняння швидкості обробки запитів фреймворків Laravel, Symfony та CodeIgniter [11]

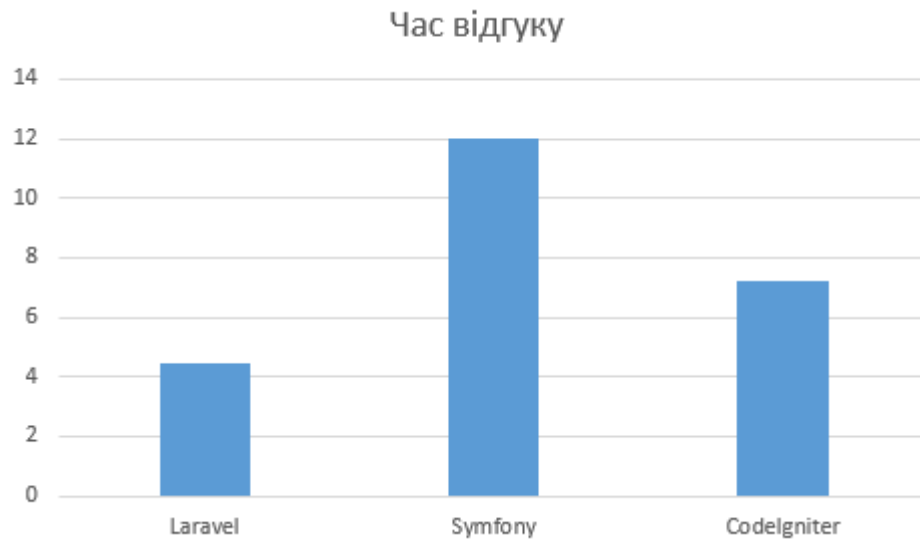


Рисунок 2.2 – Порівняння часу відгуку фреймворків [11]

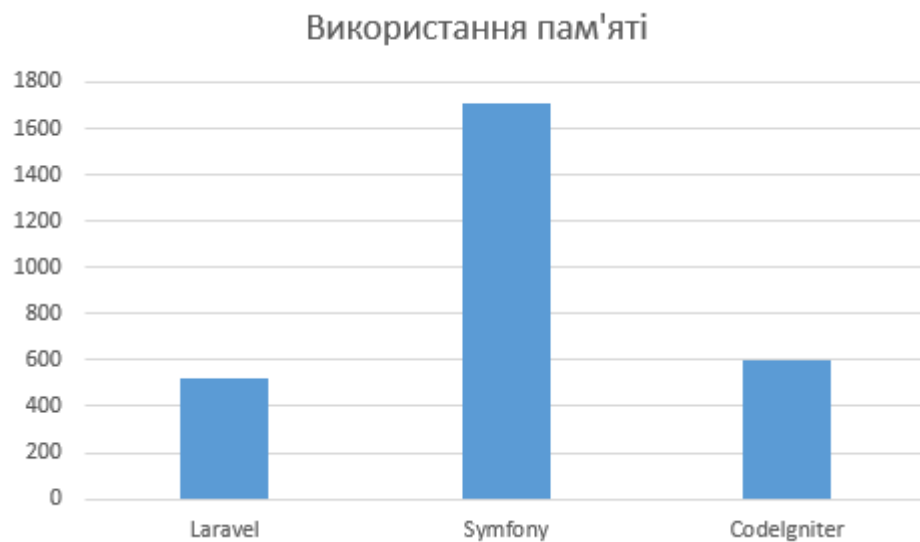


Рисунок 2.3 – Порівняння використання оперативної пам'яті [11]

Для написання програмного коду та редагування конфігураційних файлів було обрано текстовий редактор Notepad++. Вибір цього середовища розробки обґрунтовується його високою швидкістю роботи та мінімальним споживанням системних ресурсів. Це є критично важливим фактором для забезпечення стабільної та комфортної роботи на старих ПК, де запуск важких інтегрованих середовищ розробки міг би спричинити затримки в роботі.

Таким чином, для реалізації вебсистеми інтернет-магазину обрано мову програмування PHP, фреймворк Laravel та середовище розробки Notepad++. Це рішення дозволяє створювати великомасштабні додатки з мінімальними витратами ресурсів, забезпечуючи при цьому високу безпеку коду завдяки вбудованим механізмам захисту.

## **2.2 Обґрунтування вибору мови програмування PHP та фреймворку Laravel**

Для реалізації серверної частини інтернет-магазину було обрано мову PHP. На сьогодні це одна з найпопулярніших та найдоступніших мов для створення вебсистем: за статистикою, її використовують близько 79.2% усіх вебсайтів для серверної розробки [12]. Вона дозволяє створювати динамічні вебсторінки, працювати з базами даних та обробляти запити користувачів у режимі реального часу.

Проте розробка масштабного проєкту на «чистому» PHP забирає багато часу і є досить ризикованою. Адже сучасні вебдодатки постійно стикаються із загрозами безпеці. Найпоширенішими загрозами є SQL-ін'єкції, міжсайтовий скриптинг (XSS) та підробка міжсайтових запитів (CSRF) [13]. І якщо писати код з нуля, розробнику доводиться самостійно передбачати захист від усіх цих вразливостей.

Щоб уникнути цих проблем та пришвидшити процес розробки, було прийнято рішення використовувати сучасний фреймворк. Серед PHP-фреймворків найкращим вибором є Laravel. Його головна зручність полягає у використанні зрозумілої архітектури MVC (Model-View-Controller) [14]. Вона ділить весь код проєкту на три логічні частини:

- модель (Model): відповідає за роботу з даними. У Laravel для цього є зручна система Eloquent ORM. Вона дозволяє працювати з таблицями бази даних за допомогою простих команд PHP, без необхідності писати довгі SQL-запити вручну;

- представлення (View): це та частина, яку бачить користувач на екрані. Laravel використовує вбудований шаблонізатор Blade, який допомагає легко та швидко виводити інформацію на вебсторінки;
- контролер (Controller): отримує запит від користувача (наприклад, перегляд сторінки товарів), звертається до Моделі за потрібними даними і передає їх у Представлення для показу на екрані.

Схематичне представлення взаємодії компонентів MVC наведено на рисунку 2.4.

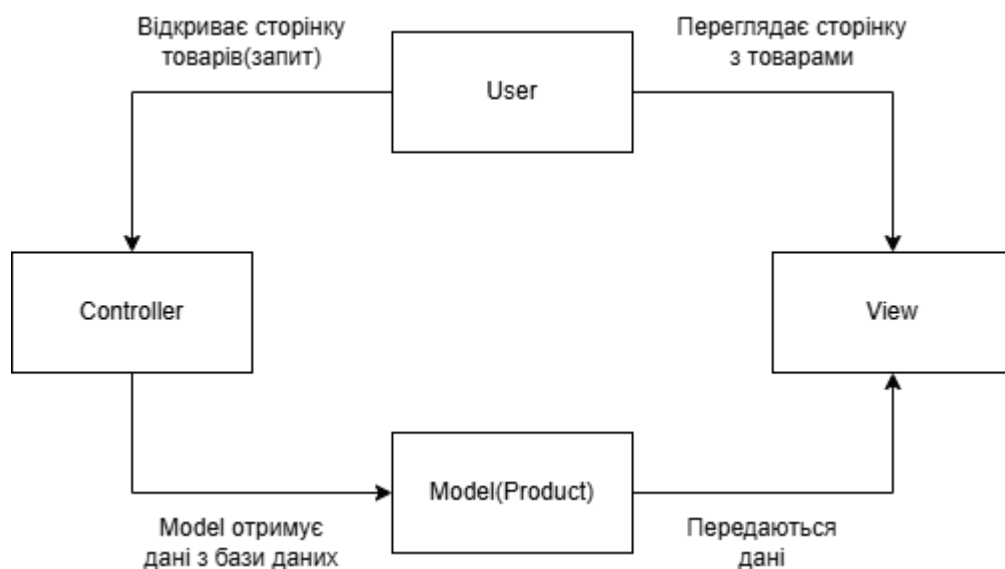


Рисунок 2.4 – Архітектурна модель MVC на прикладі процесу перегляду товарів

Важливим параметром Laravel є потужна система маршрутизації. Вона дозволяє чітко визначати URL-адреси застосунку та спрямовувати запити до відповідних контролерів, що робить систему масштабованою та легкою у супроводі [14]. Також важливим інструментом у Laravel є Artisan — вбудований інтерфейс командного рядка, який дозволяє автоматизувати повторювані завдання, такі як створення моделей, контролерів та міграцій, що суттєво економить час [12]. Для управління структурою бази даних використовуються міграції, які виконують роль системи контролю версій для

бази даних, забезпечуючи узгодженість структури таблиць на різних етапах розробки.

Окрему увагу при виборі Laravel було приділено питанням безпеки. Відповідно до аналізу загроз вебдодаткам [13], критично важливим є захист від таких атак:

- SQL-ін'єкції: Фреймворк використовує механізм підготовлених запитів, що унеможлиблює впровадження шкідливого SQL-коду;
- Cross-Site Request Forgery (CSRF): Фреймворк захищає вебсистему за допомогою генерації спеціальних токенів для кожного активного сеансу користувача;
- Cross-Site Scripting (XSS): Шаблонізатор Blade автоматично очищає будь-які дані перед їх виведенням на сторінку, що унеможлиблює виконання шкідливих XSS-скриптів.

Отже, використання мови PHP разом із фреймворком Laravel є оптимальним вибором для розробки. Це рішення поєднує в собі високу швидкість розробки, надійний захист та легкість у масштабуванні системи, що дозволяє створити якісну вебсистему у коротші терміни.

## **2.3 Вибір системи керування базами даних та додаткових технологій**

Будь-яка сучасна вебсистема, особливо інтернет-магазин, потребує надійного місця для зберігання інформації. Для цього використовуються бази даних. База даних являє собою впорядковану сукупність даних, які зберігаються в електронному вигляді та можуть оброблятися, змінюватися і використовуватися за потреби [15]. У вебдодатках вона використовується для зберігання інформації про користувачів, товари, замовлення та інші важливі дані.

Для роботи з базами даних використовуються системи керування базами даних. Вони дозволяють створювати, змінювати, видаляти та

отримувати дані, а також забезпечують їх безпеку та цілісність [16]. На сьогодні існують різні типи СКБД, кожна з яких підходить для своїх завдань [16]:

- реляційні: зберігають дані у вигляді звичних таблиць із рядками та стовпцями. Вони використовують мову SQL і найкраще підходять для структурованої інформації. Приклади: MySQL, PostgreSQL;
- нереляційні: працюють із більш гнучкими структурами, як-от документи або пари «ключ-значення». Їх обирають, коли дані часто змінюються або їх дуже багато. Приклад: MongoDB;
- об'єктно-орієнтовані: зберігають дані у вигляді об'єктів, які мають властивості та методи;
- ієрархічні та мережеві: використовують деревовидні або складні зв'язки між даними.

Для вебсистем, зокрема інтернет-магазинів, найчастіше використовуються саме реляційні бази даних, оскільки вони добре підходять для роботи зі структурованою інформацією, такою як профілі користувачів та характеристики товарів.

Для початку роботи з базою даних потрібно вибрати СКБД. Провівши аналіз сучасного ринку СКБД, можна виділити кілька лідерів [16]-[17]:

- Oracle та Microsoft SQL Server: це потужні комерційні реляційні СКБД. Вони дуже надійні, але через високу вартість ліцензій використовуються переважно великими корпораціями;
- MySQL: безкоштовна реляційна СКБД. Вона проста у використанні, швидка, за рахунок чого дуже популярна у веброзробці, особливо для простих вебзастосунків, де пріоритетними є висока швидкість читання та легкість налаштування;
- MongoDB: один з найкращих представників нереляційних систем, що підходить для роботи з великими обсягами неструктурованих даних;

– PostgreSQL: потужна та гнучка об'єктно-реляційна система з відкритим кодом, яка відмінно підходить для складних і високонавантажених систем, де важлива підтримка транзакцій та складна бізнес-логіка.

Проаналізувавши всі ці варіанти, для розробки даної вебсистеми було обрано PostgreSQL. Ця СКБД має кілька суттєвих переваг [17]-[18]:

– вона повністю безкоштовна, активно підтримується і постійно покращується розробниками;

– PostgreSQL дотримується принципів ACID (атомарність, узгодженість, ізолюваність, довговічність) у повному обсязі, на відміну від деяких конфігурацій MySQL. Це означає, що будь-яка транзакція в базі або виконується повністю (наприклад, успішна реєстрація), або скасовується цілком, що унеможливорює пошкодження або втрату даних при збоях системи;

– вона забезпечує високий рівень безпеки даних, що є критично важливим критерієм для вебсистем електронної комерції;

– у цій СКБД висока продуктивність при роботі з великими обсягами даних.

Сучасні дослідження підтверджують, що архітектура PostgreSQL забезпечує вищу стабільність та передбачувану затримку при великій кількості одночасних запитів, тоді як MySQL може демонструвати зниження швидкодії через конфлікти блокувань [19]. Окрім того, завдяки вдосконаленому планувальнику запитів, ця СКБД демонструє значну перевагу в обробці складних операцій, що включають багатотабличні зв'язки та вкладені підзапити, споживаючи при цьому менше ресурсів процесора [19].

У межах розроблюваної вебсистеми PostgreSQL буде використовуватися для реалізації найбільш відповідального модуля – системи реєстрації та авторизації. У базі даних буде зберігатися конфіденційна інформація про користувачів: унікальний логін, ім'я, дата

народження та хешований пароль, що гарантує захист облікових записів від зловмисників.

Що стосується інформації про асортимент товарів інтернет-магазину, то в рамках даного навчального проєкту було прийнято рішення зберігати її у вигляді асоціативного масиву в конфігураційному файлі проєкту. Такий підхід має вагому перевагу: він дозволяє системі миттєво завантажувати сторінки товарів без необхідності робити додаткові запити до бази даних, що суттєво економить ресурси сервера та пришвидшує відображення сторінок для клієнта. У перспективі, при значному розширенні асортименту та додаванні функціоналу адміністративної панелі, масив товарів можна буде легко мігрувати до створеної бази даних.

Таким чином, вибір PostgreSQL як основної системи керування базами даних є цілком виправданим для вебсистеми інтернет-магазину. Ця СКБД забезпечує високу швидкість обробки запитів та надійний захист персональних даних користувачів.

## **3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБСИСТЕМИ**

### **3.1 Проєктування архітектури системи та структури бази даних**

Проєктування архітектури вебсистеми є одним із ключових етапів розробки, оскільки саме на цьому етапі визначаються основні компоненти системи, їх взаємодія та принципи обробки даних. Для створення сучасного інтернет-магазину було обрано клієнт-серверну архітектуру, яка є стандартом для вебзастосунків.

Згідно з клієнт-серверною моделлю, система складається з двох основних частин: клієнта та сервера. Клієнт – це браузер користувача, який формує запити до сервера та відображає отримані результати, тоді як сервер обробляє запити, виконує бізнес-логіку та забезпечує доступ до бази даних [20]. Винесення основної обчислювальної логіки на серверну частину дозволяє уникнути дублювання програмного коду на клієнтських пристроях та економить їхні ресурси [21]. Крім того, такий підхід суттєво підвищує безпеку вебсистеми, оскільки конфіденційні дані та внутрішня структура бази даних залишаються захищеними від прямого втручання з боку користувача [21].

У даній вебсистемі реалізовано трирівневу архітектуру, яка включає:

- рівень представлення (Клієнт);
- рівень прикладної логіки (Сервер);
- рівень керування ресурсами (База даних).

Такий підхід дозволяє ефективно розподілити навантаження між компонентами системи та підвищує загальну продуктивність системи [20]. Взаємодія між цими рівнями відбувається за принципом «запит-відповідь», де на рівні представлення реалізується взаємодія користувача з вебсистемою та відображення інтерфейсу, тоді як рівень прикладної логіки відповідає за обробку запитів, реалізацію бізнес-логіки та взаємодію з базою даних. Взаємозв'язок між клієнтською та серверною частинами забезпечує повноцінне функціонування вебзастосунку [22].

Для організації взаємодії між компонентами системи було обрано архітектурний шаблон MVC (Model-View-Controller), теоретичне обґрунтування якого наведено у підрозділі 2.2. Використання MVC забезпечує «слабку пов'язаність» між компонентами застосунку, що дозволяє реалізувати принцип паралельної розробки, коли робота над інтерфейсом, логікою та базою даних може відбуватися незалежно. Окрім цього, такий підхід значно полегшує тестування системи, оскільки кожен компонент можна перевірити окремо [23].

Процес обробки користувацького запиту в межах обраної архітектури відбувається за наступним алгоритмом:

- запит: Користувач взаємодіє з інтерфейсом (View), наприклад, виконує авторизацію;
- маршрутизація та Контроль: Запит надходить до контролера, який аналізує дію користувача та визначає необхідний сценарій обробки;
- обробка даних: Контролер звертається до відповідної моделі для отримання необхідної інформації: наприклад, перевірки введеного логіна та пароля при авторизації;
- формування відповіді: Отримані дані повертаються до контролера, який передає їх у відповідне представлення (View);
- відображення: Готова сторінка повертається в браузер клієнта для візуалізації.

Схематично взаємодію компонентів та шлях проходження даних під час виконання операцій (наприклад, авторизації) зображено на рисунку 3.1.

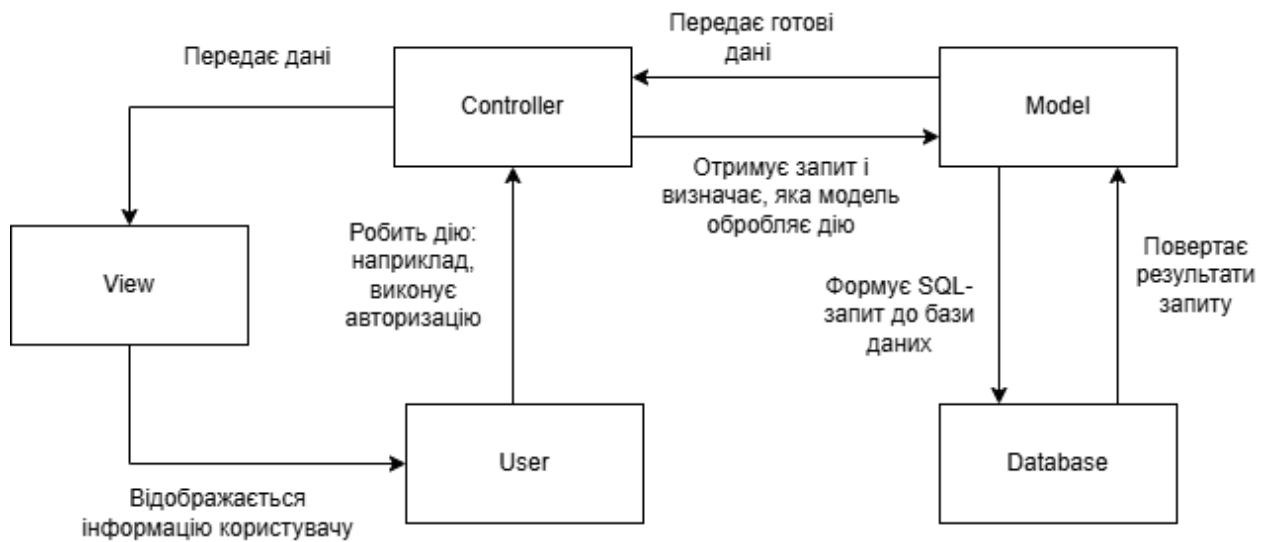


Рисунок 3.1 – Схема тривірневої архітектури та процес обробки запитів у вебсистемі

Окрім побудови логічної схеми роботи системи, важливим етапом проєктування є візуалізація майбутнього інтерфейсу. Дизайн-процес зазвичай проходить кілька стадій розробки макетів, що дозволяє поступово переходити від загальної структури до детального вигляду [24].

Першим етапом стало створення варфреймів. Це прості схеми сторінок, які допомагають швидко зафіксувати розташування кнопок, текстів та картинок, не витрачаючи час на оформлення. На рисунку 3.2 представлено варфрейм сторінки товару. Тут було визначено, як саме будуть розміщені медіафайли, характеристики та опис продукту, щоб користувачу було зручно вивчати асортимент.

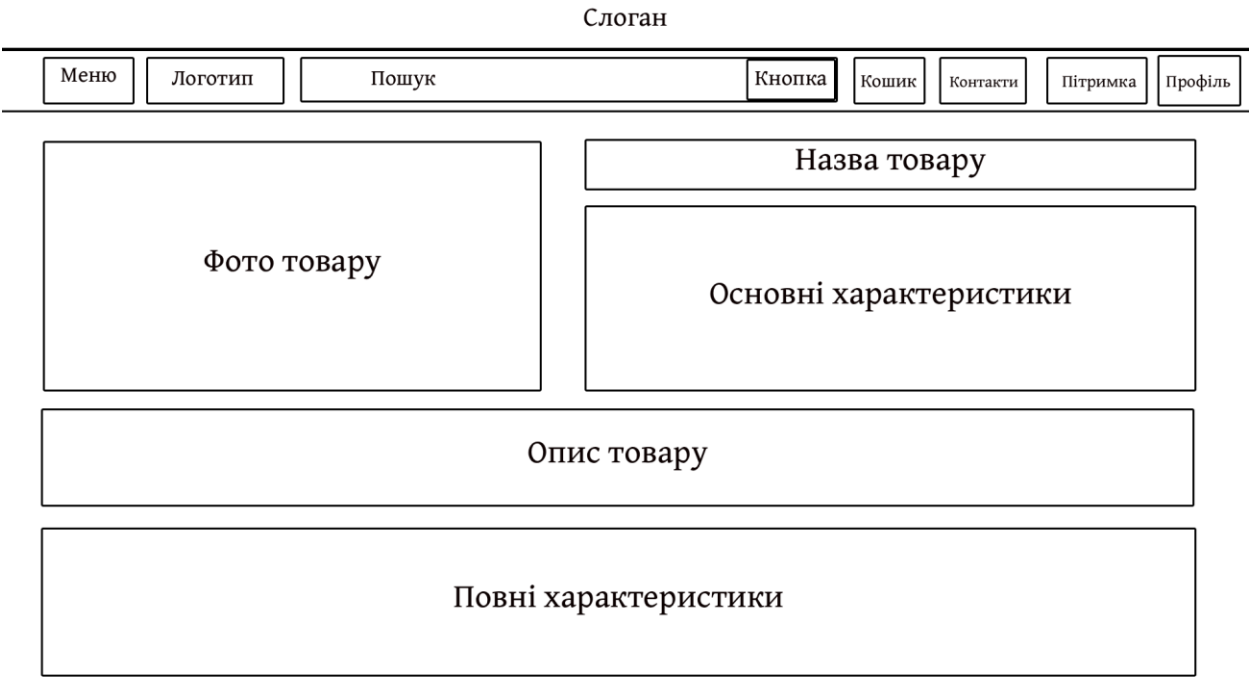


Рисунок 3.2 – Варфрейм сторінки представлення товару

Також було розроблено схему для сторінки технічної підтримки (рис. 3.3). Використання варфрейму на цьому етапі допомогло продумати зручне розташування розділу з питаннями та форми для звернень, що значно спростило подальшу верстку цих елементів.

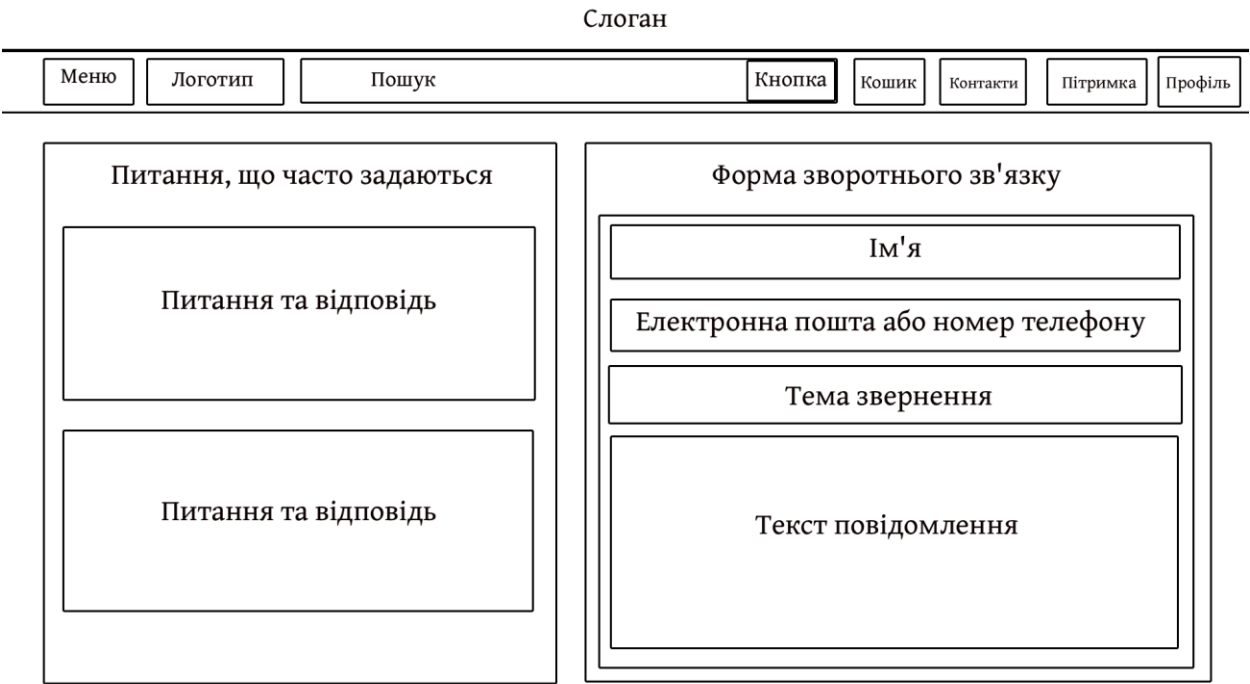


Рисунок 3.3 – Варфрейм сторінки «Довідниковий центр»

Згодом для деталізації інтерфейсу було розроблено мокап. Якщо варфрейм – це лише план, то мокап – це вже повноцінна візуальна концепція, де додаються кольори, зображення та фінальний стиль. Це дозволяє побачити, як сайт буде сприйматися користувачем у реальності [24]. На рисунку 3.4 зображено мокап головної сторінки магазину.

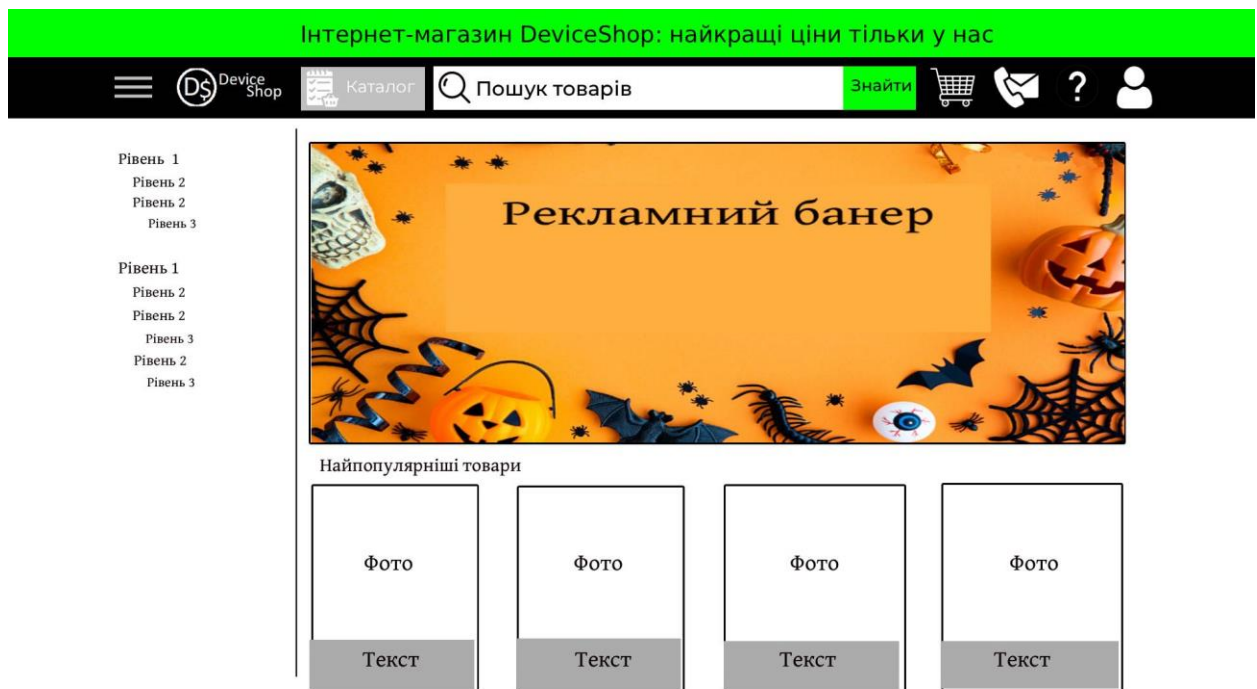


Рисунок 3.4 – Мокап головної сторінки вебсистеми

Наступним кроком проектування є визначення структури збереження даних. Оскільки вебсистема передбачає роботу з користувачами та асортиментом товарів, було спроектовано наступну модель даних:

– база даних PostgreSQL: призначена для керування обліковими записами. Проектована таблиця users містить ідентифікатор (id), повне ім'я (name), унікальний логін (login), дату народження (birthday) та зашифрований пароль (password). Також в структурі таблиці будуть службові поля created\_at та updated\_at, які дозволяють фіксувати точний час створення запису та його останнього редагування. Використання реляційної СКБД забезпечує цілісність даних та надійність виконання транзакцій;

– конфігураційне сховище товарів: на етапі розробки дані про асортимент магазину (назва, ціна, шлях до зображення) винесені в окремий асоціативний масив у файлі `products.php`. Це дозволяє моделі швидко отримувати дані без виконання складних SQL-запитів, що пришвидшує функціонування кошика та обробки вибраних товарів.

Таким чином, спроектована архітектура системи та структура даних забезпечують баланс між продуктивністю, безпекою та можливістю подальшого розширення функціоналу інтернет-магазину.

### **3.2 Програмна реалізація функціоналу вебсистеми**

Програмна реалізація вебсистеми здійснювалася у кілька етапів: від створення клієнтського інтерфейсу до налаштування серверної логіки та інтеграції з базою даних.

На початковому етапі було створено статичний каркас системи. Використання мови розмітки HTML5 дозволило сформувати структуру сторінок: головної, каталогу, сторінок 23-х окремих товарів, сторінки кошику та оформлення замовлення тощо. Для кожного товару було створено окрему сторінку з назвою, описом, характеристиками, ціною та зображенням. Для візуального оформлення застосовано CSS3, що забезпечило єдину стилістику елементів, анімаційні ефекти кнопок та коректне відображення на різних типах пристроїв. Для надання інтерфейсу інтерактивності використано JavaScript. Зокрема, було реалізовано:

- динамічне оновлення елементів кошика;
- взаємодію та валідацію даних у формах;
- обробку подій натискання кнопок.

Для переходу до динамічної розробки було підготовлено серверне середовище. Основним інструментом управління став менеджер пакетів Composer. Процес розгортання включав наступні кроки.

Крок 1. Встановлення інтерпретатора PHP та менеджера пакетів Composer.

Крок 2. Інсталяція фреймворку Laravel за допомогою команди `composer create-project laravel/laravel myshop`.

Крок 3. Налаштування структури проєкту, де статичні HTML-сторінки були інтегровані в систему шаблонізації Blade та збережені з розширенням `*.blade.php` у директорії `resources/views`. У цій теці розміщено як базові шаблони (`main.blade.php`, `catalog.blade.php`), так і файли для кожного з 23 товарів (наприклад, `controllerLogitech-Driving-Force-Shifter.blade.php`). Усі статичні ресурси (CSS-файли, JS-скрипти та зображення) було перенесено до директорії `public`, яка є відкритою точкою доступу вебсервера.

Структура створеного проєкту включає основні директорії:

- `app/` — містить контролери та моделі;
- `config/` — конфігураційні файли;
- `public/` — статичні ресурси;
- `resources/views/` — шаблони сторінок;
- `routes/` — файли маршрутів.

Наступним етапом стала інтеграція бази даних. Для збереження конфіденційних даних користувачів було встановлено СКБД PostgreSQL. Підключення до бази даних реалізовано через конфігураційний файл `.env`, де було задано такі параметри: драйвер підключення (`DB_CONNECTION=pgsql`), хост та порт (`DB_HOST=127.0.0.1`, `DB_PORT=5432`), назва бази (`DB_DATABASE=deviceshop`), а також реквізити доступу (`DB_USERNAME=postgres`, `DB_PASSWORD=1234`).

Формування структури таблиць БД здійснювалося за допомогою механізму міграцій Laravel. Було використано базову міграцію `0001_01_01_000000_create_users_table.php` для створення таблиці користувачів, після чого створено додаткову міграцію `2026_02_04_161103_add_name_and_birthday_to_users_table.php`, яка

розширила таблицю новими необхідними полями — іменем (name) та датою народження (birthday).

Оскільки серверна частина системи базується на шаблоні MVC, наступним кроком у межах реалізації було створено ключові компоненти:

а) моделі (Models): Для взаємодії з базою даних розроблено модель User.php. Вона використовує технологію Eloquent ORM, що дозволяє працювати з записами БД як з об'єктами. У моделі визначено атрибути для масового заповнення (login, password, name, birthday) та встановлено зв'язки «один-до-багатьох» із замовленнями та кошиком. Тобто один користувач може мати багато замовлень та багато товарів у кошику;

б) контролери (Controllers): Бізнес-логіка розподілена за функціональними контролерами. У системі були створені такі контролери:

1) AuthController: реалізує реєстрацію, аутентифікацію (з використанням фасаду Auth), безпечний вихід із системи, зміну паролю, оновлення інформації та повноцінне видалення акаунту;

2) CartController: керує станом кошика користувача, додаванням, зміною кількості та видаленням товарів;

3) SearchController: забезпечує функціонал пошуку товарів за назвою;

4) SupportController та CheckoutController: відповідають за обробку зворотного зв'язку та фіналізацію замовлень, виводячи підтверджувальні повідомлення користувачу.

Як було сказано в попередньому підпункті, для зберігання інформації про товари використано конфігураційний файл config/products.php, який містить асоціативний масив із характеристиками товарів:

- назва;
- ціна;
- зображення;
- маршрут сторінки.

Це зроблено для того, що коли користувач натискає кнопку «Купити» на сторінці товару, то дані з цього масиву підтягувались у кошик, тим самим додаючи товар у кошик з потрібними характеристиками.

Керування всіма запитами вебсистеми здійснюється у файлі `routes/web.php`. Усі маршрути об'єднані в групу з використанням `Middleware` для локалізації, що дозволяє системі підтримувати багатомовність. Є три типи маршрутів:

- публічні (`GET`): для відображення головної сторінки, каталогу та сторінок товарів;
- дії (`POST`): для обробки форм реєстрації, входу та керування кошиком;
- захищені (`middleware 'auth'`): для доступу до профілю користувача, зміни пароля чи видалення облікового запису.

Таким чином, програмна реалізація системи базується на принципі модульності, де кожен компонент виконує чітко визначену роль, що забезпечує стабільність роботи інтернет-магазину.

### **3.3 Опис інструкції користувача**

Розробка інструкції користувача є невід'ємною частиною проєктування вебсистеми, оскільки вона забезпечує розуміння принципів взаємодії з інтерфейсом та функціональними можливостями інтернет-магазину. Опис базується на сценаріях використання системи, починаючи від першого знайомства з головною сторінкою і закінчуючи процесом оформлення замовлення. Інтерфейс системи спроектований за принципами інтуїтивності, що дозволяє користувачеві виконувати цільові дії з мінімальною кількістю переходів.

Взаємодія з вебсистемою починається з головної сторінки (рис. 3.5), де користувача зустрічає інформаційне вітання та навігаційна панель, які будуть доступні на кожній сторінці інтернет-магазину. У верхній частині

екрана розміщено меню, яке містить логотип для швидкого повернення на головну сторінку, рядок пошуку товарів, кнопки швидкого доступу до кошика, довідникового центру, контактів та персонального кабінету. Важливою особливістю є наявність бічної панелі (Sidebar), яка викликається за допомогою кнопки «бургер». Вона дублює основні розділи системи, забезпечуючи зручну навігацію та швидкий доступ до ключових сторінок магазину. На головній сторінці також представлено баннер із актуальними новинами та акціями, а нижче розміщено блок із найпопулярнішими товарами.

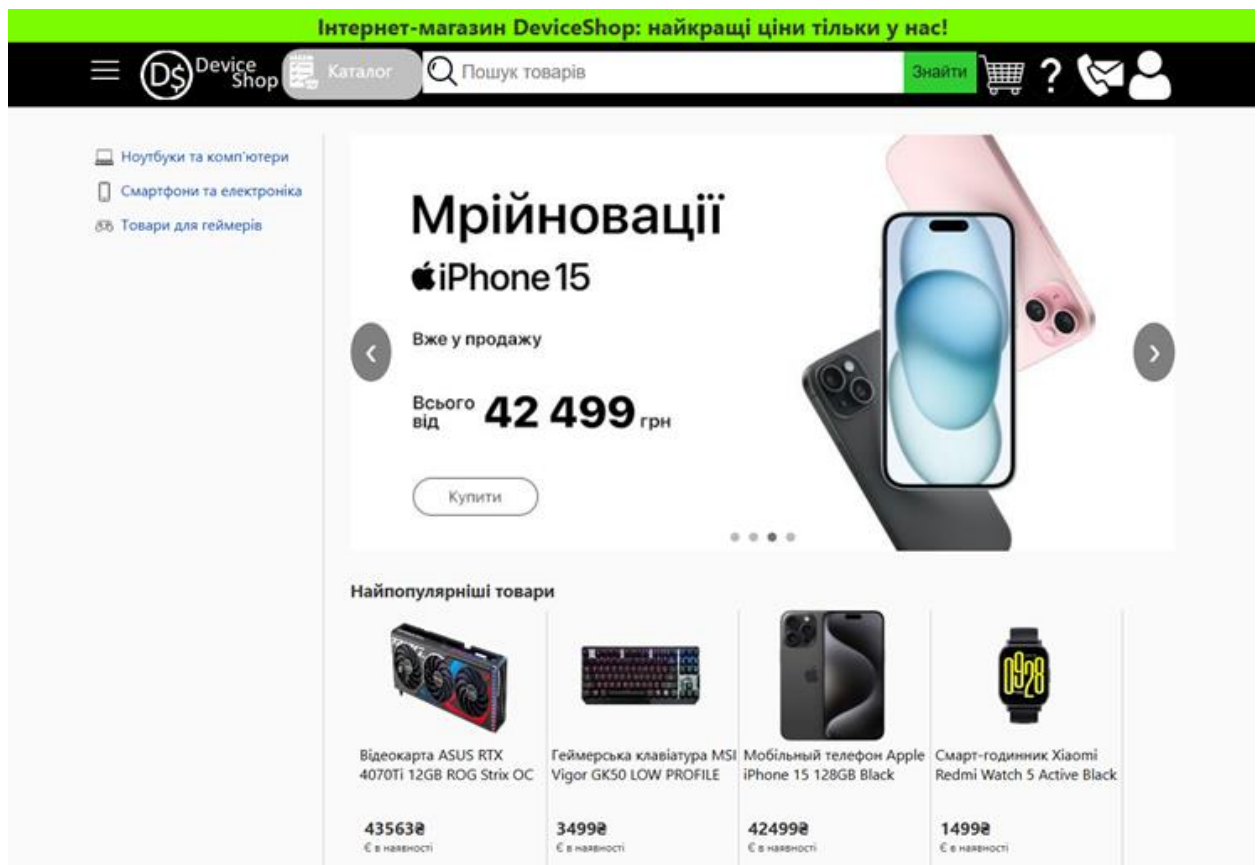


Рисунок 3.5 – Головна сторінка вебсистеми

Для отримання доступу до персоналізованого функціоналу, такого як редагування профілю та взаємодії з менеджерами, користувачеві необхідно пройти процес аутентифікації або реєстрації. При натисканні на іконку профілю у правій частині меню з'являється модальне вікно, яке дозволяє

перемикатися між формами входу та створення нового облікового запису без перезавантаження сторінки. Під час реєстрації система вимагає введення логіна та пароля з обов'язковим підтвердженням. Під час введення даних у формах реалізовано базову валідацію, що запобігає надсиланню некоректної інформації. Після успішного входу користувач отримує доступ до особистого профілю (рис. 3.6), де він може переглядати свої дані, змінювати ім'я та дату народження через спеціальні діалогові вікна редагування. Окрім базової інформації, у профілі передбачено механізми зміни пароля та видалення облікового запису, що забезпечує користувачу повний контроль над власними конфіденційними даними. Також за потреби, користувач може вийти з акаунту натисканням кнопки.

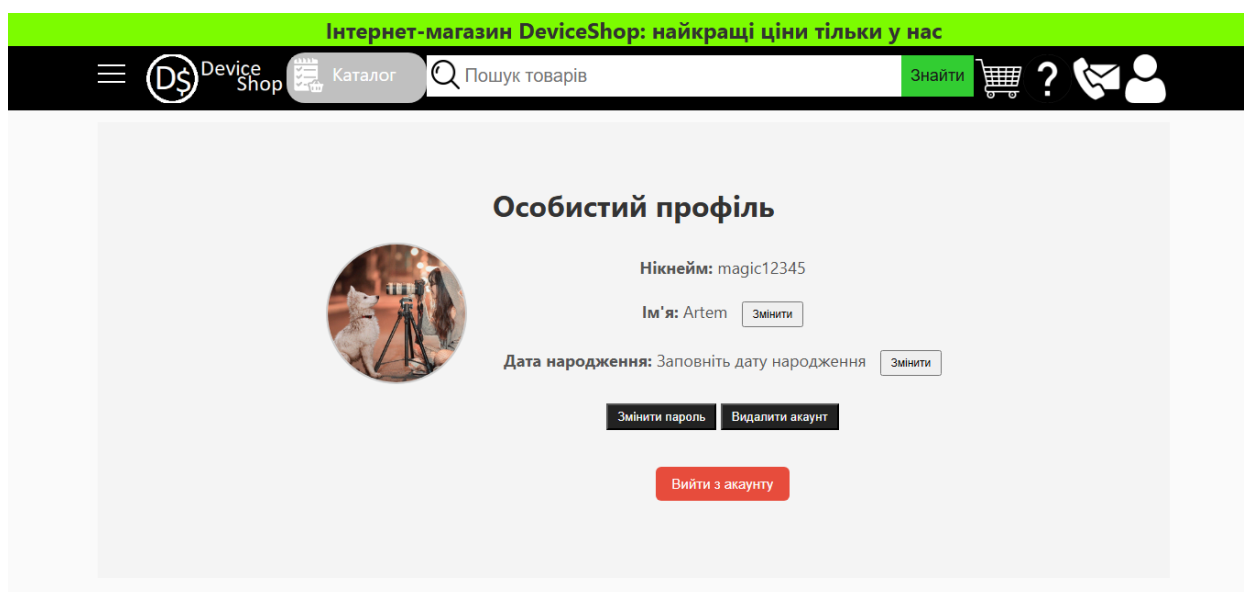


Рисунок 3.6 – Особистий профіль користувача

Пошук необхідної техніки здійснюється двома шляхами: через інтерактивний рядок пошуку в головному меню сайту або через структурований каталог товарів. Каталог розподілений за категоріями, такими як «Ноутбуки та комп'ютери», «Смартфони та електроніка» та «Товари для геймерів» (рис. 3.7), кожна з яких має власні підкатегорії. При виборі конкретного товару користувач переходить на сторінку з детальною інформацією (рис. 3.8, 3.9). Тут реалізовано зручну систему навігації за

допомогою «хлібних крихт», що дозволяє легко повернутися на попередній рівень категорій, слайдер із зображеннями пристрою, назва пристрою, блок із технічними характеристиками, ціну, детальний опис та інформацію про умови оплати й повернення. Для додавання обраної позиції до замовлення необхідно натиснути кнопку «Купити», після чого дані про товар автоматично передаються до системи кошика.



Рисунок 3.7 – Сторінка товару з основними характеристиками, зображеннями, ціною та інформацією про оплату і повернення

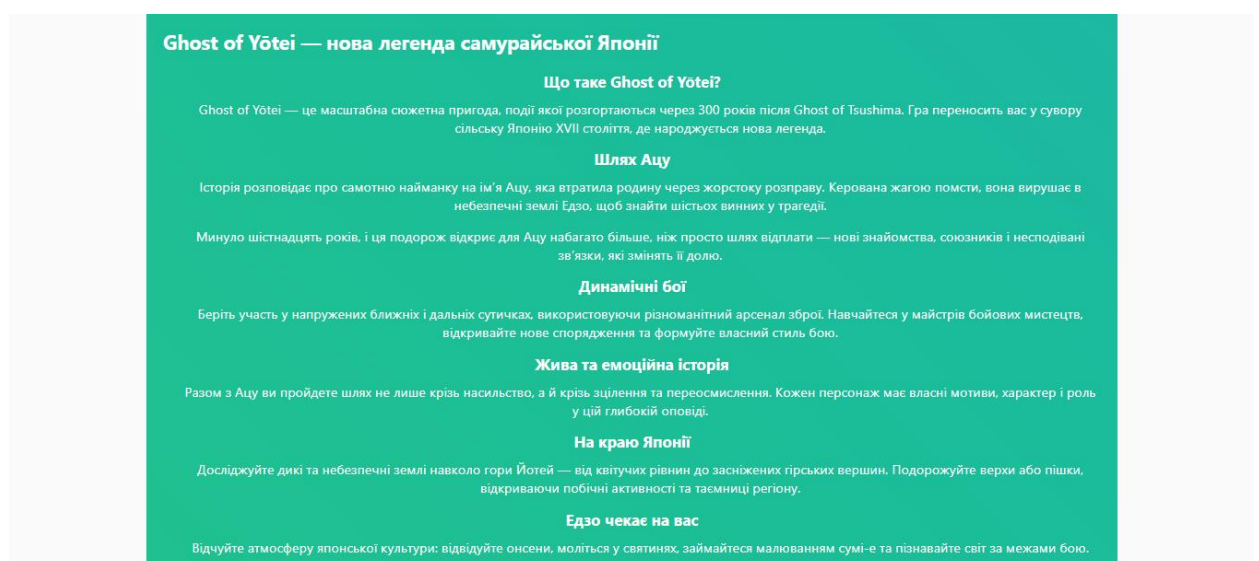


Рисунок 3.8 – Детальний опис на сторінці товару

| Характеристики          |                                      |
|-------------------------|--------------------------------------|
| Платформа               | PlayStation 5                        |
| Жанр                    | Action, Adventure                    |
| Тематика                | Про самураїв                         |
| Кількість гравців       | 1                                    |
| Вікові обмеження        | 18+                                  |
| Локалізація             | Англійська версія                    |
| Тип видання             | Стандартне                           |
| Носій                   | Blu-ray                              |
| Упаковка                | Blu-ray Box                          |
| Підтримуваний контролер | Геймпад                              |
| Особливості             | Відкритий світ                       |
| Дата світового релізу   | 02.10.2025                           |
| Видавець                | SIE (Sony Interactive Entertainment) |
| Розробник               | Sucker Punch Productions             |
| Країна-виробник         | Австрія                              |

Рисунок 3.9 – Блок з повними характеристиками товару

Керування обраними товарами відбувається на сторінці кошика, яка представлена у вигляді таблиці (рис. 3.10). Користувач може візуально оцінити свій вибір, перевірити ціну кожної позиції та змінити кількість одиниць товару безпосередньо в полі вводу, при цьому загальна сума замовлення перераховується динамічно. Також передбачена можливість видалення окремих товарів із переліку. Після перевірки всіх даних користувач переходить до фінального етапу – оформлення замовлення за допомогою відповідної кнопки, яка спрямовує його до форми заповнення контактними даними, де потрібно вписати ім'я, електронну адресу, номер телефона, адресу доставки та вибрати спосіб оплати.

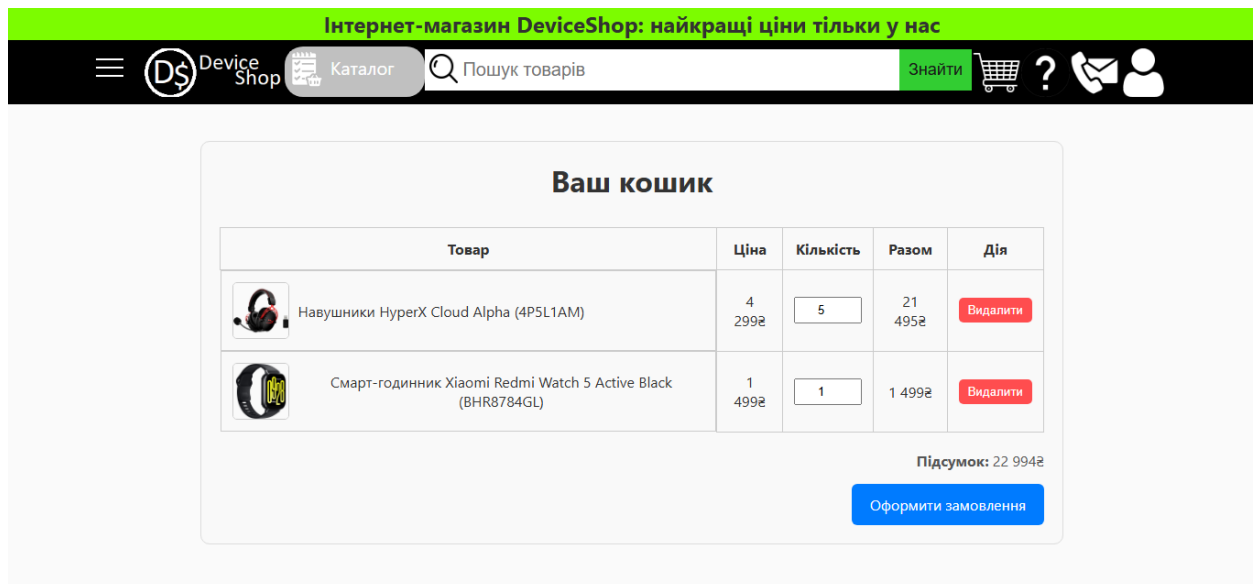


Рисунок 3.10 – Інтерфейс кошика користувача з переліком обраних товарів

Для вирішення проблемних питань або отримання додаткової інформації про роботу магазину користувач може скористатися розділами «Довідниковий центр» та «Зв'язатися з нами». У довідниковому центрі розміщено відповіді на типові запитання та форма зворотного зв'язку для зв'язку з адміністрацією, а сторінка контактів містить номери телефонів, електронні адреси та графік роботи інтернет-магазину. Це забезпечує повноцінну підтримку клієнта на всіх етапах перебування на сайті, роблячи процес використання вебсистеми комфортним та безпечним.

## **4 ТЕСТУВАННЯ ВЕБСИСТЕМИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ**

### **4.1 Розробка стратегії тестування, чеклістів та тест-кейсів**

Процес тестування розробленої вебсистеми є критично важливим етапом життєвого циклу розробки, оскільки він дозволяє не лише знайти помилки в коді, а й переконатися, що продукт відповідає всім визначеним вимогам. Основна мета полягає у верифікації та валідації системи, тобто перевірці того, чи правильно реалізовано функції та чи працюють вони так, як очікує майбутній користувач [25]. Специфіка вебдодатків додає складності через клієнт-серверну архітектуру, роботу з великою кількістю одночасних запитів та необхідність захисту конфіденційних даних користувачів. Тому тестування зосереджується на стабільності роботи системи, її здатності коректно обробляти вхідні дані та забезпечувати зручну взаємодію з інтерфейсом у різних умовах експлуатації [26]. Недостатня увага до тестування часто призводить до відтоку користувачів через баги, тому якісний контроль є критично важливим для успіху онлайн-бізнесу [27].

Для ефективної перевірки вебсистеми було розроблено стратегію тестування, яка визначає загальний план дій, необхідні ресурси та методи контролю якості. У сучасній практиці виділяють два основні підходи: ручне та автоматизоване тестування [28]. Автоматизоване тестування є ефективним для повторюваних завдань або великих проєктів, проте вона вимагає значних часових витрат на написання скриптів та налаштування середовища. Для даного навчального проєкту було обрано стратегію ручного тестування, оскільки вона дозволяє більш гнучко підійти до перевірки інтерфейсу, оцінити зручність користування та виявити специфічні візуальні дефекти, які важко відловити автоматикою. Окрім цього, ручний підхід є економічно виправданим для індивідуальної розробки, оскільки дозволяє тестувальнику імітувати реальну поведінку покупця в магазині, використовуючи власні сили для пошуку неочевидних помилок [29].

В межах обраної стратегії застосовується декілька видів тестування, кожен з яких закриває певну область роботи системи. Функціональне тестування є базовим і спрямоване на перевірку відповідності роботи системи заданим вимогам та коректності реалізації основних функцій, таких як реєстрація користувача, авторизація, пошук товарів, додавання до кошика та оформлення замовлення [26]. Окрему увагу приділено юзабіліті-тестуванню, яке допомагає оцінити зручність користування інтерфейсом, інтуїтивність навігації та легкість виконання цільових дій. Також важливим стало кросбраузерне тестування та перевірка верстки, які передбачають перевірку коректності відображення вебсистеми у різних браузерах та на різних пристроях. Це дозволило впевнитися, що сайт однаково коректно відображається як у Google Chrome на Windows 10, так і в Safari на MacBook, зберігаючи пропорції елементів та доступність функцій. Тестування безпеки в даній роботі проводилося в межах локального середовища (через вбудований сервер Laravel), де перевірялася стійкість форм до некоректного вводу даних та надійність механізмів аутентифікації.

Для систематизації процесу перевірки спочатку було сформовано чекліст, який є спрощеною формою тестової документації. Чекліст являє собою перелік конкретних функцій або сторінок сайту, які потребують обов'язкової перевірки. Це допомагає не пропустити важливі блоки системи під час повторних перевірок та швидко зафіксувати статус виконання робіт [30]. Для забезпечення повноти перевірки було розроблено розгорнутий чекліст, що охоплює всі модулі вебсистеми: від управління профілем до процесу взаємодії з кошиком. Через великий обсяг документації, у даному розділі наведено лише фрагмент розробленого чеклісту (рис. 4.1), тоді як повна версія переліку вже пройдених перевірок представлена у Додатку А. Сама структура документа передбачає відповідні поля для подальшої фіксації статусу проходження кожної перевірки («Passed» – пройдено, «Failed» – помилка), що дозволяє наочно оцінити готовність системи.

| №                   | Категорія        | Перевірка                                 | Очікуваний результат                                                          | Браузер Google Chrome<br>v. 140.0.7339.128 | Браузер Mozilla Firefox<br>v. 148.0 (64-bit) | Браузер Safari<br>v. 13.1.2 | Коментар/Баг репорт |
|---------------------|------------------|-------------------------------------------|-------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|-----------------------------|---------------------|
| 1. Функціональність |                  |                                           |                                                                               |                                            |                                              |                             |                     |
| 1.1                 | Функціональність | Запуск проекту через<br>php artisan serve | Проект запускається<br>без помилок                                            |                                            |                                              |                             |                     |
| 1.2                 | Функціональність | Перевірити реєстрацію<br>на сайті         | Реєстрація успішна -<br>відкривається профіль<br>та дані зберігаються в<br>БД |                                            |                                              |                             |                     |
| 1.3                 | Функціональність | Перевірити авторизацію<br>на сайті        | Відбувається вхід у<br>систему та<br>відкривається<br>профіль користувача     |                                            |                                              |                             |                     |
| 1.4                 | Функціональність | Перевірити вихід з<br>акаунту             | Сесія завершується і<br>відбувається перехід<br>на головну сторінку           |                                            |                                              |                             |                     |
| 1.5                 | Функціональність | Перехід між сторінками                    | Всі маршрути працюють<br>без 404                                              |                                            |                                              |                             |                     |
| 1.6                 | Функціональність | Пошук товарів за назвою                   | Відображаються<br>релевантні<br>результати                                    |                                            |                                              |                             |                     |
| 1.7                 | Функціональність | Додавання товару в кошик                  | Товар з'являється в<br>кошику та виводиться<br>візуальне<br>підтвердження     |                                            |                                              |                             |                     |

Рисунок 4.1 – Фрагмент чеклісту переліку перевірок функціональних можливостей вебсистеми

Окрім чеклісту, було розроблено тест-кейси, які описують конкретні сценарії взаємодії користувача з системою [31]. Кожен тест-кейс – це професійна ситуація, що має чітку послідовність дій для досягнення певного результату. При їх складанні важливо дотримуватися «золотої середини»: опис не повинен бути занадто коротким, щоб уникнути непорозумінь, але й не має містити зайвої другорядної інформації, яка б заважала проходженню тесту [32]. Стандартна структура тест-кейса, застосована у роботі, включає унікальний ідентифікатор (ID), лаконічну назву і тему, передумови, які готують систему до тесту, покрокову інструкцію виконання та очікуваний результат. Окремим аспектом при формуванні тест-кейсів стала розробка як позитивних сценаріїв, що базуються на використанні валідних даних для підтвердження стандартної роботи функцій, так і негативних сценаріїв, які передбачають введення некоректної інформації. Такий підхід дозволяє ще на етапі планування закласти умови для перевірки стійкості вебсистеми до помилок та впевнитися, що система здатна правильно реагувати на нестандартні дії користувача (рис. 4.2 – 4.5). Назва тест-кейса формулювалася максимально чітко, щоб будь-який інший фахівець міг зрозуміти мету перевірки без додаткових пояснень [31]. Таке

документування дозволяє перетворити абстрактні вимоги на конкретні сценарії, що гарантує цілісну та логічно завершену перевірку кожної функції вебсистеми.

| ID                                                                                                                                                                                                                                                   |                                                                                  | REG_001                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|------------------------------------------------------------------------|
| Назва                                                                                                                                                                                                                                                |                                                                                  | Реєстрація користувача з валідними даними                              |
| Тема:                                                                                                                                                                                                                                                |                                                                                  |                                                                        |
| <i>Перевірка роботи системи реєстрації користувача при введенні валідних даних</i>                                                                                                                                                                   |                                                                                  |                                                                        |
| Передумови                                                                                                                                                                                                                                           |                                                                                  |                                                                        |
| 1. Користувач не реєструвався, має валідні дані для входу<br>2. Відкрити сайт DeviceShop<br>3. Натиснути на іконку профіля на головній сторінці<br>4. Натиснути на кнопку "Зареєструватися"<br>5. Відкрита форма реєстрації користувача "Реєстрація" |                                                                                  |                                                                        |
| №                                                                                                                                                                                                                                                    | Кроки                                                                            | Очікуваний результат                                                   |
| 1                                                                                                                                                                                                                                                    | Ввести у полі Логін валідні дані                                                 | В поле Логін внесено дані                                              |
| 2                                                                                                                                                                                                                                                    | Ввести у полі Пароль валідний пароль(6 і більше символів)                        | Введено валідний пароль, який приховується крапками                    |
| 3                                                                                                                                                                                                                                                    | Ввести у полі Підтвердження пароля той пароль, який вводили в попередньому кроці | Введено валідний пароль, пароль приховується крапками                  |
| 4                                                                                                                                                                                                                                                    | Натиснути на кнопку "Зареєструватися"                                            | Акаунт успішно створений та користувача перенесено на сторінку профіля |

Рисунок 4.2 – Тест-кейс реєстрації користувача з валідними даними

| ID                                                                                                                                                                                                                                                   |                                                                                                     | REG_002                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Назва                                                                                                                                                                                                                                                |                                                                                                     | Реєстрація користувача з невалідними даними                                     |
| Тема:                                                                                                                                                                                                                                                |                                                                                                     |                                                                                 |
| <i>Перевірка роботи системи реєстрації користувача при введенні невалідних даних</i>                                                                                                                                                                 |                                                                                                     |                                                                                 |
| Передумови                                                                                                                                                                                                                                           |                                                                                                     |                                                                                 |
| 1. Користувач не реєструвався, має валідні дані для входу<br>2. Відкрити сайт DeviceShop<br>3. Натиснути на іконку профіля на головній сторінці<br>4. Натиснути на кнопку "Зареєструватися"<br>5. Відкрита форма реєстрації користувача "Реєстрація" |                                                                                                     |                                                                                 |
| №                                                                                                                                                                                                                                                    | Кроки                                                                                               | Очікуваний результат                                                            |
| 1                                                                                                                                                                                                                                                    | Ввести у полі Логін невалідні дані                                                                  | В поле Логін внесено дані                                                       |
| 2                                                                                                                                                                                                                                                    | Ввести у полі Пароль невалідний пароль(менше 6 символів)                                            | Введено пароль, який приховується крапками                                      |
| 3                                                                                                                                                                                                                                                    | Ввести у полі Підтвердження пароля інший пароль, який відрізняється від пароля в попередньому кроці | Введено пароль, пароль приховується крапками                                    |
| 4                                                                                                                                                                                                                                                    | Натиснути на кнопку "Зареєструватися"                                                               | Система не переходить до наступного кроку. Виводяться повідомлення про помилки. |

Рисунок 4.3 – Тест-кейс реєстрації користувача з невалідними даними

|                                                                                                                                                                                                |                                                           |                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|------------------------------------------------|
| ID                                                                                                                                                                                             | AUTH_001                                                  |                                                |
| Назва                                                                                                                                                                                          | Авторизація користувача з валідними даними                |                                                |
| Тема:                                                                                                                                                                                          |                                                           |                                                |
| Перевірка входу користувача до системи при введенні валідних облікових даних                                                                                                                   |                                                           |                                                |
| Передумови                                                                                                                                                                                     |                                                           |                                                |
| 1. Користувач попередньо зареєстрований у системі.<br>2. Відкрити сайт DeviceShop.<br>3. Натиснути на іконку профіля на головній сторінці.<br>4. Відкрита форма авторизації користувача "Вхід" |                                                           |                                                |
| №                                                                                                                                                                                              | Кроки                                                     | Очікуваний результат                           |
| 1                                                                                                                                                                                              | Ввести у полі Логін валідні дані                          | В поле Логін внесено валідні дані              |
| 2                                                                                                                                                                                              | Ввести у полі Пароль валідний пароль(6 символів і більше) | Введено пароль, який приховується крапками     |
| 3                                                                                                                                                                                              | Натиснути на кнопку "Увійти"                              | Користувач перенаправлений на сторінку профіля |

Рисунок 4.4 – Тест-кейс авторизації користувача з валідними даними

|                                                                                                                                                                                                |                                                          |                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|---------------------------------------------------------|
| ID                                                                                                                                                                                             | AUTH_002                                                 |                                                         |
| Назва                                                                                                                                                                                          | Авторизація користувача з невалідними даними             |                                                         |
| Тема:                                                                                                                                                                                          |                                                          |                                                         |
| Перевірка входу користувача до системи при введенні невалідних облікових даних                                                                                                                 |                                                          |                                                         |
| Передумови                                                                                                                                                                                     |                                                          |                                                         |
| 1. Користувач попередньо зареєстрований у системі.<br>2. Відкрити сайт DeviceShop.<br>3. Натиснути на іконку профіля на головній сторінці.<br>4. Відкрита форма авторизації користувача "Вхід" |                                                          |                                                         |
| №                                                                                                                                                                                              | Кроки                                                    | Очікуваний результат                                    |
| 1                                                                                                                                                                                              | Ввести у полі Логін невалідні дані                       | В поле Логін внесено дані                               |
| 2                                                                                                                                                                                              | Ввести у полі Пароль невалідний пароль(менше 6 символів) | Введено пароль, який приховується крапками              |
| 3                                                                                                                                                                                              | Натиснути на кнопку "Увійти"                             | Відображається повідомлення "Невірний логін або пароль" |

Рисунок 4.5 – Тест-кейс авторизації користувача з невалідними даними

## 4.2 Проведення тестування та фіксація дефектів у баг-репортах

Наступним кроком після підготовки тестової документації стало безпосереднє виконання перевірок. Процес тестування відбувався в умовах, максимально наближених до реального використання вебсистеми та проходив у локальному середовищі з використанням різних браузерів. Оскільки сучасні вебсайти базуються на клієнт-серверній архітектурі, користувачі отримують доступ до них через велику кількість платформ і

пристроїв, що значно ускладнює забезпечення ідентичного відображення контенту в кожній системі [33]. Це зумовлено використанням різними браузерами специфічних рушіїв рендерингу (Blink, WebKit, Gecko), які по-різному обробляють HTML та CSS-код [34]. Також відмінності в обробці JavaScript та вебстандартів часто створюють конфлікти сумісності [35]. Тому коректна кросбраузерна робота є не лише технічною вимогою, а й базовою умовою для зручної взаємодії користувача з інтерфейсом [36]. Основна увага приділялася функціональним можливостям інтернет-магазину та коректності відображення інтерфейсу в різних браузерах. Якість програмного продукту та задоволеність майбутніх користувачів прямо залежать від ефективності заходів із забезпечення якості (SQA), які дозволяють вчасно виявити та усунути помилки [37].

Першим кроком було виконання перевірок згідно з раніше розробленим чеклістом. Під час тестування було зафіксовано результати для кожного пункту, що дозволило отримати цілісну картину готовності системи. Результати виконання чеклісту наведено у Додатку А.

Більшість функціональних блоків, таких як «Авторизація», «Персональний кабінет» та «Кошик», пройшли перевірку успішно (статус «Passed»), що свідчить про коректну роботу логіки системи. Однак, під час кросбраузерного тестування верстки було виявлено дефект (пункт 2.4 чеклісту), що отримав статус «Failed». Цей дефект стосується некоректного відображення елементів інтерфейсу в браузері Safari. Це підкреслює важливість кросбраузерних перевірок, оскільки через різницю в інтерпретації вихідного коду та неоднакову підтримку вебстандартів різними браузерами виникають дефекти сумісності, які можуть призвести до розривів макета або некоректної роботи функцій [33], [34].

Для того, щоб розробник міг зрозуміти суть проблеми та виправити її, кожен знайдений дефект необхідно задокументувати у вигляді баг-репорта. Баг-репорт – це технічний документ, який чітко описує умови, за яких виникає помилка, та очікувану поведінку системи [38]. Правильно

оформлений звіт дозволяє уникнути непорозумінь між тестувальником та програмістом.

При написанні баг-репорта у даній роботі було дотримано стандартної структури, яка включає такі ключові атрибути [38]:

- тема: чітко вказано, що саме не так, де саме та коли це відбувається;
- кроки для відтворення: покроковий опис дій, який дозволяє будь-кому повторити ситуацію;
- фактичний та очікуваний результат: порівняння реального стану системи (іконки деформовані) з тим, як вони мали б виглядати;
- серйозність та пріоритет: оцінка важливості виправлення помилки та її впливу на систему;
- оточення: вказання операційної системи та браузера;
- вкладення: візуальні докази, які полегшують аналіз проблеми.

Приклад оформленого баг-репорта для виявленого дефекту в браузері Safari наведено на рис. 4.6.

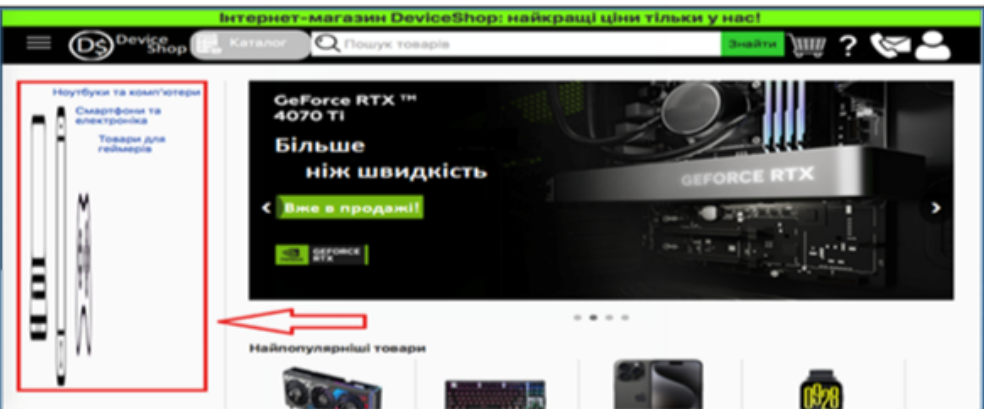
| Номер/ID              | Проект                                                                                                                                                                                                                                                                                                                 | Категорія          | Рівень доступу | Дата реєстрації  | Дата зміни       |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|----------------|------------------|------------------|
| №001                  | Інтернет-магазин DeviceShop                                                                                                                                                                                                                                                                                            | Верстка            | Загальний      | 18.03.2026 16:25 | 18.03.2026 16:58 |
| Тестувальник          | Гордієнко Артем КНТ-222                                                                                                                                                                                                                                                                                                | Пріоритет          | Середній       | Серйозність      | Низька           |
| Відтворюваність       | Постійно                                                                                                                                                                                                                                                                                                               | Статус             | Новий          | Резолюція        | Відкрита         |
| Платформа             | Safari v. 13.1.2                                                                                                                                                                                                                                                                                                       | Операційна система | macOS          | Версія ОС        | 10.15 Catalina   |
| Тема                  | Некоректне відображення іконок у блоці категорій на головній сторінці в Safari                                                                                                                                                                                                                                         |                    |                |                  |                  |
| Опис                  | При перегляді головної сторінки в Safari іконки в боковому меню категорій відображаються некоректно                                                                                                                                                                                                                    |                    |                |                  |                  |
| Кроки для відтворення | 1. Запустити локальний сервер командою: <code>php artisan serve --host=0.0.0.0 --port=8000</code> .<br>2. Відкрити браузер Safari на пристрої MacBook.<br>3. Перейти за адресою: <code>http://192.168.0.154:8000</code> .<br>4. Звернути увагу на відображення іконок у боковому меню категорій з лівого боку сторінки |                    |                |                  |                  |
| Фактичний результат   | Іконки категорій відображаються некоректно: розтягнуті по вертикалі                                                                                                                                                                                                                                                    |                    |                |                  |                  |
| Очікуваний результат  | Іконки відображаються коректно: збережені пропорції, вирівняні відносно тексту та не виходять за межі контейнера                                                                                                                                                                                                       |                    |                |                  |                  |
| Додаткова інформація  |                                                                                                                                                                                                                                    |                    |                |                  |                  |

Рисунок 4.6 – Баг-репорт №001: «Розтягування іконок категорій у браузері Safari»

У зафіксованому дефекті серйозність була визначена як «Незначна», а пріоритет як «Середній», оскільки помилка є візуальною і не блокує можливість здійснення покупок, проте негативно впливає на загальне враження користувача від інтерфейсу.

Загалом, проведення тестування та ведення чіткої звітності дозволило не лише знайти слабкі місця в інтерфейсі, а й систематизувати процес їх усунення. Це підтверджує, що регулярні заходи з контролю якості (SQA) є необхідними для створення надійних вебзастосунків у сучасному бізнес-середовищі [37].

#### 4.3 Аналіз результатів тестування та оцінка якості продукту

Підсумковим етапом розробки стало оцінювання отриманих результатів тестування та визначення ступеня готовності вебсистеми до експлуатації. Аналіз базувався на кількісних показниках виконання чекліста та якісній оцінці виявлених дефектів.

Згідно з результатами, наведеними у Додатку А, було перевірено 42 пункти, з яких 41 (97,62%) отримали статус «Passed». Виявлені невідповідності (2,38%) не мають критичного впливу на бізнес-логіку системи. Співвідношення успішних та неуспішних перевірок візуалізовано на рис. 4.7.

Співвідношення успішних та неуспішних перевірок

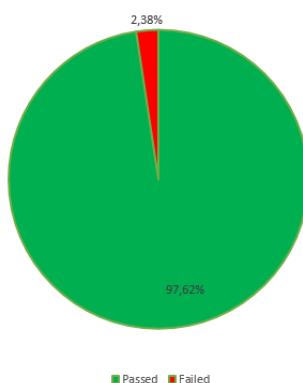


Рисунок 4.7 – Співвідношення успішно пройдених та провалених тестів

Детальний аналіз показав, що основна частина функціоналу, яка відповідає за роботу з динамічними сторінками, кошиком, персональним кабінетом та базою даних, працює без зауважень. Це є критично важливим, оскільки для систем електронної комерції надійність та коректність обробки запитів є пріоритетними характеристиками [39].

Єдиний виявлений дефект (пункт 2.4 чекліста), що стосувався розтягування іконок у браузері Safari, був класифікований як дефект відображення. Оскільки доступ до сучасних вебсайтів здійснюється через велику кількість браузерів, такі нюанси рендерингу є очікуваними через обмеження гнучкості програмування в середовищі WWW [39]. Після фіксації цього дефекту в баг-репорті, помилку було оперативно усунуто шляхом коригування CSS-стилів контейнерів. Повторна перевірка підтвердила коректне відображення елементів у всіх цільових браузерах.

Оцінюючи якість продукту з точки зору зручності (Usability), було враховано, що успіх інтернет-магазину прямо залежить від простоти взаємодії користувача з інтерфейсом. Аналіз проводився за такими критеріями [40]:

- ефективність: користувачі можуть швидко знаходити потрібні товари та додавати їх до кошика;
- зрозумілість: структура навігації є логічною, що мінімізує зусилля на вивчення системи при першому відвідуванні;
- задоволеність: візуальне оформлення та відсутність помилок на критичних етапах (наприклад, при авторизації) формують позитивний досвід користувача.

Завдяки використанню сучасного фреймворку та оптимізованої архітектури вдалося забезпечити стабільну роботу вебсистеми. Високий відсоток успішно пройдених тестів свідчить про те, що архітектура вебсистеми є цілісною, а програмна логіка – захищеною від типових збоїв.

Для об'єктивної оцінки ефективності розроблених рішень було проведено порівняльний аналіз швидкодії розробленої вебсистеми. В межах

аналізу було здійснено порівняння часу відгуку вебсистеми DeviceShop із раніше розглянутими аналогами за двома критичними операціями: завантаженням динамічного контенту кошика та обробкою запиту на зміну персональної інформації. Результати порівняльного аналізу наведено в таблиці 4.1.

Таблиця 4.1 – Порівняльний аналіз часу відгуку вебсистем (у секундах)

| Операція користувача           | ROZETKA | COMFY | TELEMART | DeviceShop |
|--------------------------------|---------|-------|----------|------------|
| Відкриття кошика з товарами    | 4,03    | 13,16 | 2,52     | 1,85       |
| Редагування персональних даних | 1,73    | 1,59  | 1,67     | 1,32       |

Як видно з даних табл. 4.1, розроблена система демонструє кращі показники швидкодії. Зокрема, час відкриття кошика у DeviceShop у 1,36 раза менший за найближчий за показниками аналог – Telemart та у 7,1 раза швидший за систему Comfy. Таких результатів вдалося досягти завдяки мінімізації зовнішніх запитів, відсутності надлишкових аналітичних скриптів, використанню продуктивної СКБД PostgreSQL та оптимізованих конфігураційних файлів для зберігання структури каталогу товарів.

Отже, на основі проведеного аналізу можна зробити висновок, що розроблена вебсистема для електронної комерції повністю відповідає встановленим вимогам якості. Продукт є працездатним, зручним у використанні та може слугувати надійною базою для подальшого масштабування або адаптації під реальні бізнес-процеси. Ознайомитися з роботою вебсистеми та протестувати її функціонал у режимі реального часу можна вже зараз [41].

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено функціональну вебсистему для електронної комерції та реалізовано комплекс заходів із забезпечення її якості. На основі проведеного аналізу ринку було сформовано перелік вимог, що стали фундаментом для створення надійного програмного продукту.

Головної мети проєкту було досягнуто шляхом детального аналізу вимог та проєктування гнучкої архітектури:

- скорочення часу обробки користувацьких запитів на перегляд товарів в кошику у 1,36-7,1 рази за рахунок використання оптимізованих конфігураційних структур даних;
- стабільна швидкість виконання запитів та скорочення часу обробки персональної інформації користувачів в 1,3 рази завдяки використанню вбудованих методів Laravel та оптимізованих взаємодій з БД PostgreSQL;
- збільшення захищеності від типових вразливостей (SQL-ін'єкцій та XSS-атак) до 95% за рахунок автоматичного екранування даних у Blade-шаблонах та використання підготовлених запитів в Eloquent ORM;
- зручність супроводу програмного коду: скорочення об'єму коду в 2 рази завдяки використанню архітектури MVC та вбудованих функцій фреймворку.

Технологічний стек на основі PHP, фреймворку Laravel та СКБД PostgreSQL дозволив реалізувати архітектурний патерн MVC, що забезпечило чітке розділення логіки та інтерфейсу.

Проведене комплексне тестування підтвердило повну відповідність системи встановленим вимогам з рівнем надійності 97,62% успішно пройдених пунктів чекліста. Своєчасне виявлення та усунення дефектів кросбраузерної сумісності дозволило отримати цілісний продукт, який демонструє стабільну роботу, має інтуїтивний інтерфейс та володіє потенціалом для подальшого масштабування під реальні потреби бізнесу.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Марусей Т. В. Основні тенденції розвитку ринку електронної комерції в Україні. *Економіка та суспільство*. 2018. № 14. С. 1011-1015.
2. Різниця між електронною комерцією та електронним бізнесом. *WEZOM*. URL: <https://wezom.com.ua/ua/blog/raznitsa-mezhdu-elektronnoj-kommertsiej-i-elektronnym-biznesom> (дата звернення: 14.04.2026).
3. Капітанюк Е. В. Аналіз поточного розвитку e-commerce в українському середовищі. *Економічний простір*. 2025. № 208. С. 26-31. URL: <https://doi.org/10.30838/EP.208.26-31> (дата звернення: 14.04.2026).
4. Антонюк Т. Українці за 2025 рік витратили на онлайн-покупки 256 млрд грн. *Forbes*. URL: <https://forbes.ua/news/ukraintsi-vitratili-na-onlayn-pokupki-256-mlrd-grn-prom-21012026-35702> (дата звернення: 14.04.2026).
5. Іллященко С. М., Іванова Т. Є. Перспективи та основні проблеми розвитку інтернет-торгівлі в Україні. *Механізм регулювання економіки*. 2014. №3. С. 72-81.
6. Як створити веб-додаток: типи, переваги, принцип роботи. *WEZOM*. URL: <https://wezom.com.ua/ua/blog/kak-sozdat-veb-prilozhenie> (дата звернення: 14.04.2026).
7. Вербицький Я. С. Психологічні фактори в поведінці споживачів електронної комерції. *Економіка та управління підприємствами*. 2023. № 1. С. 47-54. DOI: 10.36919/2312-7812.1.2023.47 (дата звернення: 14.04.2026).
8. Ahmad F., Asmadi S. E., Janom N. E-commerce requirement analysis and design with Fogg Behavior Model (FBM). *Malaysian Journal of Computing*. 2025. Vol. 10, no. 2. P. 2219-2233.
9. Що таке функціональне тестування? Типи, приклади, контрольний список і впровадження. *ZAPTEST*. URL: <https://www.zaptest.com/uk/%D1%89%D0%BE-%D1%82%D0%B0%D0%BA%D0%B5-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D0%BE%D0%BD%>

D0%B0%D0%BB%D1%8C%D0%BD%D0%B5-  
 %D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%  
 D0%BD%D1%8F-%D1%82%D0%B8%D0%BF (дата звернення: 15.04.2026).

10. Onyemaobi C. B., Ajah I. A. Comparative analysis of web development languages performances. *Int. J. Web Science*. 2017. Vol. 3, no. 1. P. 16-31.

11. Patel S. I. A Comparative Study of PHP Frameworks Performance. *The International Journal of Multidisciplinary Research*. 2024. Vol. 10, no. 1. P. 1-10.

12. Parray H. F., Bansal K. Laravel Framework as the Best Framework for PHP based Web Development of Small Medium, Large and E-Commerce Web Applications. *International Journal of Electrical and Computer Engineering*. 2021. P. 1-7.

13. Kumar V., Patil D., Maurya N. A Study of Attack on PHP and Web Security. *Communications on Applied Electronics (CAE)*. 2015. Vol. 1, no. 4. P. 1-13.

14. Subecz Z. Web-development with Laravel framework. *Gradus*. 2021. Vol. 8, no. 1. P. 211-218. URL: <https://doi.org/10.47833/2021.1.CSC.006> (date of access: 15.04.2026).

15. Що таке бази даних?. *Sigma Software University*. URL: <https://university.sigma.software/what-is-database/> (дата звернення: 16.04.2026).

16. Лебединський А., Черкасов М. Аналіз сучасних систем управління базами даних. *Комп'ютерні технології і мехатроніка*. 2023. С. 32-36.

17. PostgreSQL чи MySQL. Як обрати оптимальну базу даних для вашого проєкту. *DOU*. URL: <https://dou.ua/forums/topic/51621/> (дата звернення: 16.04.2026).

18. База даних PostgreSQL: інформація та короткий гайд. *itProger*. URL: <https://itproger.com/ua/news/baza-dannih-postgresql-informatsiya-i-kratkiy-gayd> (дата звернення: 16.04.2026).

19. Schneider M., Martínez D. A Comparative Benchmark Analysis of Transactional and Analytical Performance in PostgreSQL and MySQL. *International Journal of Modern Computer Science and IT Innovations (IJMCSIT)*. 2025. Vol. 2, no. 10. P. 51-63.
20. Клієнт-серверна архітектура. *QATestLab*. URL: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/> (дата звернення: 17.04.2026).
21. Розуміння Клієнт-Серверної Архітектури на прикладах. *DOU*. URL: <https://dou.ua/forums/topic/44636/> (дата звернення: 17.04.2026).
22. Що таке фронтенд і бекенд-розробка: різниця, особливості та взаємодія. *BRANDER*. URL: <https://brander.ua/blog/shcho-take-frontend-i-bekend-rozrobka> (дата звернення: 17.04.2026).
23. Majeed A., Rauf I. MVC Architecture: A Detailed Insight to the Modern Web Applications Development. *Peer Review Journal of Solar & Photoenergy Systems*. 2018. Vol. 1, no. 1. P. 1-7.
24. Не соромно запитати: що таке вайрфрейм, мокап і прототип. *Skvot*. URL: <https://skvot.io/uk/blog/ne-soromno-zapitati-shcho-take-vayrfreym-mokap-i-prototip> (дата звернення: 18.04.2026).
25. Conference Proceedings: SAS Users Group International 30 April 10-13, 2005, Philadelphia, Pennsylvania, 141-30 paper.
26. Sabharwal S., Sibal R., Sharma C. A Survey of Testing Techniques for Testing Web based Applications. *International Journal Web Applications*. 2015. Vol. 7, no. 2. P. 43-55.
27. Все що потрібно знати про тестування сайтів: методи, етапи та переваги. *WEZOM*. URL: <https://wezom.com.ua/ua/blog/kak-protestirovat-sajt> (дата звернення: 18.04.2026).
28. Panchal D., Kumar V., Kumar A. Navigating the Testing Terrain: A Comparative Study of Manual and Automated Testing Approaches. *International Research Journal of Engineering and Technology (IRJET)*. 2024. Vol. 11, no. 10. P. 750-758.

29. Maurya V. N., Kumar R. Analytical Study on Manual vs. Automated Testing Using with Simplistic Cost Model. *International Journal of Electronics and Electrical Engineering*. 2012. Vol. 2, no. 1. P. 23-35.

30. Тест-документація: як писати, який її необхідний мінімум і що може статися на проєкті без неї. *DOU*. URL: <https://dou.ua/forums/topic/42571/> (дата звернення: 18.04.2026).

31. Правила оформлення теми тест-кейса. *QATestLab*. URL: <https://training.qatestlab.com/blog/course-materials/test-case-topic/> (дата звернення: 18.04.2026).

32. Що таке цілісність тест-кейса або як правильно описати тест-кейс, щоб перевірити функціонал?. *QATestLab*. URL: <https://training.qatestlab.com/blog/course-materials/test-case-description/> (дата звернення: 19.04.2026).

33. Sabaren L. N., Mascheroni M. A., Greiner C. L., Irrazábal E. A Systematic Literature Review in Cross-browser Testing. *Journal of Computer Science and Technology*. 2018. Vol. 18, no. 1. P. 18-36. URL: <https://doi.org/10.24215/16666038.2018.e01> (date of access: 19.04.2026).

34. Talluri M. Cross-Browser Compatibility Challenges And Solutions In Enterprise Applications. *International Journal of Environmental Sciences*. 2025. Vol. 11, no. 20s. P. 60-65.

35. Vemuri K., Padma V. Methods, Tools and Strategies of Cross Browser Testing a Web Application. *International Journal for Multidisciplinary Research (IJFMR)*. 2022. Vol. 4, no. 1. P. 1-6.

36. Devarapalli C. A. Cross-Browser Compatibility: Modern Web Development with Enhanced User Experience. *Journal of Scientific and Engineering Research*. 2021. Vol. 8, no. 7. P. 242-246.

37. Mujtaba G., Mahmood T., Nasir Z. A Holistic Approach to Software Defect Analysis and Management. *Australian Journal of Basic and Applied Sciences*. 2011. Vol. 5, no. 6. P. 1632-1640.

38. Основні атрибути баг-репорта. *QATestLab*. URL:

<https://training.qatestlab.com/blog/course-materials/attributes-bug-report/> (дата звернення: 20.04.2026).

39. Cachia E., Micallef M. Measuring the Functionality of Online Stores. *3rd Computer Science Annual Workshop (CSAW'05)*. 2005. P. 3-9.

40. Muhamad Z. H. A Comprehensive Model for Web-based Applications Usability Testing: A Comparative Analysis. *International Journal of Computer Applications*. 2025. Vol. 187, no. 50. P. 10-19.

41. Інтернет-магазин DeviceShop. *DeviceShop*. URL: <http://192.168.0.154:8000> (дата звернення: 10.05.2026)

## **ДОДАТОК А**

### **Результати виконання чеклісту тестування вебсистеми**

| №                   | Категорія        | Перевірка                                                            | Очікуваний результат                                                                                                                                                | Браузер Google Chrome v. 140.0.7339.128 | Браузер Mozilla Firefox v. 148.0 (64-bit) | Браузер Safari v. 13.1.2 | Коментар/Баг репорт |
|---------------------|------------------|----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|-------------------------------------------|--------------------------|---------------------|
| 1. Функціональність |                  |                                                                      |                                                                                                                                                                     |                                         |                                           |                          |                     |
| 1.1                 | Функціональність | Запуск проєкту через php artisan serve                               | Проект запускається без помилок                                                                                                                                     | Passed                                  | Passed                                    | Passed                   |                     |
| 1.2                 | Функціональність | Перевірити реєстрацію на сайті                                       | Реєстрація успішна - відкривається профіль та дані зберігаються в БД                                                                                                | Passed                                  | Passed                                    | Passed                   |                     |
| 1.3                 | Функціональність | Перевірити авторизацію на сайті                                      | Відбувається вхід у систему та відкривається профіль користувача                                                                                                    | Passed                                  | Passed                                    | Passed                   |                     |
| 1.4                 | Функціональність | Перевірити вихід з акаунту                                           | Сесія завершується і відбувається перехід на головну сторінку                                                                                                       | Passed                                  | Passed                                    | Passed                   |                     |
| 1.5                 | Функціональність | Перехід між сторінками                                               | Всі маршрути працюють без 404                                                                                                                                       | Passed                                  | Passed                                    | Passed                   |                     |
| 1.6                 | Функціональність | Пошук товарів за назвою                                              | Відображаються релевантні результати                                                                                                                                | Passed                                  | Passed                                    | Passed                   |                     |
| 1.7                 | Функціональність | Додавання товару в кошик                                             | Товар з'являється в кошику та виводиться візуальне підтвердження                                                                                                    | Passed                                  | Passed                                    | Passed                   |                     |
| 1.8                 | Функціональність | Зміна кількості товару                                               | Загальна сума оновлюється коректно                                                                                                                                  | Passed                                  | Passed                                    | Passed                   |                     |
| 1.9                 | Функціональність | Видалення товару з кошика                                            | Товар зникає з кошика                                                                                                                                               | Passed                                  | Passed                                    | Passed                   |                     |
| 1.10                | Функціональність | Можливість повернутися до кошика при несподіваному закритті браузера | Перевірити, чи зберігається вміст кошика користувача у випадку несподіваного закриття браузера або вкладки та чи відновлюється він після повторного відкриття сайту | Passed                                  | Passed                                    | Passed                   |                     |
| 1.11                | Функціональність | Перевірити оновлення профілю користувача (ім'я та дата народження)   | Дані коректно оновлюються та оновлені дані зберігаються в БД                                                                                                        | Passed                                  | Passed                                    | Passed                   |                     |
| 1.12                | Функціональність | Перевірити зміну паролю в особистому кабінеті                        | Пароль змінюється та хешується                                                                                                                                      | Passed                                  | Passed                                    | Passed                   |                     |
| 1.13                | Функціональність | Перевірити функцію "Видалення акаунту"                               | Користувач видаляється з БД та відбувається перехід на головну сторінку                                                                                             | Passed                                  | Passed                                    | Passed                   |                     |
| 1.14                | Функціональність | Перевірити функцію перегортання фото                                 | Користувач може перегортати зображення товару (вперед/назад). Зміна зображень відбувається коректно, без затримок і помилок.                                        | Passed                                  | Passed                                    | Passed                   |                     |
| 1.15                | Функціональність | Перевірити валідацію полів для всіх можливих форм заповнення         | Система перевіряє коректність введених даних перед їх збереженням. У разі некоректного вводу відображається відповідне повідомлення про помилку.                    | Passed                                  | Passed                                    | Passed                   |                     |

Рисунок А.1 – Результати тестування категорії «Функціональність»

| №          | Категорія | Перевірка                                                                     | Очікуваний результат                                                                                                                                  | Браузер Google Chrome<br>v. 140.0.7339.128 | Браузер Mozilla Firefox<br>v. 148.0 (64-bit) | Браузер Safari<br>v. 13.1.2 | Коментар/Баг репорт |
|------------|-----------|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|-----------------------------|---------------------|
| 2. Верстка |           |                                                                               |                                                                                                                                                       |                                            |                                              |                             |                     |
| 2.1        | Верстка   | Перевірити наявність логотипу                                                 | Логотип сайту відображається на сторінці, має коректні розміри та не спотворений                                                                      | Passed                                     | Passed                                       | Passed                      |                     |
| 2.2        | Верстка   | Перевірити наявність головного меню                                           | Головне меню відображається коректно, містить необхідні пункти навігації та забезпечує доступ до основних розділів сайту                              | Passed                                     | Passed                                       | Passed                      |                     |
| 2.3        | Верстка   | Перевірити коректне відображення сторінок                                     | Усі сторінки сайту відображаються без помилок, елементи інтерфейсу не накладаються один на одного, структура сторінок збережена                       | Passed                                     | Passed                                       | Passed                      |                     |
| 2.4        | Верстка   | Перевірити коректне відображення кнопок, блоків меню та ін.                   | Кнопки, меню та інші елементи інтерфейсу відображаються коректно, мають зрозумілу форму та не виходять за межі своїх контейнерів                      | Passed                                     | Passed                                       | Failed                      |                     |
| 2.5        | Верстка   | Перевірити стиль та коректне відображення шрифтів тексту                      | Шрифти відображаються коректно на всіх сторінках, мають задані розміри, стилі та не спотворюються при завантаженні сторінки                           | Passed                                     | Passed                                       | Passed                      |                     |
| 2.6        | Верстка   | Перевірити відображення кольорової гами усіх елементів                        | Кольори відображаються коректно та не викликають проблем із читабельністю                                                                             | Passed                                     | Passed                                       | Passed                      |                     |
| 2.7        | Верстка   | Перевірити коректне розміщення банера на головній сторінці                    | Банер відображається у визначеній зоні головної сторінки, не перекриває інші елементи та коректно масштабується                                       | Passed                                     | Passed                                       | Passed                      |                     |
| 2.8        | Верстка   | Перевірити коректне вирівнювання елементів (кнопки, посилання, заголовки)     | Усі елементи інтерфейсу вирівняні, дотримані відступи, інтерфейс виглядає структурованим та охайним                                                   | Passed                                     | Passed                                       | Passed                      |                     |
| 2.9        | Верстка   | Перевірити зміну курсору на pointer при наведенні на клікабельний елемент     | При наведенні курсору на клікабельні елементи курсор змінюється на pointer, що візуально підтверджує можливість взаємодії                             | Passed                                     | Passed                                       | Passed                      |                     |
| 2.10       | Верстка   | Перевірити реагування активних елементів на наведення курсора (підсвічування) | При наведенні курсора на активні елементи інтерфейсу (кнопки, посилання тощо) відбувається візуальна зміна стану, що підтверджує можливість взаємодії | Passed                                     | Passed                                       | Passed                      |                     |
| 2.11       | Верстка   | Перевірити орфографію/граматику контенту сайту                                | Текстовий контент сайту не містить орфографічних та граматичних помилок, сформульований зрозуміло та відповідає нормам літературної мови              | Passed                                     | Passed                                       | Passed                      |                     |

Рисунок А.2 – Результати тестування категорії «Верстка»

| №    | Категорія | Перевірка                                                                                  | Очікуваний результат                                                                                                                                    | Браузер Google Chrome<br>v. 140.0.7339.128 | Браузер Mozilla Firefox<br>v. 148.0 (64-bit) | Браузер Safari<br>v. 13.1.2 | Коментар/Баг репорт |
|------|-----------|--------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|-----------------------------|---------------------|
| 2.12 | Верстка   | Перевірити відповідність характеристик товару його назві                                   | Характеристики товару повністю відповідають його назві та опису, відсутні розбіжності між заявленими параметрами та фактичними даними                   | Passed                                     | Passed                                       | Passed                      |                     |
| 2.13 | Верстка   | Перевірити адаптацію елементів сайту для роздільної здатності 1280x800                     | Елементи сайту коректно відображаються при роздільній здатності 1280x800, не виходять за межі екрану, зберігається читабельність та зручність взаємодії | Passed                                     | Passed                                       | Passed                      |                     |
| 2.14 | Верстка   | Перевірити відсутність горизонтальної прокрутки після збільшення масштабу сторінки до 150% | При збільшенні масштабу сторінки до 150% не з'являється горизонтальна прокрутка, контент коректно масштабується та залишається доступним для перегляду  | Passed                                     | Passed                                       | Passed                      |                     |

Рисунок А.2, аркуш 2

| №                       | Категорія | Перевірка                                         | Очікуваний результат                                                                                                                                                                     | Браузер Google Chrome<br>v. 140.0.7339.128 | Браузер Mozilla Firefox<br>v. 148.0 (64-bit) | Браузер Safari<br>v. 13.1.2 | Коментар/Баг репорт |
|-------------------------|-----------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|-----------------------------|---------------------|
| 3. Юзабіліті-тестування |           |                                                   |                                                                                                                                                                                          |                                            |                                              |                             |                     |
| 3.1                     | Юзабіліті | Однозначність інтерфейсу                          | Структура сайту однакова на всіх сторінках                                                                                                                                               | Passed                                     | Passed                                       | Passed                      |                     |
| 3.2                     | Юзабіліті | Однозначність і зрозумілість                      | Потрапивши на будь-яку сторінку сайту, користувач одразу ж розуміє, що це за сайт, якої тематики                                                                                         | Passed                                     | Passed                                       | Passed                      |                     |
| 3.3                     | Юзабіліті | Швидкість завантаження                            | Сайт швидко завантажується (1-3 секунди)                                                                                                                                                 | Passed                                     | Passed                                       | Passed                      |                     |
| 3.4                     | Юзабіліті | Зручність взаємодії з логом сайту                 | Логотип клікабельний та веде на головну сторінку                                                                                                                                         | Passed                                     | Passed                                       | Passed                      |                     |
| 3.5                     | Юзабіліті | Передбачуване місцезнаходження ключових елементів | Всі стандартні елементи відвідувач знаходить на звичних місцях:<br>- логотип компанії - зліва вгорі,<br>- контакти - справа вгорі,<br>- рядок пошуку - угорі ліворуч або вгорі по центру | Passed                                     | Passed                                       | Passed                      |                     |
| 3.6                     | Юзабіліті | Наявність "хлібних крихт"                         | Наявність "хлібних крихт" для зручного відстеження шляху до поточної сторінки                                                                                                            | Passed                                     | Passed                                       | Passed                      |                     |
| 3.7                     | Юзабіліті | Наявність інтуїтивно зрозумілих іконок            | Скрізь, де це доречно і можливо, текстові пункти доповнені іконками                                                                                                                      | Passed                                     | Passed                                       | Passed                      |                     |
| 3.8                     | Юзабіліті | Легкість орієнтування                             | Користувач зайшовши на сайт без зусиль орієнтується в розділах, а шлях до цільової сторінки є очевидним та передбачуваним                                                                | Passed                                     | Passed                                       | Passed                      |                     |

Рисунок А.3 – Результати тестування категорії «Юзабіліті»

| №                     | Категорія | Перевірка                               | Очікуваний результат                                                                                                                                                                  | Браузер Google Chrome<br>v. 140.0.7339.128 | Браузер Mozilla Firefox<br>v. 148.0 (64-bit) | Браузер Safari<br>v. 13.1.2 | Коментар/Баг репорт |
|-----------------------|-----------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|-----------------------------|---------------------|
| 4. Тестування безпеки |           |                                         |                                                                                                                                                                                       |                                            |                                              |                             |                     |
| 4.1                   | Безпека   | CSRF-захист                             | Усі форми мають @csrf                                                                                                                                                                 | Passed                                     | Passed                                       | Passed                      |                     |
| 4.2                   | Безпека   | Хешування паролів                       | Паролі не зберігаються відкрито                                                                                                                                                       | Passed                                     | Passed                                       | Passed                      |                     |
| 4.3                   | Безпека   | Забезпечення захисту персональних даних | Використовується auth middleware                                                                                                                                                      | Passed                                     | Passed                                       | Passed                      |                     |
| 4.4                   | Безпека   | Перевірити захист від SQL-ін'єкцій      | Всі операції з БД виконуються з використанням Eloquent ORM або Query Builder Laravel, що унеможливорює виконання SQL-ін'єкцій шляхом автоматичного застосування підготовлених запитів | Passed                                     | Passed                                       | Passed                      |                     |
| 4.5                   | Безпека   | Захист профілю                          | Дані доступні лише власнику                                                                                                                                                           | Passed                                     | Passed                                       | Passed                      |                     |

Рисунок А.4 – Результати тестування категорії «Безпека»

**ДОДАТОК Б**  
**Текст програми**

### Б.1 Файл `account.js` – Динамічна робота з формою авторизації

```

$(document).ready(function() {

    $.ajaxSetup({
        headers: {
            'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
        }
    });
    // Відкриття/закриття вікна аккаунту
    $(".account-trigger").click(function() {
        if ($('.sidebar').hasClass('sidebar_active')) {
            $('.sidebar').removeClass('sidebar_active');
            $('.burger').removeClass('active');
            $('.overlay').hide().removeClass('disabled');
            $('.catalog-list, .menu, .news, .name-
app').removeClass('disabled');
            $('body').css('overflow', 'auto');
        }

        $("body").toggleClass("no-scroll");
        $(".account-popup, .account-overlay").toggleClass("show");

        $("#login-form").show();
        $("#register-form").hide();
    });

    $(".account-overlay, .account-popup button.close-btn").click(function() {
        $("body").removeClass("no-scroll");
        $(".account-popup, .account-overlay").removeClass("show");
    });

    // Перемикання між формами
    $(".switch-to-register").click(function(e) {
        e.preventDefault();
        $("#login-form").hide();
        $("#register-form").show();
    });

    $(".switch-to-login").click(function(e) {
        e.preventDefault();
        $("#register-form").hide();
        $("#login-form").show();
    });
});

```

```

// Налаштування CSRF для AJAX
$.ajaxSetup({
  headers: {
    'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
  }
});

// Функція для показу помилок під полями
function showErrors(form, errors) {
  // спочатку видаляємо старі повідомлення
  $(form).find('.error-message').remove();

  // ітеруємо по всіх полях з помилками
  for (let field in errors) {
    let messages = errors[field]; // масив помилок для поля
    let input = $(form).find('[name="'+field+'"]'); // шукаємо input по name

    // вставляємо після input
    input.after('<div class="error-message" style="color:red;font-size:12px;margin-top:3px;">' + messages.join('<br>') + '</div>');
  }
}

// Логін через AJAX
$("#login-form").submit(function(e) {
  e.preventDefault();

  $.ajax({
    type: "POST",
    url: $(this).attr("action"), // беремо з action форми
    data: $(this).serialize(), // всі поля форми
    success: function(response) {
      if(response.success) {
        window.location.href = response.redirect;
      }
    },
    error: function(xhr) {
      if(xhr.responseJSON && xhr.responseJSON.errors) {
        showErrors('#login-form', xhr.responseJSON.errors);
      } else {
        alert("Помилка входу");
      }
    }
  });
});

```

```

});

// Реєстрація AJAX
$("#register-form").submit(function(e) {
    e.preventDefault();

    $.ajax({
        type: "POST",
        url: "/register",
        data: {
            login: $("#register-login").val(),
            password: $("#register-password").val(),
            password_confirmation: $("#register-password-
confirm").val(),
            _token: $("meta[name='csrf-token']").attr("content")
        },
        success: function(response) {
            if(response.success) {
                window.location.href = response.redirect;
            }
        },
        error: function(xhr) {
            if(xhr.responseJSON && xhr.responseJSON.errors) {
                showErrors('#register-form', xhr.responseJSON.errors);
            } else {
                alert("Помилка реєстрації");
            }
        }
    });
});

// Відкрити popup
$("#open-change-password").click(function () {
    $(".password-overlay, .password-popup").fadeIn();
});

// Закрити popup
$("#close-password-popup, .password-overlay").click(function () {
    $(".password-overlay, .password-popup").fadeOut();
});

// Зміна пароля AJAX
$("#change-password-form").submit(function (e) {
    e.preventDefault();

```

```

$.ajax({
    type: "POST",
    url: "/change-password",
    data: $(this).serialize(),
    success: function (response) {
        alert(response.message);
        $(".password-overlay, .password-popup").fadeOut();
        $("#change-password-form")[0].reset();
    },
    error: function (xhr) {
        if (xhr.responseText && xhr.responseText.errors) {

alert(Object.values(xhr.responseText.errors).flat().join("\n"));
            } else {
                alert("Помилка зміни пароля");
            }
        }
    });
});

$("#edit-name").click(function () {
    $("#edit-title").text("Зміна імені");
    $("#edit-label").text("Ім'я");
    $("#edit-field").val("name");

    $("#edit-value")
        .attr("type", "text")
        .val($("#name").text().trim());

    $(".edit-overlay, .edit-popup").fadeIn();
});

$("#edit-birthday").click(function () {
    $("#edit-title").text("Дата народження");
    $("#edit-label").text("Дата народження");
    $("#edit-field").val("birthday");

    $("#edit-value")
        .attr("type", "date")
        .val("");

    $(".edit-overlay, .edit-popup").fadeIn();
});

$("#close-edit-popup, .edit-overlay").click(function () {

```

```

        $(".edit-overlay, .edit-popup").fadeOut();
    });

    $("#edit-profile-form").submit(function (e) {
        e.preventDefault();

        $.ajax({
            type: "POST",
            url: "/update-profile-field",
            data: $(this).serialize(),
            success: function (response) {
                $("##" + response.field).text(response.value);
                $(".edit-overlay, .edit-popup").fadeOut();
            },
            error: function (xhr) {

                alert(Object.values(xhr.responseJSON.errors).flat().join("\n"));
            }
        });
    });

    // Відкрити попуп видалення акаунту
    $("#open-delete-account").click(function () {
        $(".delete-overlay, .delete-popup").fadeIn();
    });

    // Закрити попуп
    $("#close-delete-popup, .delete-overlay").click(function () {
        $(".delete-overlay, .delete-popup").fadeOut();
    });

    $("#delete-account-form").submit(function (e) {
        e.preventDefault();

        if (!confirm("Ви точно хочете видалити акаунт?")) return;

        $.ajax({
            type: "POST",
            url: "/delete-account",
            data: $(this).serialize(),
            success: function (response) {
                alert(response.message);
                window.location.href = response.redirect;
            },
            error: function () {

```

```

        alert("Помилка видалення акаунту");
    }
    });
});

});

```

## Б.2 Файл support.js – Динамічна робота з формою підтримки

```

function toggleFaq(question) {
    const faqItem = question.parentNode;
    faqItem.classList.toggle('opened');
    faqItem.classList.toggle('closed');
    const answer = faqItem.querySelector('.faq-answer');
    answer.style.display = faqItem.classList.contains('opened') ? 'block' : 'none';
}

function toggleContactField() {
    var contactType = document.getElementById('contactType').value;
    var phoneField = document.getElementById('phoneField');
    var emailField = document.getElementById('emailField');

    if (contactType === 'phone') {
        phoneField.style.display = 'block';
        emailField.style.display = 'none';
    } else if (contactType === 'email') {
        phoneField.style.display = 'none';
        emailField.style.display = 'block';
    }
}

$('input, textarea, select').on('input change', function () {
    $(this).next('.error').text("");
});

// Функція для відображення повідомлення про помилку під конкретним полем
function displayError(elementId, errorMessage) {
    document.getElementById(elementId).innerHTML = errorMessage;
}

// Всі перевірки пройдені, форму можна відправити на сервер
var formData = new FormData();
formData.append("name", name);

```

```

formData.append("contactType", contactType);
formData.append("phone", phone);
formData.append("email", email);
formData.append("topic", topic);
formData.append("message", message);

// Відправка форми на сервер (PHP)
fetch('process-form.php', {
  method: 'POST',
  body: formData
})
.then(response => response.json())
.then(data => {
  if (data.ok) {
    alert('Форма успішно відправлена!');
  } else {
    alert('Помилка відправлення форми. Будь ласка, спробуйте пізніше.');

```

```

const nameRegex = /^[A-Za-zA-Яa-яİiLiЄєГг's-]{2,}$/;
const phoneRegex = /^\\+?\\d{10,12}$/;
const emailRegex = /^[^\\s@]+@[^\\s@]+\\.\\.[^\\s@]+$/;

// Ім'я
if (name === "") {
    $('#nameError').text("Будь ласка, введіть ім'я");
    isValid = false;
} else if (!nameRegex.test(name)) {
    $('#nameError').text("Ім'я має містити лише літери та мінімум 2  
символи");
    isValid = false;
}

// Тип зв'язку
if (!contactType) {
    $('#contactTypeError').text("Оберіть спосіб зв'язку");
    isValid = false;
}

// Телефон
if (contactType === 'phone') {
    if (phone === "") {
        $('#phoneError').text("Введіть номер телефону");
        isValid = false;
    } else if (!phoneRegex.test(phone)) {
        $('#phoneError').text("Номер телефону має містити 10–12 цифр");
        isValid = false;
    }
}

// Email
if (contactType === 'email') {
    if (email === "") {
        $('#emailError').text("Введіть електронну пошту");
        isValid = false;
    } else if (!emailRegex.test(email)) {
        $('#emailError').text("Некоректний формат email");
        isValid = false;
    }
}

// Тема
if (!topic) {
    $('#topicError').text("Оберіть тему повідомлення");
}

```

```

    isValid = false;
}

// Повідомлення
if (message === "") {
    $('#messageError').text("Напишіть повідомлення");
    isValid = false;
} else if (message.length < 10) {
    $('#messageError').text("Повідомлення має містити мінімум 10
символів");
    isValid = false;
}

if (!isValid) return;

// AJAX
let form = $('#myForm');

$.ajax({
    url: form.attr('action'),
    type: 'POST',
    data: form.serialize(),
    success: function () {
        form.fadeOut(300, function () {
            $('.support-success-message').fadeIn();
        });
    },
    error: function () {
        alert('Сталася помилка. Спробуйте пізніше ☹️');
    }
});

}

// очищення помилки при вводі
$('input, textarea, select').on('input change', function () {
    $(this).next('.error').text("");
});

$('#phone').on('input', function () {
    // зберігаємо початковий знак +
    let val = this.value;

    // якщо перший символ + — залишаємо його, а решту тільки цифри
    if (val.startsWith('+')) {
        this.value = '+' + val.slice(1).replace(/\D/g, "");
    }
});

```

```

    } else {
        this.value = val.replace(/^\D/g, "");
    }
});

```

### Б.3 Файл shoppingCart.js – Динамічна робота з функціями кошика

```

$(document).ready(function () {

    // Відкриття/закриття кошика
    $(".cart-trigger").click(function () {
        $("body").toggleClass("no-scroll");
        $("#cart-popup, .cart-overlay").toggleClass("show");
        updateCartDisplay();
    });

    $(".cart-overlay, .cart-popup .cart-close").click(function () {
        $("body").removeClass("no-scroll");
        $("#cart-popup, .cart-overlay").removeClass("show");
    });

    // Додати товар у кошик (кнопка Купити)
    $(".buy-button").click(function () {
        let productId = $(this).data('id');
        $.ajax({
            url: '/cart/add/' + productId,
            type: 'POST',
            data: { _token: $('meta[name="csrf-token"]').attr('content') },
            success: function (response) {
                alert(response.message);
                updateCartDisplay();
            }
        });
    });

    // Видалення товару з кошика
    $(document).on('click', '.remove-btn', function () {
        let row = $(this).closest('tr');
        let productId = row.data('id');

        $.ajax({
            url: '/cart/remove/' + productId,
            type: 'POST',

```

```

        data: { _token: $('meta[name="csrf-token"]').attr('content') },
        success: function (response) {
            alert(response.message);
            row.remove();
            updateCartDisplay();
        }
    });
});

// Зміна кількості
$(document).on('change', '.quantity-input', function () {
    let row = $(this).closest('tr');
    let productId = row.data('id');
    let quantity = $(this).val();

    $.ajax({
        url: '/cart/update/' + productId,
        type: 'POST',
        data: { quantity: quantity, _token: $('meta[name="csrf-
token"]').attr('content') },
        success: function (response) {
            alert(response.message);
            updateCartDisplay();
        }
    });
});

// Оновлення відображення кошика
function updateCartDisplay() {
    $.get('/cart', function (html) {
        let newCart = $(html).find('#cart-items').html();
        $('#cart-items').html(newCart);

        let total = $(html).find('#total').text();
        $('#total').text(total);
    });
}

$('.checkout-form').submit(function (e) {
    e.preventDefault();

    let form = $(this);
    let url = form.attr('action');
    let data = form.serialize();

```

```

$.ajax({
    url: url,
    type: 'POST',
    data: data,
    headers: {
        'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
    },
    success: function (response) {
        form.hide();
        $('.checkout-message').fadeIn();
    },
    error: function (xhr) {
        // показуємо реальну помилку від сервера
        console.log(xhr.responseText);
        alert('Сталася помилка при оформленні замовлення: ' + xhr.status);
    }
});
});
});

```

#### **Б.4 Файл AuthController.php – Контролер для обробки блоку авторизації/реєстрації**

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Auth;

class AuthController extends Controller
{
    /**
     * Реєстрація користувача
     */
    public function register(Request $request)
    {
        // Валідація введених даних
        $request->validate([
            'login' => 'required|string|min:3|max:255|unique:users,login',
            'password' => 'required|string|min:6|confirmed',
        ], [

```

```

        'login.required' => 'Логін обов'язковий',
        'login.unique' => 'Такий логін вже існує',
        'login.min' => 'Логін занадто короткий',
        'password.required' => 'Пароль обов'язковий',
        'password.confirmed' => 'Паролі не співпадають',
        'password.min' => 'Пароль мінімум 6 символів',
    );

    // Створення користувача в базі
    $user = User::create([
        'login' => $request->login,
        'password' => Hash::make($request->password),
    ]);

    // Автоматична авторизація після реєстрації
    Auth::login($user);

    // Відправка JSON відповіді для JS
    return response()->json([
        'success' => true,
        'redirect' => route('profile'),
        'user_id' => $user->id,
        'auth_check' => Auth::check()
    ]);
}

/**
 * Логін користувача
 */
public function login(Request $request)
{
    // Валідація даних
    $request->validate([
        'login' => 'required|string',
        'password' => 'required|string',
    ], [
        'login.required' => 'Поле логін обов'язкове',
        'password.required' => 'Поле пароль обов'язкове',
    ]);

    $credentials = $request->only('login', 'password');

    // Спроба авторизації
    if (Auth::attempt($credentials)) {
        $request->session()->regenerate(); // оновлення сесії після логіну
    }
}

```

```

        return response()->json([
            'success' => true,
            'redirect' => route('profile')
        ]);
    }

    // Якщо логін/пароль невірний
    return response()->json([
        'success' => false,
        'errors' => ['password' => ['Невірний логін або пароль']]
    ], 422);
}

/**
 * Вихід користувача
 */
public function logout(Request $request)
{
    Auth::logout();
    $request->session()->invalidate();
    $request->session()->regenerateToken();

    return redirect()->route('main'); // або на головну сторінку
}

public function changePassword(Request $request)
{
    $request->validate([
        'current_password' => 'required',
        'new_password' => 'required|min:6|confirmed',
    ], [
        'current_password.required' => 'Введіть старий пароль',
        'new_password.required' => 'Введіть новий пароль',
        'new_password.confirmed' => 'Паролі не співпадають',
    ]);

    $user = Auth::user();

    if (!Hash::check($request->current_password, $user->password)) {
        return response()->json([
            'errors' => [
                'current_password' => ['Старий пароль невірний']
            ]
        ], 422);
    }
}

```

```

    }

    $user->password = Hash::make($request->new_password);
    $user->save();

    return response()->json([
        'success' => true,
        'message' => 'Пароль успішно змінено'
    ]);
}

public function updateProfileField(Request $request)
{
    $request->validate([
        'field' => 'required|in:name,birthday',
        'value' => 'required',
    ]);

    $user = Auth::user();

    if ($request->field === 'birthday') {
        $request->validate([
            'value' => 'date',
        ]);
    }

    $user->{$request->field} = $request->value;
    $user->save();

    return response()->json([
        'field' => $request->field,
        'value' => $request->value,
    ]);
}

public function deleteAccount(Request $request)
{
    $user = Auth::user();

    Auth::logout(); // спочатку розлогін

    $user->delete(); // видалення з БД

    $request->session()->invalidate();
    $request->session()->regenerateToken();
}

```

```

        return response()->json([
            'success' => true,
            'message' => 'Акаунт успішно видалено',
            'redirect' => route('main')
        ]);
    }
}

```

### **Б.5 Файл SearchController.php – Контролер для обробки пошуку за назвою товару**

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class SearchController extends Controller
{
    public function index(Request $request)
    {
        $query = trim($request->get('q'));

        $products = collect(config('products'))
            ->filter(function ($product) use ($query) {
                return str_contains(
                    mb_strtolower($product['name']),
                    mb_strtolower($query)
                );
            });

        return view('search', [
            'query' => $query,
            'products' => $products
        ]);
    }
}

```

## Б.6 Файл products.php – Конфігураційний файл для збереження товару

```
<?php
```

```
return [
    1 => [
        'name' => 'Комп'ютер ARTLINE Gaming X69WHITE (X69WHITEv42)',
        'price' => 65999,
        'image' => 'computerARTLINE-Gaming-X69WHITEv42-4.jpg',
        'route' => 'goodsComputerARTLINE-Gaming-X69WHITEv42',
    ],
    2 => [
        'name' => 'Мобільний телефон Apple iPhone 15 128GB Black',
        'price' => 42499,
        'image' => 'iPhone15(1).jpg',
        'route' => 'goodsIphone15',
    ],
    3 => [
        'name' => 'Геймерська клавіатура MSI Vigor GK50 LOW PROFILE UA',
        'price' => 3499,
        'image' => 'vigor-gk-50.png',
        'route' => 'goodsKeyboardVigorGK50',
    ],
    4 => [
        'name' => 'Ноутбук ASUS Zenbook S 16 UM5606KA-RK741W Zumaia Gray',
        'price' => 59999,
        'image' => 'laptopASUSZenbookS16-UM5606KA-RK741W-2.jpg',
        'route' => 'goodsLaptopASUSZenbookS16-UM5606KA-RK741W',
    ],
    5 => [
        'name' => 'Мобільний телефон POCO X3 Pro 6Gb/128Gb Phantom Black',
        'price' => 7652,
        'image' => 'poco-x3pro.jpg',
        'route' => 'goodsPocoX3PRO',
    ],
    6 => [
        'name' => 'Відеокарта ASUS RTX 4070Ti 12GB ROG Strix Gaming OC (ROG-STRIX-RTX4070TI-O12G-GAMING)',
        'price' => 43563,
```

```

    'image' => 'asus-rog1.jpg',
    'route' => 'goodsRTX4070TI',
],
  7 => [
    'name' => 'Планшет Samsung Galaxy Tab A11 LTE 8/128GB Gray (SM-
X135FZAEEUC)',
    'price' => 7599,
    'image' => 'tabletSamsung-Galaxy-Tab-A11-LTE-1.jpg',
    'route' => 'goodsTabletSamsung-Galaxy-Tab-A11-LTE',
],
  8 => [
    'name' => 'Ігрова приставка Sony PlayStation 5 Slim Digital Edition',
    'price' => 25999,
    'image' => 'playstation5slim-4.jpg',
    'route' => 'goodsSony-PlayStation5slim',
],
  9 => [
    'name' => 'Смарт-годинник Xiaomi Redmi Watch 5 Active Black
(BHR8784GL)',
    'price' => 1499,
    'image' => 'smartwatchXiaomi-Redmi-Watch-5-2.jpg',
    'route' => 'smartwatchXiaomi-Redmi-Watch-5',
],
  10 => [
    'name' => 'Фітнес-браслет Samsung Galaxy Fit3 Grey (SM-
R390NZAASEK)',
    'price' => 1999,
    'image' => 'smartbraceletSamsung-Galaxy-Fit3-1.jpg',
    'route' => 'smartbraceletSamsung-Galaxy-Fit3',
],
  11 => [
    'name' => 'Гра Ghost of Yotei для PS5 (Blu-ray диск)',
    'price' => 2999,
    'image' => 'gameGhost-of-Yotei-1.jpg',
    'route' => 'gameGhost-of-Yotei',
],
  12 => [
    'name' => 'Навушники HyperX Cloud Alpha (4P5L1AM)',
    'price' => 4299,
    'image' => 'headsetHyperX-Cloud-Alpha-1.jpg',
    'route' => 'headsetHyperX-Cloud-Alpha',
],
  13 => [
    'name' => 'Миша Logitech G102 Lightsync USB Black (910-005823)',
    'price' => 1399,

```

```

        'image' => 'mouseLogitech-G102-1.png',
        'route' => 'mouseLogitech-G102',
    ],
    14 => [
        'name' => 'Крісло для геймерів PROOVE Gaming Geek White/Purple (GCGK00001016)',
        'price' => 17999,
        'image' => 'armchairPROOVE-Gaming-Geek-2.jpg',
        'route' => 'armchairPROOVE-Gaming-Geek',
    ],
    15 => [
        'name' => 'Корпус Gamdias ATLAS P2 CG Mid-Tower PC Case',
        'price' => 6999,
        'image' => 'pcframeGamdias-ATLAS-P2-CG-1.jpg',
        'route' => 'pcframeGamdias-ATLAS-P2-CG',
    ],
    16 => [
        'name' => 'Кулер процесорний APNX AP1 Black (APTC-PF30517.11)',
        'price' => 1799,
        'image' => 'coolerAPNX-AP1-Black-1.jpg',
        'route' => 'coolerAPNX-AP1-Black',
    ],
    17 => [
        'name' => 'Оперативна пам'ять Kingston Fury Beast DDR5-6000 RGB 32768MB PC5-48000 (Kit of 2x16384) Black (KF560C30BBAK2-32)',
        'price' => 17999,
        'image' => 'ramKINGSTON-32GB-DDR5-6000MHz-Fury-Beast-RGB-1.jpg',
        'route' => 'ramKingston-Fury-Beast-DDR5-6000-RGB-32768MB',
    ],
    18 => [
        'name' => 'Процесор Intel Core i5-14400F 2.5GHz/20MB (BX8071514400F) s1700 BOX',
        'price' => 8999,
        'image' => 'cpuINTEL-CoreI5-14400F-BOX-1.jpg',
        'route' => 'cpuINTEL-CoreI5-14400F-BOX',
    ],
    19 => [
        'name' => 'Материнська плата ASUS PRIME B550M-A(sAM4, AMD B550, PCI-Ex16)',
        'price' => 3999,
        'image' => 'motherboardASUS-PRIME-B550M-A-1.jpg',
        'route' => 'motherboardASUS-PRIME-B550M-A',
    ],
    20 => [

```

```

        'name' => 'SSD диск Kingston SSDNow A400 960GB 2.5" SATAIII 3D V-
        NAND (SA400S37/960G)',
        'price' => 6999,
        'image' => 'ssdKINGSTON -A400-960GB-1.jpg',
        'route' => 'ssdKINGSTON -A400-960GB',
    ],
    21 => [
        'name' => 'Блок живлення CHIEFTEC Photon 750W (CTG-750C-RGB)',
        'price' => 3699,
        'image' => 'pcpowerCHIEFTEC-Photon-750W-1.jpg',
        'route' => 'pcpowerCHIEFTEC-Photon-750W',
    ],
    22 => [
        'name' => 'Коробка передач Logitech Driving Force Shifter PC/Xbox
        One/PS3/PS4 Black (941-000130)',
        'price' => 2799,
        'image' => 'controllerLogitech-Driving-Force-Shifter-1.jpg',
        'route' => 'controllerLogitech-Driving-Force-Shifter',
    ],
    23 => [
        'name' => 'Ігрова поверхня GAMEPRO MP345B',
        'price' => 599,
        'image' => 'mousepadGAMEPRO-MP345B-1.jpg',
        'route' => 'mousepadGamePro-MP345B',
    ],
];

```

## **Б.7 Файл SupportController.php – Контролер для обробки AJAX-форми підтримки**

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class SupportController extends Controller
{
    public function store(Request $request)
    {
        // (опціонально) валідація
        $request->validate([
            'name' => 'required|string',

```

```

        'contactType' => 'required',
        'message' => 'required|string',
    ]);

    return response()->json([
        'message' => 'Дякуємо за звернення!<br>Ми отримали ваше повідомлення та зв'яжемося з вами найближчим часом.'
    ]);
}
}

```

## **Б.8 Файл CartController.php – Контролер для взаємодії з кошиком і товаром у ньому**

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class CartController extends Controller
{
    // Показати кошик
    public function index()
    {
        $cart = session()->get('cart', []);
        return view('cart', compact('cart'));
    }

    // Додати товар в кошик через AJAX
    public function add(Request $request, $productId)
    {
        $products = config('products');

        if (!isset($products[$productId])) {
            return response()->json(['error' => 'Товар не знайдено'], 404);
        }

        $cart = session()->get('cart', []);

        if (isset($cart[$productId])) {
            $cart[$productId]['quantity'] += 1;
        } else {

```

```

        $cart[$productId] = [
            'name' => $products[$productId]['name'],
            'price' => $products[$productId]['price'],
            'image' => $products[$productId]['image'],
            'quantity' => 1,
        ];
    }

    session()->put('cart', $cart);

    return response()->json([
        'success' => true,
        'message' => 'Товар додано до кошика!',
        'cart_count' => count($cart),
    ]);
}

// Оновити кількість товару
public function update(Request $request, $productId)
{
    $cart = session()->get('cart', []);

    if (isset($cart[$productId])) {
        $quantity = max(1, (int) $request->quantity);
        $cart[$productId]['quantity'] = $quantity;
        session()->put('cart', $cart);

        return response()->json(['success' => true, 'message' => 'Кількість оновлено!']);
    }

    return response()->json(['error' => 'Товар не знайдено в кошику.'], 404);
}

// Видалити товар з кошика
public function remove(Request $request, $productId)
{
    $cart = session()->get('cart', []);

    if (isset($cart[$productId])) {
        unset($cart[$productId]);
        session()->put('cart', $cart);

        return response()->json(['success' => true, 'message' => 'Товар видалено з кошика!']);
    }
}

```

```

    }

    return response()->json(['error' => 'Товар не знайдено в кошику.'], 404);
}

// Очистити весь кошик
public function clear()
{
    session()->forget('cart');
    return response()->json(['success' => true, 'message' => 'Кошик очищено!']);
}
}

```

### **Б.9 Файл CheckoutController.php – Контролер для відправлення повідомлення користувачу про успішне оформлення замовлення**

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class CheckoutController extends Controller
{
    public function store(Request $request)
    {
        return response()->json([
            'message' => 'Дякуємо за замовлення. Очікуйте повідомлення.'
        ]);
    }
}

```

### **Б.10 Файл User.php – Файл моделі для реалізації логіки користувача**

```

<?php

namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;

```

```

use Illuminate\Notifications\Notifiable;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    protected $fillable = [
        'login',
        'password',
        'name',
        'birthday',
    ];

    protected $hidden = [
        'password',
    ];

    // Користувач може мати багато замовлень
    public function orders()
    {
        return $this->hasMany(Order::class);
    }

    // Користувач може мати багато товарів у кошику
    public function carts()
    {
        return $this->hasMany(Cart::class);
    }
}

```

### **Б.11 Файл web.php – Маршрути проєкту**

```

<?php

use App\Http\Controllers\ProfileController;
use App\Http\Controllers\AuthController;
use Illuminate\Support\Facades\Route;
use Mcamara\LaravelLocalization\Facades\LaravelLocalization;
use App\Http\Controllers\CartController;
use App\Http\Controllers\SearchController;
use App\Http\Controllers\CheckoutController;
use App\Http\Controllers\SupportController;

```

```

Route::group(
[
    'prefix' => LaravelLocalization::setLocale(),
    'middleware' => ['web'],
],
function () {
    Route::get('/', fn() => view('main'))->name('main');
    Route::get('/catalog', fn() => view('catalog'))->name('catalog');
    Route::get('/contacts', fn() => view('contacts'))->name('contacts');
    Route::get('/support', fn() => view('support'))->name('support');
    Route::post('/support', [SupportController::class, 'store']);
    Route::get('/information', fn() => view('information'))->name('information');
    Route::get('/laptopsAndComputers', fn() => view('laptopAndComputers'))->name('laptopAndComputers');
    Route::get('/smartphonesAndElectronics', fn() => view('smartphonesAndElectronics'))->name('smartphonesAndElectronics');
    Route::get('/smartphonesAndElectronics/smartphones', fn() => view('smartphones'))->name('smartphones');
    Route::get('/goodsForGamers/gamingPeripherals', fn() => view('gamingPeripherals'))->name('gamingPeripherals');
    Route::get('/laptopsAndComputers/complectsPC', fn() => view('complectsPC'))->name('complectsPC');
    Route::get('/laptopsAndComputers/complectsPC/videocards', fn() => view('videoCards'))->name('videoCards');
    Route::get('/laptopsAndComputers/complectsPC/RAMs', fn() => view('rams'))->name('rams');
    Route::get('/laptopsAndComputers/complectsPC/CPU', fn() => view('cpu'))->name('cpu');
    Route::get('/laptopsAndComputers/complectsPC/SSDandHDD', fn() => view('ssd'))->name('ssd');
    Route::get('/laptopsAndComputers/complectsPC/pcpower', fn() => view('pcpower'))->name('pcpower');
    Route::get('/laptopsAndComputers/complectsPC/motherboards', fn() => view('motherboards'))->name('motherboards');
    Route::get('/laptopsAndComputers/complectsPC/pcframe', fn() => view('pcframe'))->name('pcframe');
    Route::get('/laptopsAndComputers/laptops', fn() => view('laptop'))->name('laptop');
    Route::get('/laptopsAndComputers/computers', fn() => view('computers'))->name('computers');
    Route::get('/goodsForGamers/consoles', fn() => view('consoles'))->name('consoles');
    Route::get('/smartphonesAndElectronics/tablets', fn() => view('tablets'))->name('tablets');
}

```

```

Route::get('/smartphonesAndElectronics/wearableGadgets', fn() =>
view('wearable-gadgets'))->name('wearable-gadgets');
Route::get('/smartphonesAndElectronics/wearableGadgets/smartwatches', fn()
=> view('smartwatches'))->name('smartwatches');
Route::get('/smartphonesAndElectronics/wearableGadgets/fitnessbracelets',
fn() => view('smartbracelets'))->name('smartbracelets');
Route::get('/smartphonesAndElectronics/wearableGadgets/fitnessbracelets/S
amsung-Galaxy-Fit3', fn() => view('smartbraceletSamsung-Galaxy-Fit3'))-
>name('smartbraceletSamsung-Galaxy-Fit3');
Route::get('/smartphonesAndElectronics/tablets/Samsung-Galaxy-
Tab-A11-LTE', fn() => view('goodsTabletSamsung-Galaxy-Tab-A11-LTE'))-
>name('goodsTabletSamsung-Galaxy-Tab-A11-LTE');
Route::get('/goodsForGamers/gamingPeripherals/keyboards', fn() =>
view('keyboards'))->name('keyboards');
Route::get('/goodsForGamers/gamingPeripherals/mice', fn() =>
view('mice'))->name('mice');
Route::get('/goodsForGamers/gamingPeripherals/headsets', fn() =>
view('headsets'))->name('headsets');
Route::get('/goodsForGamers/gamingPeripherals/armchairs', fn() =>
view('armchairs'))->name('armchairs');
Route::get('/goodsForGamers/gamingPeripherals/mousepads', fn() =>
view('mousepads'))->name('mousepads');
Route::get('/goodsForGamers/gamingPeripherals/controllers', fn() =>
view('controllers'))->name('controllers');
Route::get('/goodsForGamers', fn() => view('goodsForGamers'))-
>name('goodsForGamers');
Route::get('/goodsForGamers/games', fn() => view('games'))-
>name('games');
Route::get('/goodsForGamers/games/gameGhost-of-Yotei', fn() =>
view('gameGhost-of-Yotei'))->name('gameGhost-of-Yotei');
Route::get('/goodsForGamers/gamingPeripherals/mice/Logitech-
G102', fn() => view('mouseLogitech-G102'))->name('mouseLogitech-G102');
Route::get('/goodsForGamers/gamingPeripherals/mousepads/GamePro-
MP345B', fn() => view('mousepadGamePro-MP345B'))-
>name('mousepadGamePro-MP345B');
Route::get('/laptopsAndComputers/complectsPC/pcframe/Gamdias-
ATLAS-P2-CG', fn() => view('pcframeGamdias-ATLAS-P2-CG'))-
>name('pcframeGamdias-ATLAS-P2-CG');
Route::get('/laptopsAndComputers/complectsPC/CPU/INTEL-
CoreI5-14400F-BOX', fn() => view('cpuINTEL-CoreI5-14400F-BOX'))-
>name('cpuINTEL-CoreI5-14400F-BOX');
Route::get('/laptopsAndComputers/complectsPC/SSDandHDD/KINGSTON
-A400-960GB', fn() => view('ssdKINGSTON -A400-960GB'))-
>name('ssdKINGSTON -A400-960GB');

```

```

Route::get('/laptopsAndComputers/complectsPC/videocards/ASUS-
RTX-4070Ti-12GB-ROG-Strix-Gaming-OC', fn() => view('goodsRTX4070TI'))-
>name('goodsRTX4070TI');
Route::get('/goodsForGamers/gamingPeripherals/keyboards/MSI-
Vigor-GK50-LOW-PROFILE', fn() => view('goodsKeyboardVigorGK50'))-
>name('goodsKeyboardVigorGK50');
Route::get('/smartphonesAndElectronics/smartphones/Apple-iPhone-
15-128GB', fn() => view('goodsIphone15'))->name('goodsIphone15');
Route::get('/smartphonesAndElectronics/smartphones/POCO-X3-Pro-
6Gb-128Gb', fn() => view('goodsPocoX3PRO'))->name('goodsPocoX3PRO');
Route::get('/goodsForGamers/consoles/Sony-PlayStation5slim', fn()
=> view('goodsSony-PlayStation5slim'))->name('goodsSony-PlayStation5slim');
Route::get('/none', fn() => view('none'))->name('none');
Route::post('/register', [AuthController::class, 'register'])-
>name('register');
Route::post('/login', [AuthController::class, 'login'])->name('login');
Route::post('/logout', function () {
    Auth::logout();
    return redirect()->route('main');
})->name('logout');
Route::middleware('auth')->group(function () {
    Route::get('/profile', function () {
        return view('profile');
    })->name('profile');
    Route::patch('/profile', [ProfileController::class, 'update'])-
>name('profile.update');
    Route::delete('/profile', [ProfileController::class, 'destroy'])-
>name('profile.destroy');
});
Route::post('/change-password', [AuthController::class,
'changePassword'])
->middleware('auth')
->name('change.password');
Route::post('/update-profile-field', [AuthController::class,
'updateProfileField'])
->middleware('auth');
Route::post('/delete-account', [AuthController::class, 'deleteAccount'])
->middleware('auth');
Route::get('/cart', [CartController::class, 'index'])->name('cart');
Route::get('/checkout', fn() => view('checkout'))->name('checkout');
Route::post('/checkout', [CheckoutController::class, 'store']);
Route::post('/cart/add/{id}', [CartController::class, 'add'])-
>name('cart.add');
Route::post('/cart/update/{id}', [CartController::class, 'update'])-
>name('cart.update');

```

```

        Route::post('/cart/remove/{id}', [CartController::class, 'remove'])-
>name('cart.remove');
        Route::post('/cart/clear', [CartController::class, 'clear'])-
>name('cart.clear');
        Route::get('/search', [SearchController::class, 'index'])-
>name('search');
    }
);

```

**Б.12    Файл    0001\_01\_01\_000000\_create\_users\_table.php    –    Файл  
міграції для створення таблиці користувачів**

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('login')->unique();
            $table->string('password');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('users');
    }
}

```

**Б.13    Файл    2026\_02\_04\_161103\_add\_name\_and\_birthday\_to\_users\_table.php    –    Файл  
міграції для модифікації таблиці користувачів**

```

<?php

use Illuminate\Database\Migrations\Migration;

```

```

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    public function up(): void
    {
        Schema::table('users', function (Blueprint $table) {
            $table->string('name')->nullable()->after('login');
            $table->date('birthday')->nullable()->after('login');
        });
    }

    public function down(): void
    {
        Schema::table('users', function (Blueprint $table) {
            $table->dropColumn(['name', 'birthday']);
        });
    }
};

```

#### Б.14 Файл `catalog.blade.php` – Сторінка каталогу вебсистеми

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Каталог | DeviceShop</title>
        <link rel="stylesheet" href="{{ asset('css/catalog.css') }}">
        <link rel="stylesheet" href="{{ asset('css/base.css') }}">
        <link rel="stylesheet" href="{{ asset('css/sidebar.css') }}">
        <link rel="stylesheet" href="{{ asset('css/menu.css') }}">
        <link rel="stylesheet" href="{{ asset('css/account.css') }}">
        <link rel="stylesheet" href="{{ asset('css/shoppingCart.css') }}">
        <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
        <script src="{{ asset('js/sidebar.js') }}"></script>
        <script src="{{ asset('js/search.js') }}"></script>
        <script src="{{ asset('js/news.js') }}"></script>
        <script src="{{ asset('js/account.js') }}"></script>
        <script src="{{ asset('js/shoppingCart.js') }}"></script>
        <meta name="csrf-token" content="{{ csrf_token() }}">
    </head>
    <body>

```

```

<div class="header">
  <div class="container">
    <div class="name-app">
      <h2>Інтернет-магазин DeviceShop: найкращі ціни тільки у
нас</h2>
    </div>
  </div>
</div>
<div class="overlay"></div>
<div class="header-menu">
  <div class="sidebar">
    <a href="#" class="sidebar-btn">✕</a>
    <nav class="sidebar-list">
      <a href="{{ route('main') }}"> </a>
      <hr>
      <div class="sidebar-account">
        @auth
          <a href="{{ route('profile') }}">
            
          </a>
          <a href="{{ route('profile')
}}">Особистий профіль</a>
        @else
          {{-- Гість --}}
          <a href="#" class="account-trigger">
            
          </a>
          <a href="#" class="account-
trigger">Особистий профіль</a>
        @endauth
      </div>
      <div class="sidebar-catalog">
        <a href="{{ route('catalog') }}"> </a>
        <a href="{{ route('catalog') }}">Каталог</a>
      </div>
      <div class="sidebar-shopping-cart">
        <a href="{{ route('cart') }}">
          
          </a>
          <a href="{{ route('cart') }}">Кошик</a>
      </div>
    </nav>
  </div>
</div>

```

```

        </div>
        <div class="sidebar-contacts">
            <a href="{{ route('contacts') }}"> </a>
            <a href="{{ route('contacts') }}">Зв'язатися з нами</a>
        </div>
        <div class="sidebar-info">
            <a href="{{ route('information') }}"> </a>
            <a href="{{ route('information') }}">Інформація про
компанію</a>
        </div>
        <div class="sidebar-support">
            <hr>
            <a href="{{ route('support') }}"> </a>
            <a href="{{ route('support') }}">Довідниковий центр</a>
        </div>
    </nav>
</div>
<div class="container">
    <div class="menu">
        <nav class="navbar">
            <div class="burger">
                <span></span>
            </div>
        </nav>

        <div class="logo">
            <a href="{{ route('main') }}"> </a>
        </div>

        <div class="catalog">
            <a href="{{ route('catalog') }}"> </a>
            <a href="{{ route('catalog') }}">Каталог</a>
        </div>
        <div class="search">
            <a href="#" id="search-link">
                
            </a>
            <form id="search-form" action="{{ route('search') }}"
method="GET">
                <input

```

```

type="text"
id="search-input"
name="q"
placeholder="Пошук товарів"
value="{{ request('q') }}"
>
<button
type="submit">Знайти</button>
</form>
</div>

<div class="shopping-cart">
<a href="{{ route('cart') }}">

</a>
</div>
<div class="cart-popup" id="cart-popup">
<div class="cart-content">
<div class="cart-header">
<h2>Кошик</h2>
<span class="cart-close">✕</span>
</div>
<hr>
<p>Кошик порожній. Але це ніколи не пізно виправити
:)</p>
</div>
</div>

<div class="support">
<a href="{{ route('support') }}"> </a>
</div>

<div class="contacts">
<a href="{{ route('contacts') }}"> </a>
</div>

<div class="account">
@auth
<a href="{{ route('profile') }}"
class="profile-link">


```

```

        </a>
        @else
        <div class="account-overlay"></div>
        <a href="#" class="account-trigger"> </a>
        <div class="account-popup">
            <!-- Login Form -->
            <form class="account-form" id="login-form" method="POST"
action="{{ route('login') }}">

                @csrf
                <h2>Вхід</h2>
                <hr>

                <label>Логін:</label>
                <input type="text"
name="login" required>

                <label>Пароль:</label>
                <input type="password"
name="password" required>

                <button
type="submit">Увійти</button>

                <p class="switch-text">
                    Немає облікового
запису?
                    <button type="button"
class="switch-to-register" id="to-register">Зареєструватися</button>
                </p>
            </form>
            <!-- Registration Form -->
            <form class="account-form" id="register-form" method="POST"
action="{{ route('register') }}">

                @csrf
                <h2>Реєстрація</h2>
                <hr>

                <label>Логін:</label>
                <input type="text"
name="login" id="register-login" required>

                <label>Пароль:</label>
                <input type="password"
name="password" id="register-password" required>

```

```

        <label>Підтвердження
пароля:</label>
        <input      type="password"
name="password_confirmation" id="register-password-confirm" required>

        <button
type="submit">Зареєструватися</button>

        <p class="switch-text">
            Вже є обліковий запис?
            <button      type="button"
class="switch-btn" id="switch-to-login">Увійти</button>
        </p>
    </form>
    <button class="close-btn">✕</button>
</div>

    @endauth
</div>
</div>
</div>
</div>
<div class="container">
    <div class="catalogs">
        <div class="catalog-1">
            <a href="{{ route('laptopAndComputers') }}"> Ноутбуки та
комп'ютери</a>
            <a href="{{ route('laptopAndComputers') }}"> </a>
            <ul class="sub-catalog-link">
                <li><a href="{{ route('laptop') }}">Ноутбуки</a></li>
                <li><a href="{{ route('computers') }}">Комп'ютери</a></li>
                <li><a href="{{ route('complectsPC') }}">Комп'ютерні
комплектуючі</a></li>
            </ul>
        </div>
        <div class="catalog-2">
            <a href="{{ route('smartphonesAndElectronics') }}"> Смартфони та
електроніка</a>
            <a href="{{ route('smartphonesAndElectronics') }}"> </a>
            <ul class="sub-catalog-link">
                <li><a href="{{ route('smartphones') }}">Смартфони</a></li>
                <li><a href="{{ route('tablets') }}">Планшети</a></li>

```

```

        <li><a href="{{ route('wearable-gadgets') }}">Носимі
гаджети</a></li>
    </ul>
</div>
<div class="catalog-3">
    <a href="{{ route('goodsForGamers') }}">Товари для геймерів</a>
    <a href="{{ route('goodsForGamers') }}"> </a>
    <ul class="sub-catalog-link">
        <li><a href="{{ route('consoles') }}">Ігрові приставки</a></li>
        <li><a href="{{ route('games') }}">Ігри</a></li>
        <li><a href="{{ route('gamingPeripherals') }}">Ігрова
периферія</a></li>
    </ul>
</div>
</div>
</div>
</body>
</html>

```

**ДОДАТОК В**  
**Слайди презентації**

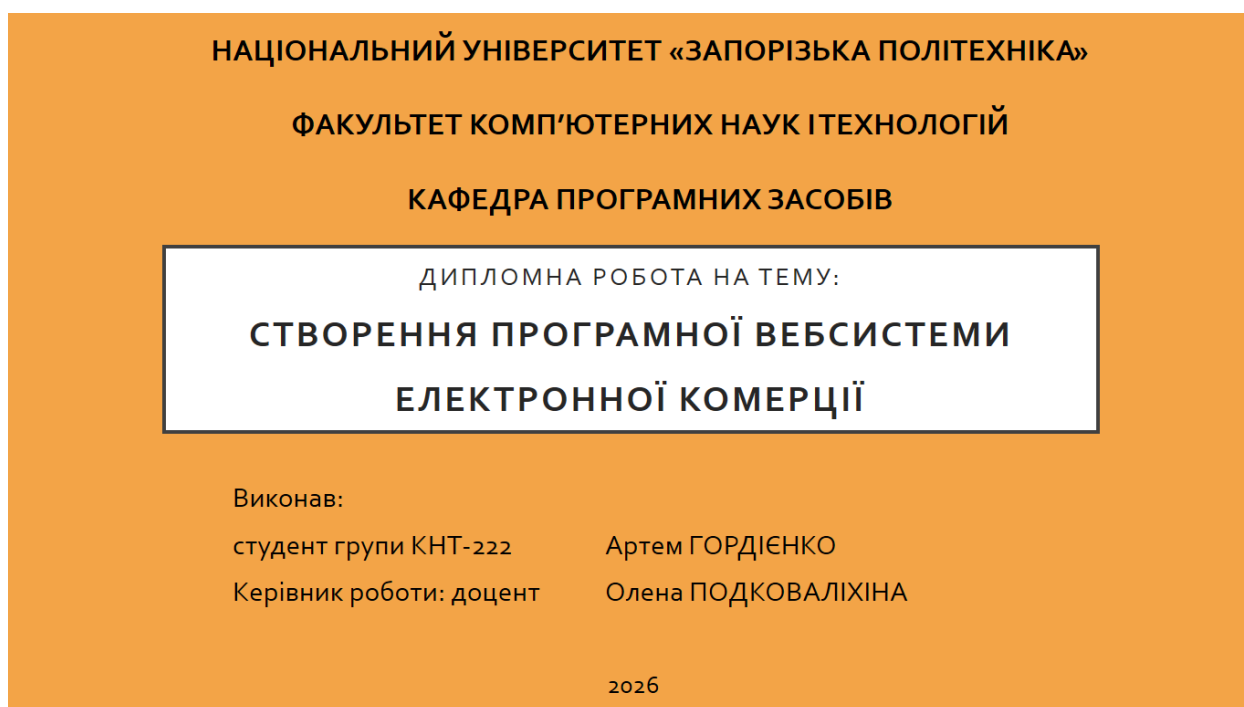


Рисунок В.1 – Перший слайд презентації

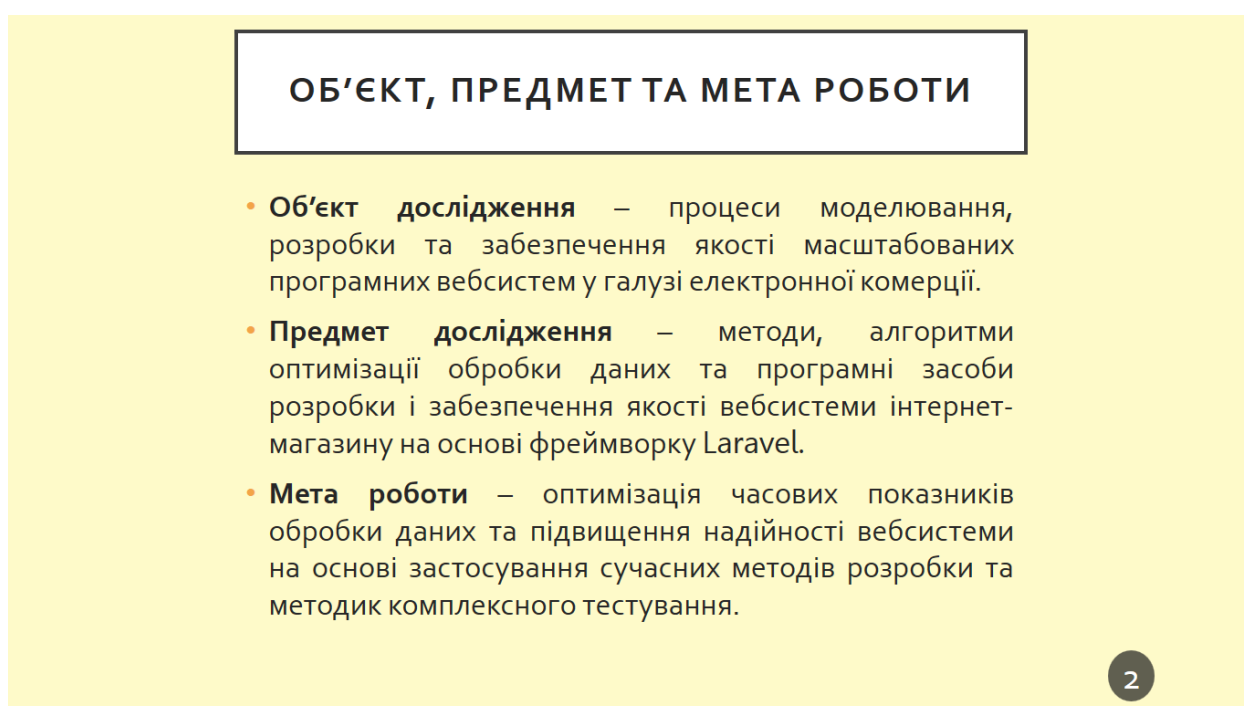


Рисунок В.2 – Другий слайд презентації

## АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ



3

Рисунок В.3 – Третій слайд презентації

## ПОРІВНЯННЯ ІСНУЮЧИХ АНАЛОГІВ

| Критерій порівняння                        | ROZETKA | COMFY    | TELEMART  | DeviceShop |
|--------------------------------------------|---------|----------|-----------|------------|
| Швидкість завантаження та зворотній відгук | Висока  | Середня  | Середня   | Висока     |
| Фокус на цільову товарну нішу              | Низький | Середній | Високий   | Високий    |
| Зручність інтерфейсу                       | Висока  | Висока   | Середня   | Висока     |
| Спеціалізована фільтрація                  | Базова  | Базова   | Просунута | Базова     |
| Наявність особистого кабінету              | Так     | Так      | Так       | Так        |

4

Рисунок В.4 – Четвертий слайд презентації

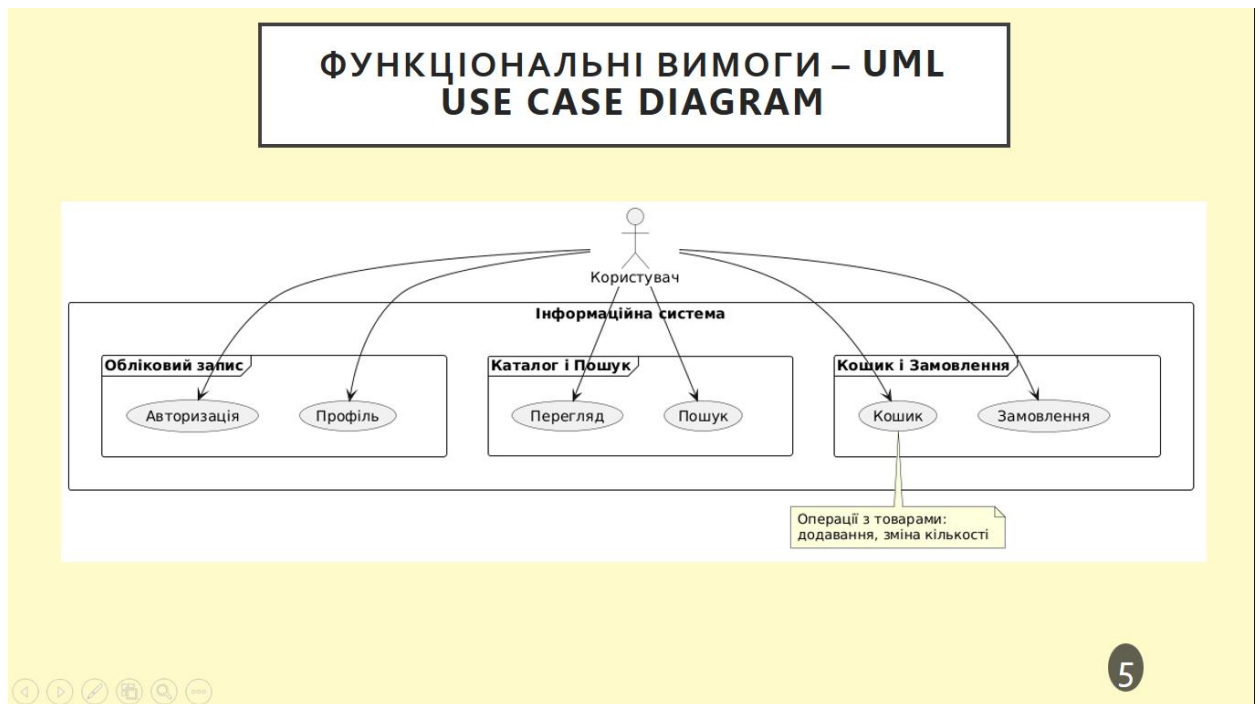


Рисунок В.5 – П'ятий слайд презентації

### НЕФУНКЦІОНАЛЬНІ ВИМОГИ

| Категорія      | Метрика                                                                                                                                                                     | Засіб реалізації                                                          |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Безпека        | Захист від SQL-ін'єкцій та XSS-атак(90% покриття)                                                                                                                           | Екранування даних у Blade-шаблонах, підготовлені запити в Eloquent ORM.   |
| Продуктивність | Час відгуку системи(до 3 секунд)                                                                                                                                            | Оптимізовані конфігураційні структури даних, вбудовані методи Laravel.    |
| Адаптивність   | <ul style="list-style-type: none"> <li>Кросбраузерна сумісність(Chrome, Safari, Firefox),</li> <li>Коректне відображення інтерфейсу на мобільних пристроях і ПК.</li> </ul> | Manual testing, Chrome DevTools, Fluid layout, Кросбраузерна верифікація. |
| Надійність     | <ul style="list-style-type: none"> <li>Показник успішних тест-кейсів(95% покриття)</li> </ul>                                                                               | Manual testing.                                                           |

6

Рисунок В.6 – Шостий слайд презентації

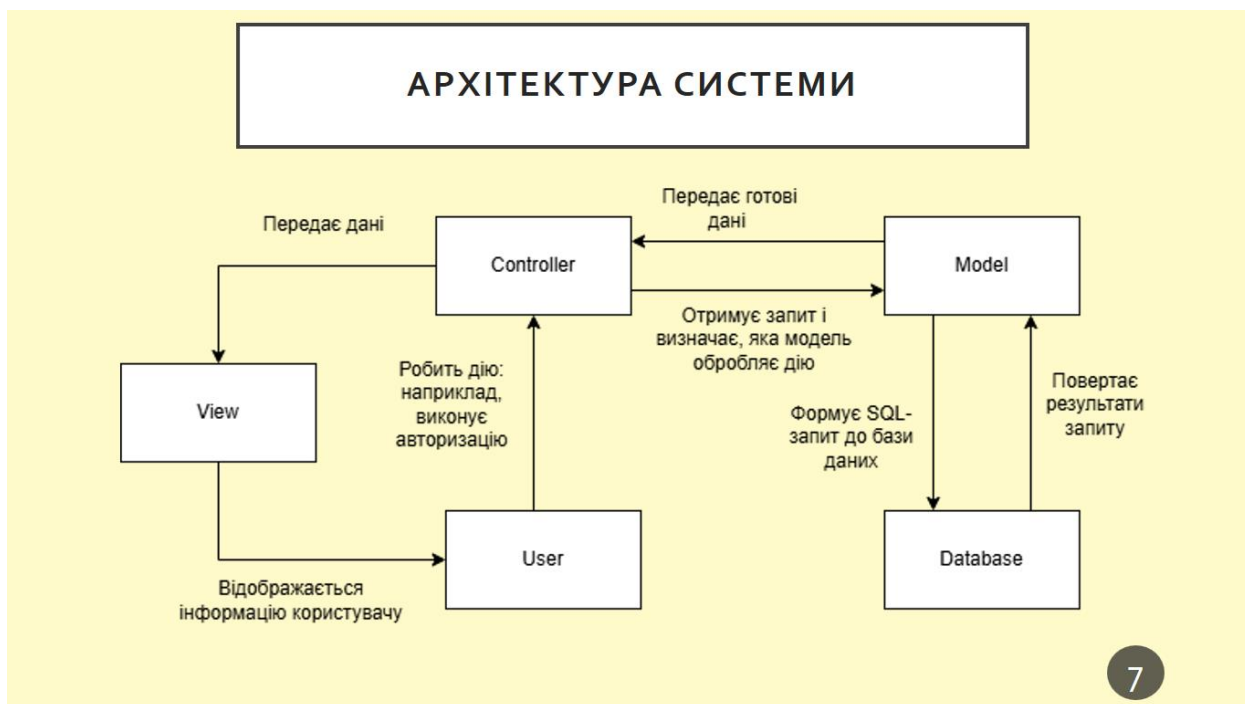


Рисунок В.7 – Сьомий слайд презентації

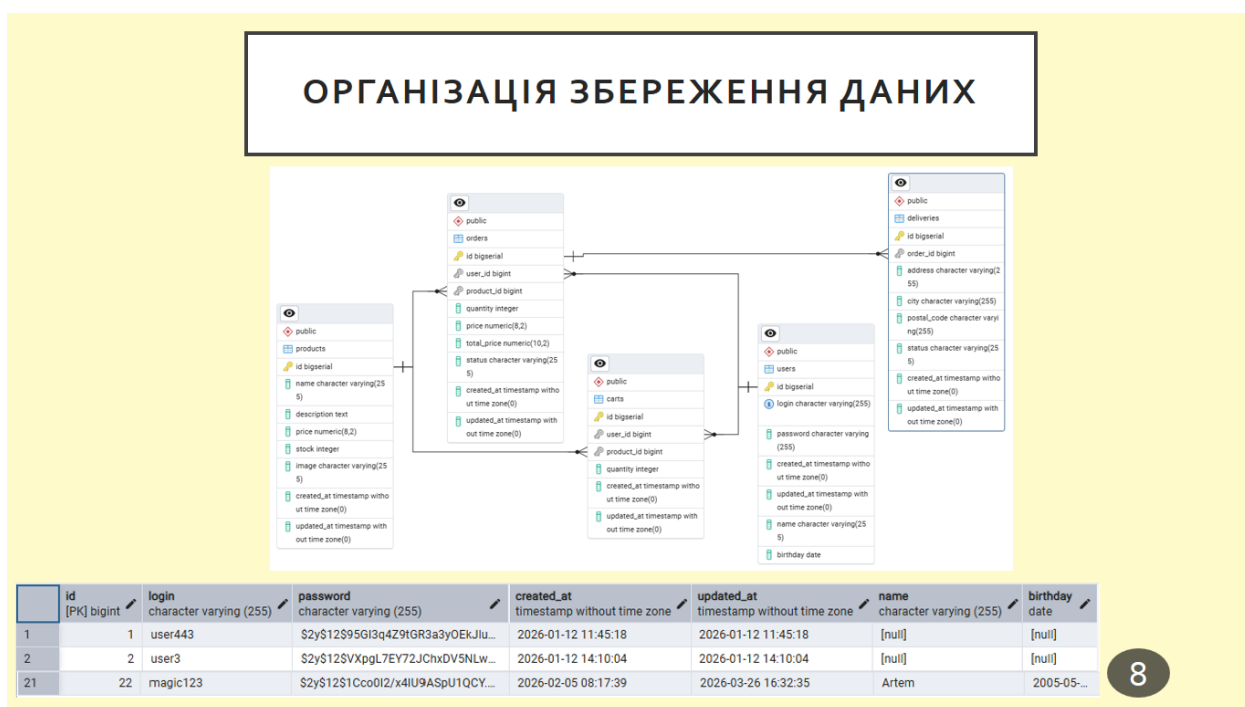


Рисунок В.8 – Восьмий слайд презентації

ІНЖЕНЕРІЯ ІНТЕРФЕЙСУ

| Компонент | Технічна реалізація                          |
|-----------|----------------------------------------------|
| Header    | Blade Templating (Layouts)                   |
| Catalog   | Hierarchical Hover Menu & CSS Implementation |
| Banner    | Flexbox-based & JS Logic                     |
| Sidebar   | Fixed Positioning & CSS Transforms           |

| HCI Метрики                | Значення показника |
|----------------------------|--------------------|
| SUS Score                  | 100/100            |
| Task Time (Search to Cart) | 12s                |

Інтернет-магазин DeviceShop: найкращі ціни тільки у нас!

КаталогПошук товарів

Навігація та комп'ютери  
Смартфони та електроніка  
Товари для геймерів

Мрійновачії  
iPhone 15  
Вже у продажу  
Всього від 42 499 грн  
Купити

Найпопулярніші товари  
Відомарта ASUS RTX 4070Ti 12GB ROG Strix OC 435638  
Геймерська клавіатура MSI Vigor GK50 LOW PROFILE 34998  
Мобільний телефон Apple iPhone 15 128GB Black 424998  
Смарт-годинник Xiaomi Redmi Watch 5 Active Black 14998

Головна сторінка

9

Рисунок В.9 – Дев’ятий слайд презентації

ПРОЄКТУВАННЯ ІНТЕРФЕЙСІВ

| Етап проектування | Мета                           | Результат                               |
|-------------------|--------------------------------|-----------------------------------------|
| Варфрейм          | Оптимізація User Flow          | Скорочення шляху до покупки до 3 кліків |
| Мокап             | Тестування візуальної ієрархії | Sus Score – 100/100                     |

Слоган

МенюЛоготипПошукКнопкаКошикВолодінняПитанняПрофіль

Питання, що часто задаються

Питання та відповідь

Питання та відповідь

Форма зворотнього зв'язку

Ім'я

Електронна пошта або номер телефону

Тема звернення

Текст повідомлення

Інтернет-магазин DeviceShop: найкращі ціни тільки у нас!

КаталогПошук товарів

Рекламний банер

Найпопулярніші товари

ФотоТекстФотоТекстФотоТекст

Варфрейм сторінки «Довідниковий центр»

Мокап головної сторінки

10

Рисунок В.10 – Десятий слайд презентації

## МОДЕЛЬ ОЦІНКИ ЧАСУ ОБРОБКИ ЗАМОВЛЕНЬ

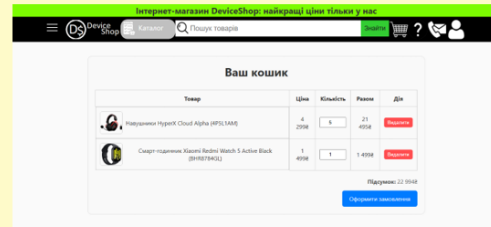
Математична модель оцінки часу:

$$T_{total} = T_{auth} + \sum_{i=1}^n T_{proc}(p_i)$$

$T_{total}$  — **Total Time** сумарний час виконання операції

$T_{auth}$  — **Authentication/Authorization Time** час на верифікацію користувача

$T_{proc}(p_i)$  — час обробки та завантаження даних для  $i$ -го об'єкта (товару)



11

Рисунок В.11 – Одинадцятий слайд презентації

## ФРАГМЕНТ КОДУ: АЛГОРИТМ ВЗАЄМОДІЇ З КОШИКОМ

Контролер для взаємодії з кошиком і товаром у ньому

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class CartController extends Controller
{
    // Показати кошик
    public function index()
    {
        $cart = session()->get('cart', []);
        return view('cart', compact('cart'));
    }

    // Додати товар в кошик через AJAX
    public function add(Request $request, $productId)
    {
        $products = config('products');

        if (!isset($products[$productId])) {
            return response()->json(['error' => 'Товар не знайдено'], 404);
        }

        $cart = session()->get('cart', []);
```

```
        if (isset($cart[$productId])) {
            $cart[$productId]['quantity'] += 1;
        } else {
            $cart[$productId] = [
                'name' => $products[$productId]['name'],
                'price' => $products[$productId]['price'],
                'image' => $products[$productId]['image'],
                'quantity' => 1,
            ];
        }

        session()->put('cart', $cart);

        return response()->json([
            'success' => true,
            'message' => 'Товар додано до кошика',
            'cart_count' => count($cart),
        ]);
    }

    // Оновити кількість товару
    public function update(Request $request, $productId)
    {
        $cart = session()->get('cart', []);

        if (isset($cart[$productId])) {
            $quantity = max(1, (int) $request->quantity);
            $cart[$productId]['quantity'] = $quantity;
            session()->put('cart', $cart);
        }

        return response()->json(['success' => true, 'message' => 'Кількість оновлено!']);
    }
}
```

```
        return response()->json(['error' => 'Товар не знайдено в кошику'], 404);
    }

    // Видалити товар з кошика
    public function remove(Request $request, $productId)
    {
        $cart = session()->get('cart', []);

        if (isset($cart[$productId])) {
            unset($cart[$productId]);
            session()->put('cart', $cart);

            return response()->json(['success' => true, 'message' => 'Товар видалено з кошика!']);
        }

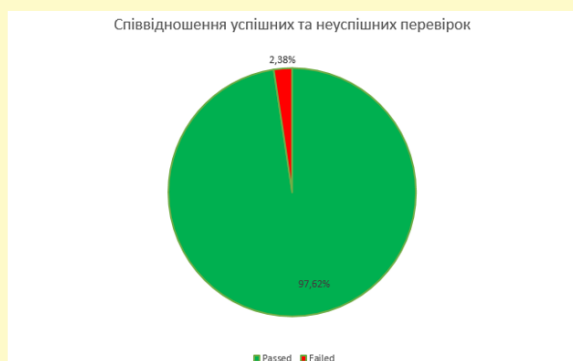
        return response()->json(['error' => 'Товар не знайдено в кошику'], 404);
    }

    // Очистити весь кошик
    public function clear()
    {
        session()->forget('cart');
        return response()->json(['success' => true, 'message' => 'Кошик очищено!']);
    }
}
```

12

Рисунок В.12 – Дванадцятий слайд презентації

## ТЕСТУВАННЯ РОЗРОБЛЕНОЇ ВЕБСИСТЕМИ



| Результати перевірок        | Відсоток виконання |
|-----------------------------|--------------------|
| Успішність                  | 97,62%             |
| Функціональність            | 100% (15 із 15 )   |
| Юзабіліті                   | 100% (8 із 8)      |
| Безпека                     | 100% (5 із 5)      |
| Верстка та кросбраузерність | 92,86% (13 із 14 ) |

13

Рисунок В.13 – Тринадцятий слайд презентації

## ПОРІВНЯЛЬНИЙ АНАЛІЗ ЧАСУ ВІДГУКУ ВЕБСИСТЕМ (У СЕКУНДАХ)

| Операція користувача           | ROZETKA | COMFY | TELEMART | DeviceShop |
|--------------------------------|---------|-------|----------|------------|
| Відкриття кошика з товарами    | 4,03    | 13,16 | 2,52     | 1,85       |
| Редагування персональних даних | 1,73    | 1,59  | 1,67     | 1,32       |

14

Рисунок В.14 – Чотирнадцятий слайд презентації

## ВИСНОВКИ

- У ході виконання кваліфікаційної роботи було розроблено функціональну вебсистему для електронної комерції та реалізовано комплекс заходів із забезпечення її якості.
- Головної мети проєкту було досягнуто шляхом детального аналізу вимог та проєктування гнучкої архітектури:
  - 1) скорочення часу обробки користувацьких запитів на перегляд товарів в кошику у 1,36-7,1 рази за рахунок використання оптимізованих конфігураційних структур даних;
  - 2) стабільна швидкість виконання запитів та скорочення часу обробки персональної інформації користувачів в 1,3 рази завдяки використанню вбудованих методів Laravel та оптимізованій взаємодії з БД PostgreSQL;
  - 3) збільшення захищеності від типових вразливостей (SQL-ін'єкцій та XSS-атак) до 95% за рахунок автоматичного екранування даних у Blade-шаблонах та використання підготовлених запитів в Eloquent ORM;
  - 4) зручність супроводу програмного коду: скорочення об'єму коду в 2 рази завдяки використанню архітектури MVC та вбудованих функцій фреймворку.
- Проведене комплексне тестування підтвердило повну відповідність системи встановленим вимогам з рівнем надійності 97,62% успішно пройдених пунктів чекліста. Своєчасне виявлення та усунення дефектів кросбраузерної сумісності дозволило отримати цілісний продукт, який демонструє стабільну роботу, має інтуїтивний інтерфейс та володіє потенціалом для подальшого масштабування під реальні потреби бізнесу.

15

Рисунок В.15 – П'ятнадцятий слайд презентації