

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

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-сайту інтернет-магазину «Unic_Mag»

Виконав: студент IV курсу, групи СН-43

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Ягленко Д.С.

(прізвище та ініціали)

Керівник

(підпис)

Марценко С.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2025

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

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» червня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

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

Студенту Ягленку Денису Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-сайту інтернет-магазину «Unic_Mag»

Керівник роботи Марценко Сергій Володимирович, кандидат технічних наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» травня 2025 року № 4/7-444

2. Термін подання студентом завершеної роботи 23 червня 2025 р.

3. Вихідні дані до роботи

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. РОЗДІЛ 1. Аналіз предметної області, 1.1 Поняття інтернет-магазину та його роль у сучасній економіці, 1.2 Класифікація та структура інтернет-магазинів, 1.3 PHP як мова програмування для розробки інтернет-магазинів, 1.4 Огляд Laravel та Laragon, 1.4.1 Laravel, 1.4.2 Laragon, 1.5 Аналіз функціоналу конкурентів, 1.6 Висновок до першого розділу, РОЗДІЛ 2. Проектування веб-сайту, 2.1 Визначення функціональних та нефункціональних вимог, 2.2 Проектування структури сайту, 2.3 Архітектура та вибір технологій, 2.4 Проектування бази даних, 2.5 Ролі користувачів та система авторизації, 2.6 Доменне ім'я та хостинг, 2.7 Висновок до другого розділу, РОЗДІЛ 3. Реалізація проекту, 3.1 Налаштування середовища розробки, 3.2 Реалізація базової структури сайту, 3.3 Реалізація функціоналу користувача, 3.4 Реалізація адміністративної панелі, 3.5 Робота з базою даних через Eloquent ORM, 3.6 Захист і валідація даних, 3.7 Інтеграція з поштою та повідомленнями, 3.8 Публікація сайту на хостингу, 3.9 Висновок до третього розділу, РОЗДІЛ 4. Безпека життєдіяльності, основи охорони праці, 4.1 Ергономічні проблеми безпеки життєдіяльності, 4.2 Економічне значення заходів щодо покращення умов та охорони праці, 4.3 Планування робіт щодо охорони праці, 4.4 Висновок до четвертого розділу. Висновки. Перелік джерел
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

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

7. Дата видачі завдання 27 червня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

Ягленко Д.С.

(прізвище та ініціали)

Керівник роботи

(підпис)

Марценко С.В.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка веб-сайту інтернет-магазину «Unic_Mag» // Кваліфікаційна робота освітнього рівня «Бакалавр» // Ягленко Денис Сергійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-43 // Тернопіль, 2025 // С. 62, рис. – 14, табл. – 4, кресл. – 0, додат. – 0, бібліогр. – 41.

Ключові слова: інтернет-магазин, РНР, сайт, база даних, адаптивність, безпека, розробка..

Кваліфікаційна робота присвячена розробці веб-сайту інтернет-магазину з використанням мови програмування РНР та пов'язаних веб-технологій.

Мета роботи: створити з нуля функціональний інтернет-магазин, який відповідає вимогам технічного завдання, включає базовий функціонал керування товарами, адаптований під мобільні пристрої та має адміністративну частину для управління вмістом.

У першому розділі кваліфікаційної роботи розглянуто типи сайтів, можливості РНР у веб-розробці, проведено аналіз аналогічних рішень та сформовано вимоги до сайту.

У другому розділі описано побудову структури сайту, взаємодію з базою даних, реалізацію авторизації та загальні принципи роботи з користувачами.

У третьому розділі розглянуто процес розробки інтернет-магазину, реалізацію адміністративної панелі, функціонал для керування товарами та адаптацію сайту під мобільні пристрої.

У розділі з безпеки розглянуто основні ризики для здоров'я програміста та заходи щодо створення безпечного робочого середовища.

ANNOTATION

Development of the «Unic_Mag» Online Store Website // Qualification work of the educational level «Bachelor» // Yahlenko Denys // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-43 // Ternopil, 2025 // P. 62, fig. – 14, tabl. – 4, chair. – 0, annexes. – 0, references – 41.

Keywords: online store, PHP, website, database, adaptability, security, development.

The qualification work is devoted to the development of an online store website using the PHP programming language and related web technologies.

The purpose of the work is to create a functional online store from scratch that meets the requirements of the technical task, includes basic product management functionality, is adapted for mobile devices, and has an administrative part for content management.

The first chapter of the qualification work describes the types of websites, PHP capabilities in web development, analyzes similar solutions, and formulates the requirements for the website.

The second section describes the construction of the site structure, interaction with the database, authorization implementation, and general principles of working with users.

The third section describes the process of developing an online store, the implementation of the administrative panel, the functionality for managing goods, and the adaptation of the site for mobile devices.

The security section describes the main risks to programmer's health and measures to create a safe working environment.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних.

CMS – система керування вмістом – платформа для створення та адміністрування сайтів.

CSS – каскадні таблиці стилів.

SEO – оптимізація сайту для пошукових систем.

URL – адреса веб-ресурсу в інтернеті.

UI/UX – інтерфейс користувача / досвід користувача.

CRUD – базові операції з даними: створення, читання, оновлення, видалення.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Поняття інтернет-магазину та його роль у сучасній економіці	10
1.2 Класифікація та структура інтернет-магазинів	11
1.3 PHP як мова програмування для розробки інтернет-магазинів.....	14
1.4 Огляд Laravel та Laragon.....	16
1.4.1 Laravel.....	16
1.4.2 Laragon.....	17
1.5 Аналіз функціоналу конкурентів	18
1.6 Постановка задачі	20
1.7 Висновок до першого розділу	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБ-САЙТУ	23
2.1 Визначення функціональних та нефункціональних вимог	23
2.2 Проєктування структури сайту	24
2.3 Архітектура та вибір технологій.....	26
2.4 Проєктування бази даних	26
2.5 Ролі користувачів та система авторизації	29
2.6 Доменне ім'я та хостинг	30
2.7 Висновок до другого розділу	32
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЄКТУ	34
3.1 Налаштування середовища розробки.....	34
3.2 Реалізація базової структури сайту	35
3.2.1 Головна сторінка	37
3.2.2 Сторінка каталогу товарів.....	38
3.3 Реалізація функціоналу користувача.....	39
3.4 Реалізація адміністративної панелі.....	40
3.5 Робота з базою даних через Eloquent ORM	42
3.6 Захист і валідація даних.....	44

3.7 Інтеграція з поштою та повідомленнями	46
3.8 Публікація сайту на хостингу	47
3.9 Висновок до третього розділу	49
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	50
4.1 Ергономічні проблеми безпеки життєдіяльності	50
4.2 Економічне значення заходів щодо покращення умов та охорони праці	51
4.3 Планування робіт щодо охорони праці	52
4.4 Висновок до четвертого розділу	55
ВИСНОВКИ	56
ПЕРЕЛІК ДЖЕРЕЛ	58

ВСТУП

Актуальність теми. Внаслідок глобалізації та стрімкого розвитку цифрових технологій дедалі більше підприємств і споживачів переходять до використання електронної комерції. Інтернет-магазини стали не лише ефективним каналом продажу товарів, а й важливим інструментом побудови бренду, формування лояльності споживачів і розширення ринків збуту. За останнє десятиліття значно змінилися споживчі звички: користувачі очікують від веб-сервісів зручного інтерфейсу, швидкого доступу до інформації, мобільної адаптивності та високого рівня безпеки при здійсненні покупок.

Паралельно з цим зростає конкуренція між компаніями, які прагнуть створити максимально ефективні онлайн-платформи. Це зумовлює потребу у фахівцях, здатних створювати індивідуальні, функціональні та масштабовані веб-рішення. Однією з найбільш доступних і гнучких мов програмування, що використовується в розробці серверної частини веб-сайтів, є PHP. Завдяки відкритому коду, широкій спільноті, безлічі фреймворків та можливості інтеграції з базами даних, PHP залишається популярним вибором для створення інтернет-магазинів, особливо для малого та середнього бізнесу.

Таким чином, розробка веб-сайту інтернет-магазину з використанням PHP є актуальним напрямком сучасних досліджень в галузі інформаційних технологій, що поєднує у собі прикладне програмування, веб-дизайн, питання UI/UX та електронну комерцію.

Мета і задачі дослідження. Метою цієї кваліфікаційної роботи є підвищення якості послуг у сфері електронної комерції шляхом створення функціонального, зручного та масштабованого веб-сайту інтернет-магазину із застосуванням мови програмування PHP.

Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- проаналізувати аналоги веб-сайтів конкурентів в якості натхнення;

- обґрунтувати вибір технологічного інструментарію для реалізації веб-сайту;
- спроектувати структуру та архітектуру системи;
- реалізувати основний функціонал інтернет-магазину (каталог товарів, кошик, оформлення замовлення, обліковий запис користувача);
- протестувати готову систему та проаналізувати її ефективність [36].

Практичне значення одержаних результатів. Практичне значення одержаних результатів полягає у створенні робочого прототипу веб-сайту інтернет-магазину домашнього взуття, який може бути використаний як основа для впровадження в діяльність малого чи середнього бізнесу. Розроблене рішення демонструє можливості використання РНР у побудові зручних і адаптивних веб-систем, що відповідають сучасним вимогам електронної комерції.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття інтернет-магазину та його роль у сучасній економіці

Інтернет-магазин — це спеціалізований веб-сайт, який забезпечує можливість дистанційного продажу товарів або послуг через мережу Інтернет. На відміну від інформаційних сайтів, котрі мають переважно ознайомлювальну функцію, інтернет-магазини дозволяють здійснювати повноцінну комерційну діяльність: перегляд товарів, формування замовлення, його оплата та організація доставки [1].

У сучасному глобалізованому світі, де цифрові технології охоплюють усі сфери життя, електронна комерція стала невід’ємною складовою бізнес-процесів. Згідно зі звітом аналітичної компанії Statista, обсяг світового ринку e-commerce у 2023 році перевищив 6 трильйонів доларів США, і продовжує зростати [2]. Все більше компаній включно з малим та середнім бізнесом, інтегрують онлайн-продажі у свою діяльність, адже це дозволяє розширити клієнтську базу, знизити витрати на оренду торгових площ і забезпечити безперервний доступ до товарів для покупців.

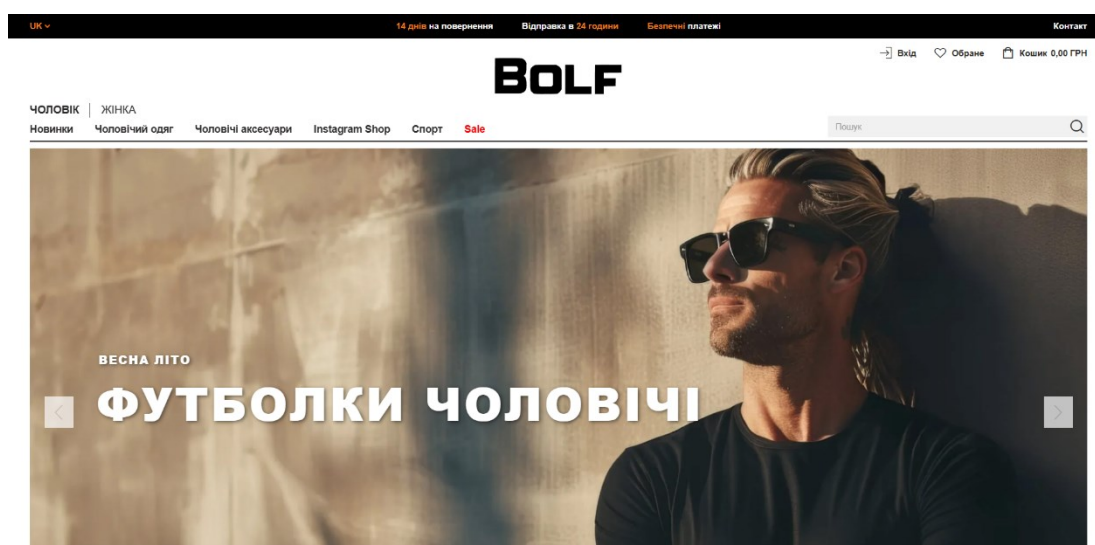


Рисунок 1.1 – Приклад інтернет-магазину одягу

Інтернет-магазини мають низку переваг:

- Доступність 24/7 – користувач може оформити замовлення у будь-який час;
- Гнучкість – можливість швидко оновлювати інформацію про товари;
- Зменшення витрат – у порівнянні з традиційною роздрібною торгівлею;
- Розширення ринку збуту – охоплення клієнтів незалежно від їхнього місця проживання;
- Інтеграція з аналітикою та CRM – дозволяє ефективно керувати маркетингом і клієнтською базою.

Водночас розробка інтернет-магазину – це не просто створення вітрини товарів. Це складна інформаційна система, яка повинна відповідати сучасним вимогам до юзабіліті, адаптивності, безпеки, продуктивності та масштабованості. У зв'язку з цим надзвичайно важливим є правильний вибір архітектури, технологій та інструментів розробки.

Науковці визначають інтернет-магазин як «інформаційно-комунікаційну систему, яка дозволяє автоматизувати процеси купівлі-продажу та взаємодії з клієнтами на базі веб-технологій» [3].

1.2 Класифікація та структура інтернет-магазинів

Інтернет-магазини, як невід'ємна складова електронної комерції, відіграють важливу роль у сучасній економіці, забезпечуючи зручний та ефективний спосіб купівлі та продажу товарів і послуг через мережу Інтернет. Вони можуть істотно відрізнятися між собою за цілями, функціональністю, технічним виконанням, масштабом діяльності, цільовою аудиторією, асортиментом продукції, способом організації логістики та іншими параметрами. Така різноманітність обумовлює необхідність класифікації інтернет-магазинів для кращого розуміння їхньої структури, принципів функціонування та підходів до розробки. Тому в науковій та практичній

літературі поширеним є підхід до класифікації інтернет-магазинів за низкою ознак.

Класифікація інтернет-магазинів:

1. За моделлю взаємодії [4] (табл. 1.1);
2. За форматом розгортання (табл.1.2).

Таблиця 1.1 Інтернет-магазини за моделлю взаємодії

B2C (Business-to-Customer)	B2B (Business-to-Business)	C2C (Customer-to-Customer)
Класична модель, коли компанія продає товари кінцевим споживачам (приклади: Rozetka, Allo)	Платформи для оптової торгівлі або взаємодії між підприємствами	Онлайн-майданчики, що дають змогу користувачам продавати товари один одному (наприклад, OLX)

На рисунку 1.2 представлено вигляд сайту моделі B2B:

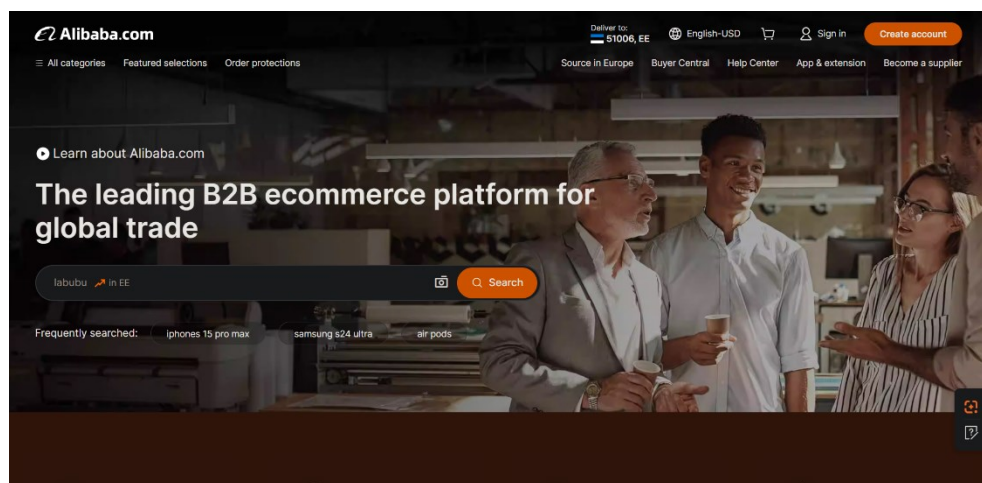


Рисунок 1.2 – Приклад сайту B2B

На рисунку 1.3 є вигляд сайту моделі B2C:

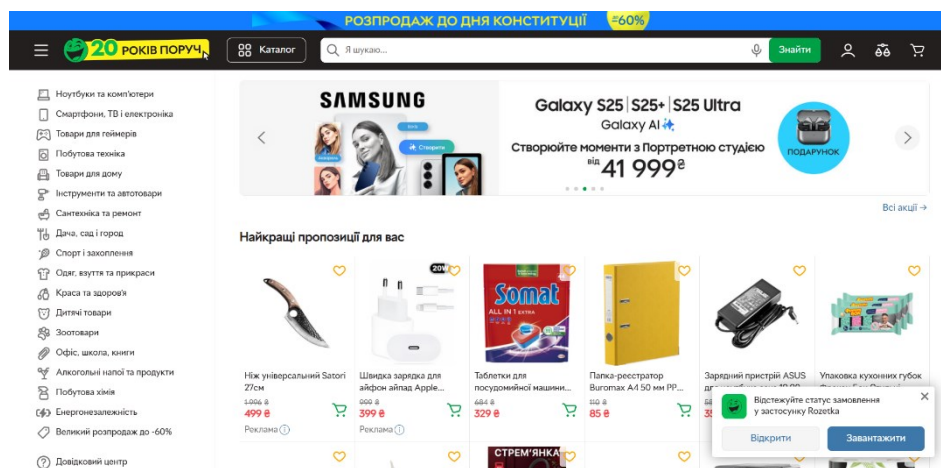


Рисунок 1.3 – Приклад сайту B2C

Вигляд сайту моделі C2C можна переглянути на рисунку 1.4:

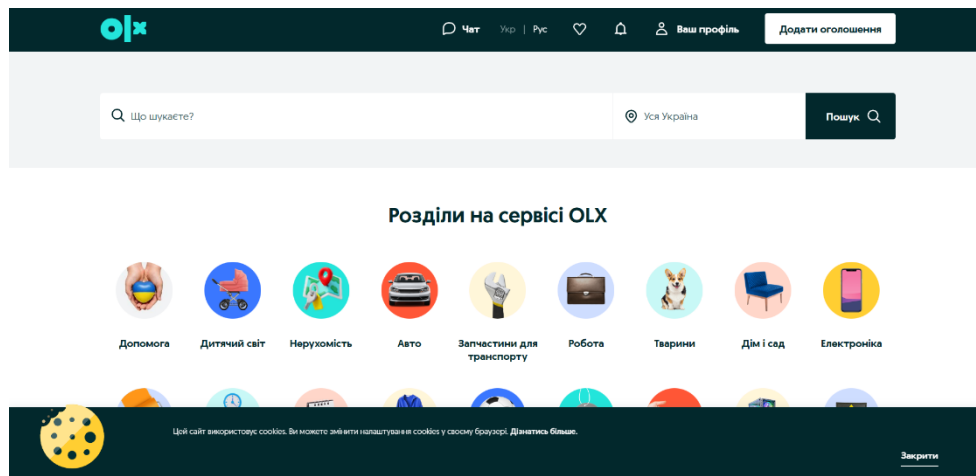


Рисунок 1.4 – Приклад сайту C2C

Таблиця 1.2 Інтернет-магазини за форматом розгортання

Класичні	Маркетплейси	Конструктори/платформи
Індивідуально розроблені сайти з повним контролем над функціоналом	Агрегатори товарів багатьох продавців (Prom.ua, Amazon)	Сервіси на кшталт Shopify або Tilda, котрі дозволяють створити магазин без програмування

З точки зору функціональності, більшість інтернет-магазинів мають подібну структуру, що охоплює такі основні компоненти:

- Головна сторінка – стартова точка з банерами, акціями, категоріями.
- Каталог товарів – список продукції з можливістю фільтрації, сортування, пошуку.
- Картка товару – детальна інформація про окремий товар.
- Кошик – тимчасове сховище обраних товарів із можливістю редагування.
- Оформлення замовлення – сторінка з формою введення даних клієнта та вибору способу оплати й доставки.
- Особистий кабінет користувача. – доступ до історії замовлень, налаштувань, адрес [35].
- Адміністративна панель – інструменти менеджера магазину або адміністратора (додавання товарів, перегляд замовлень, звіти, аналітика).
- Блок підтримки – форма зворотного зв'язку, чат, контакти.

Кожен із вказаних компонентів має виконувати свою роль у логіці взаємодії користувача з магазином, забезпечуючи зручність, зрозумілу навігацію та безперебійний процес покупки. Добре спроектована структура є основою позитивного користувацького досвіду (UX), що, у свою чергу, прямо впливає на конверсію та прибутковість ресурсу.

1.3 PHP як мова програмування для розробки інтернет-магазинів

PHP (Hypertext Preprocessor) – це скриптова мова програмування, призначена для створення динамічних веб-застосунків і веб-сайтів [5]. Вона є однією з найпоширеніших мов у галузі веб-розробки: за даними статистики W3Techs, станом на 2024 рік понад 76% веб-сайтів у світі на стороні сервера використовують PHP [6].

Однією з переваг використання PHP є зосередженість на веб-розробці. Адже PHP спочатку створювався саме як мова для розробки веб-застосунків, а отже має вбудовану підтримку роботи з HTTP-запитами, сесіями, формами,

файлами cookie, завдяки чому значно спрощується розробка повноцінного інтернет-магазину.

Завдяки простій логіці, великій кількості посібників та документації, PHP залишається доступним для студентів та розробників-початківців, не знижуючи при цьому ефективності в складних проєктах [7].

Існує велика кількість бібліотек, плагінів, CMS та фреймворків, які пришвидшують розробку, а також форуми та спільноти, де можна отримати допомогу або знайти готове рішення [8]. Такі фреймворки, як Laravel, Symfony, Yii дозволяють реалізувати MVC-архітектуру, забезпечити гнучку структуру проєкту, масштабованість, захист даних і автоматизоване тестування.

PHP-сервери підтримуються більшістю хостинг-провайдерів у світі, включно з дешевими тарифами, що робить цю технологію доступною для малого бізнесу [10].

Із появою PHP 7 та подальших версій продуктивність мови значно зростає: впроваджено JIT-компіляцію, покращено керування пам'яттю та швидкість виконання коду [9].

PHP неідеальна мова, тож вона має свої недоліки:

- Низька якість старого коду – вирішується використанням сучасних фреймворків (Laravel, Symfony).
- Відсутність строгої типізації – частково усувається з версій PHP 7.4+ завдяки підтримці типів і класових обмежень.
- Критика за безпеку – сучасні фреймворки мають вбудовані механізми захисту від XSS, CSRF, SQL-ін'єкцій тощо.

PHP активно використовується у світових e-commerce проєктах:

- WordPress + WooCommerce – одна з найпопулярніших комбінацій для малого бізнесу.
- OpenCart, PrestaShop – готові CMS для створення інтернет-магазинів на PHP.
- Laravel – часто використовується для створення кастомних, складних магазинів із унікальним функціоналом.

Отже, PHP забезпечує оптимальний баланс між простотою, гнучкістю та функціональністю, що робить його доцільним вибором для реалізації інтернет-магазину в рамках дипломного проєкту.

1.4 Огляд Laravel та Laragon

У контексті розробки сайтів на PHP важливими інструментами для ефективного та швидкого створення проєктів є фреймворки та локальне середовище, де розгортатиметься веб-сайт. Використання фреймворків полегшить написання коду, адже не потрібно буде вигадувати велосипед, так як багато функцій та механізмів уже є в бібліотеці фреймворку. В рамках розробки веб-сайту інтернет-магазину на мові програмування PHP було обрано фреймворк Laravel та середовище Laragon. Такий вибір зумовлений їх функціональністю, зручністю, широким застосуванням у промисловій розробці та високою якістю документації.

1.4.1 Laravel

Laravel – це один з найпотужніших PHP-фреймворків, який активно використовується для розробки веб-застосунків, включаючи інтернет-магазини [11]. Його популярність обумовлена великою кількістю вбудованих інструментів та функціональностей, котрі спрощують процес розробки складних проєктів.

Laravel підтримує архітектуру MVC [14], що дозволяє чітко структурувати код і розділяти логіку роботи з базою даних, інтерфейс та користувацькі взаємодії. Це робить проєкти більш масштабованими та підтримуваними. Він використовує Eloquent ORM для роботи з базою даних, що дозволяє створювати та взаємодіяти з таблицями БД за допомогою об'єктів, що робить простішою роботу із запитами та зменшуючи кількість SQL-коду.

Цей фреймворк надає гнучку систему маршрутизації, яка дозволяє швидко налаштувати URL-шляхи і контролери для обробки запитів. Що є важливим для

реалізації різних функцій інтернет-магазину (каталог товарів, кошик, замовлення). Також Laravel має вбудовану систему аутентифікації та авторизації, що дозволяє швидко реалізувати процеси реєстрації, входу користувачів, управління ролями (адмін, користувач) без необхідності писати код для цих функцій самотужки.

Laravel використовує шаблонізатор Blade, котрий дозволяє розділяти структуру HTML і бізнес-логіку, завдяки чому код стає чистим і зручним для підтримки. Blade дозволяє легко працювати з умовними конструкціями, циклами та даними, що є дуже зручним при створенні динамічних сторінок веб-сайту.

Laravel підтримує систему міграцій для зручної роботи з базами даних, а також має потужну підтримку юніт-тестування, що дозволяє забезпечити якість коду та коректну роботу усіх компонентів інтернет-магазину.

Важлива складова цього фреймворку – підтримка безпеки. Laravel забезпечує вбудовані механізми захисту від SQL-ін'єкцій, CSRF, XSS, що є надзвичайно важливим для веб-сервісу, де зберігаються особисті та платіжні дані користувачів [20].

1.4.2 Laragon

Laragon – це локальне середовище для розробки на PHP, яке дозволяє швидко налаштувати сервер для веб-розробки на Windows [12].

Це середовище дозволяє за лічені хвилини створити локальний сервер з Apache, Nginx, MySQL, PHP та іншими необхідними інструментами без складних налаштувань, що робить його ідеальним для швидкого старту розробки.

Крім стандартних стеків PHP, Laragon підтримує також Node.js, Python, Ruby, що дає змогу використовувати різні інструменти для розробки сучасних веб-застосунків.

До того ж всіма необхідними інструментами можна керувати через простий інтерфейс: запускати сервіси, налаштовувати бази даних, переглядати логи та статистику роботи сервера.

Laragon дозволяє легко налаштувати локальні сайти з HTTPS, що є важливим для тестування безпеки інтернет-магазину.

За допомогою інструмента Composer можна швидко встановлювати PHP-бібліотеки та фреймворки, що значно спрощує управління залежностями проєкту.

До основних переваг Laragon належать:

- Автоматична генерація віртуальних хостів (наприклад, `unicmag.test`) — що дозволяє працювати з проєктом за адресою, подібною до реального доменного імені;
- Швидке створення нових проєктів — через команду `laravel new`, або відповідне контекстне меню;
- Ізольоване середовище — кожен проєкт може мати власну версію PHP, Composer-залежностей тощо;
- Підтримка SSL (HTTPS) — дозволяє імітувати захищене з'єднання під час локального тестування, що особливо актуально для платіжних систем;
- Менеджер служб — через інтерфейс Laragon можна легко запускати, зупиняти або перезапускати компоненти серверного середовища без потреби звертатися до командного рядка.

1.5 Аналіз функціоналу конкурентів

При розробці будь-якого продукту важливим підготовчим етапом є аналіз конкурентів у цій сфері. Це дозволяє оглянути вже готові рішення та визначити їхні позитивні сторони для натхнення та негативні сторони задля уникнення подібних помилок. Для визначення ринкових тенденцій та формування оптимального функціоналу майбутнього інтернет-магазину було здійснено аналіз двох конкурентних онлайн-платформ. Обидва ресурси орієнтовані на

реалізацію домашнього взуття, проте мають різні підходи до організації торгових процесів, дизайну інтерфейсу та взаємодії з цільовою аудиторією.

Сайт LD Карсі (див. рисунок 1.5) спеціалізується на продажі різного роду взуття: від домашнього до вуличного. До сильних сторін цього ресурсу можна віднести чітку структуру каталогу, поділ на чоловічі, жіночі та дитячі моделі, а також наявність фільтрів, що дозволяють сортувати товари за основними параметрами – ціною, розміром та кольором. Картки товарів мають доволі інформативний вигляд: до кожного товару додається фотогалерея, опис матеріалів і поради з догляду. Крім того на сайті є окремий розділ з акційними пропозиціями, що стимулює продажі.

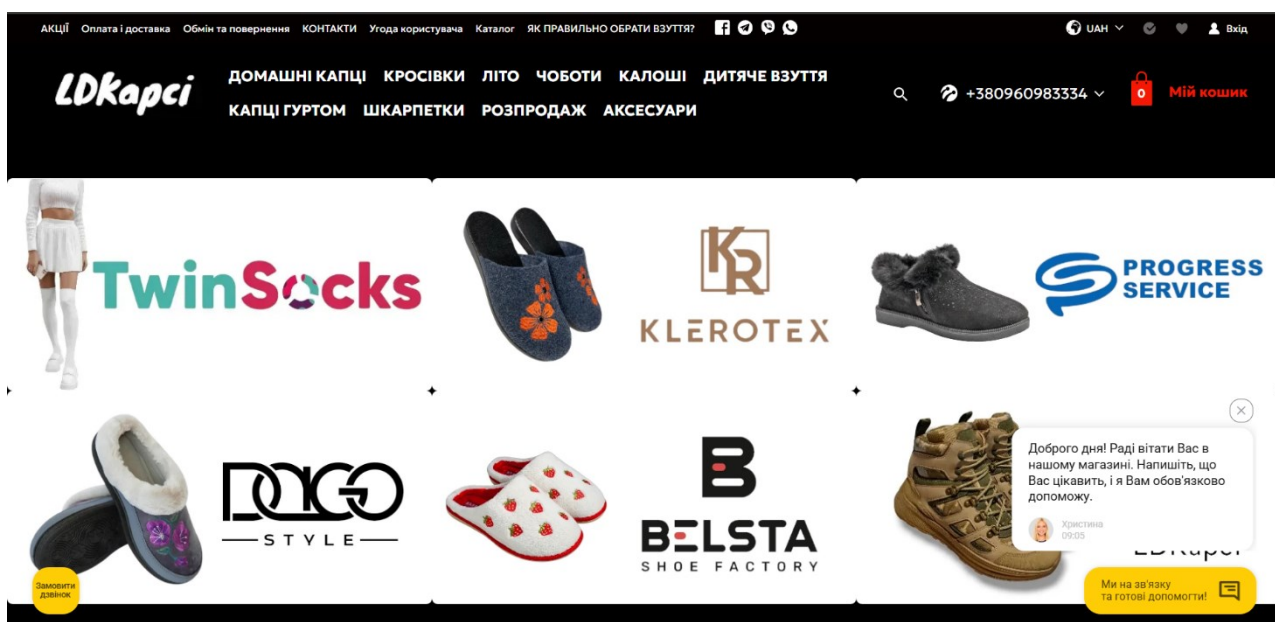


Рисунок 1.5 – Сайт LD Карсі

Проте недоліки цього інтернет-магазину також очевидні. Зокрема, візуальне оформлення є застарілим і не відповідає сучасним стандартам дизайну, що негативно впливає на загальне враження користувача. Також відсутня система відгуків, що позбавляє потенційних клієнтів змоги ознайомитись із досвідом попередніх покупців. Інформація щодо оплати та доставки є поверхневою та не завжди зрозумілою, що може викликати недовіру з боку нових користувачів.

Сайт Belsta (див. рисунок 1.6) також пропонує широкий асортимент взуття. Основною перевагою Belsta є сучасний адаптивний дизайн, який коректно відображається на мобільних пристроях і забезпечує позитивний користувацький досвід [26]. Навігація на сайті інтуїтивно зрозуміла: присутній зручний поділ за категоріями, можливість швидкого пошуку за розміром або призначенням. До кожного товару додаються відгуки покупців, що підвищує довіру бренду. Додатковими плюсами є безкоштовна доставка при великому замовленні та наявність блогу з корисними порадами щодо вибору взуття.

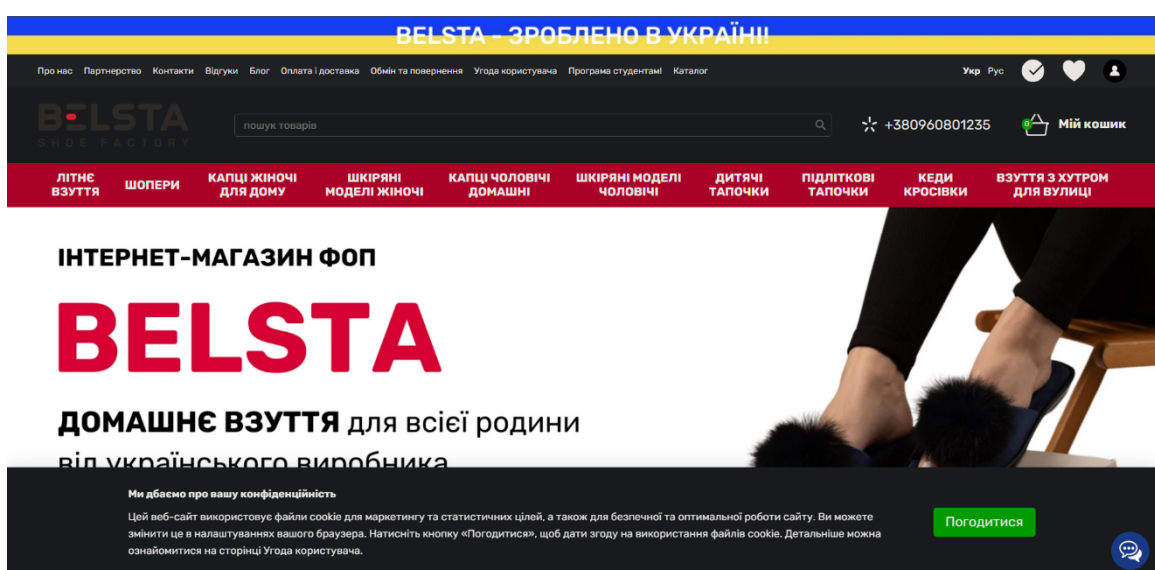


Рисунок 1.6 – Сайт Belsta

Втім, і на цьому сайті існують недоліки. У деяких позиціях опис товару є надто лаконічним, без вказівки матеріалів або характеристик, а в деяких категоріях спостерігається дефіцит доступних моделей, що свідчить про нерівномірність асортименту.

1.6 Постановка задачі

На сучасному етапі розвитку електронної комерції інтернет-магазин є невід’ємною складовою успішного бізнесу. Його наявність дозволяє

підприємству автоматизувати процеси продажу, оптимізувати взаємодію з клієнтами, забезпечити зручність вибору продукції та підвищити загальний рівень обслуговування. У зв'язку з цим постає необхідність розробки повноцінного веб-сайту, який би відповідав функціональним потребам підприємства та сучасним вимогам до дизайну, безпеки й масштабованості.

Метою даної дипломної роботи є створення веб-сайту інтернет-магазину «Unic_Mag», що спеціалізується на продажу побутових аксесуарів — домашнього взуття, косметичок та органайзерів. Реалізація сайту повинна враховувати як потреби кінцевого користувача (перегляд товарів, додавання до кошика, оформлення замовлення), так і адміністративні потреби (додавання, редагування, видалення товарів).

Для досягнення цієї мети необхідно вирішити такі практичні задачі:

- Проаналізувати технічні засоби та інструменти, що будуть використані для реалізації сайту, зокрема обґрунтувати вибір фреймворку Laravel та середовища Laragon.
- Сформулювати вимоги до сайту, включно з функціональними (реєстрація, каталог, кошик, адміністративна панель) та нефункціональними (безпека, адаптивність, продуктивність) характеристиками.
- Створити базову структуру веб-сайту, включно з головною сторінкою, сторінкою каталогу товарів, навігаційним меню та підключенням стилів.
- Реалізувати функціонал для користувача, що дозволяє переглядати товари за категоріями, бачити детальну інформацію про товар, додавати товари до кошика та оформлювати замовлення.
- Розробити адміністративну панель, за допомогою якої адміністратор зможе керувати товарами: додавати нові, редагувати або видаляти наявні.
- Забезпечити взаємодію з базою даних через ORM Eloquent, включно з міграціями, сидерами та зв'язками між сутностями.
- Забезпечити захист даних та валідацію форм, реалізувати перевірку введених користувачем даних як на стороні сервера, так і на клієнті.

- Реалізувати базову інтеграцію з поштою або системою сповіщень, що дозволяє отримувати повідомлення про успішну реєстрацію чи замовлення.
- Виконати розгортання сайту на хостингу, з урахуванням базових вимог до безперебійної роботи сервісу.
- Оцінити ефективність розробленої системи та зробити висновки щодо доцільності її використання в реальному бізнес-середовищі.

1.7 Висновок до першого розділу

У першому розділі було розглянуто основні поняття, типи та структуру інтернет-магазинів, а також обґрунтовано вибір технологій для реалізації майбутнього веб-сайту. Проведено огляд мови програмування PHP та фреймворку Laravel для розробки проєкту. Проведено аналіз конкурентних платформ.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБ-САЙТУ

2.1 Визначення функціональних та нефункціональних вимог

Процес розробки веб-сайту інтернет-магазину передбачає чітке визначення вимог до системи, оскільки саме на цьому етапі формується уявлення про її функціональність, якість та зручність використання. Вимоги умовно поділяються на функціональні та нефункціональні, і кожна з цих груп відіграє важливу роль у побудові стабільного та конкурентоспроможного ресурсу.

2.1.1. Функціональні вимоги

Функціональні вимоги описують, що система повинна робити. У межах розробки інтернет-магазину до основних функціональних вимог належать:

- Каталог товарів: відображення продукції за категоріями, можливість фільтрації за ціною, сортування та пошуку.
- Картка товару: перегляд детальної інформації про кожну одиницю товару (назва, опис, фото, розміри, колір, ціна, наявність, догляд).
- Кошик: додавання й видалення товарів, редагування кількості, підрахунок загальної вартості.
- Оформлення замовлення: форма введення особистих даних, вибір доставки та оплати.
- Реєстрація/вхід користувача: створення облікового запису, авторизація, відновлення паролю.
- Особистий кабінет користувача: перегляд історії замовлень, зміна даних, відгуки.
- Адміністративна панель: управління товарами, категоріями, замовленнями та користувачами.
- Сторінка з умовами доставки та оплати, а також форма зворотного зв'язку або онлайн-чат.

- Система відгуків і рейтингу для товарів.

2.1.2 Нефункціональні вимоги

Нефункціональні вимоги визначають якісні характеристики системи:

- Адаптивність: сайт мусить коректно відображатися на мобільних пристроях, планшетах і десктопах.
- Продуктивність: сторінки повинні завантажуватись не довше, ніж за 2-3 секунди при стандартному інтернет-з'єднанні.
- Безпека: захист від SQL-ін'єкцій, CSRF-атак, XSS [19]; шифрування паролів [18]; контроль доступу до адмін-панелі.
- Масштабованість: можливість розширення функціоналу без необхідності переписувати код.
- Зручність навігації (UX): інтуїтивний інтерфейс, логічна структура меню та зрозумілі підказки [27].
- Пошукова оптимізація (SEO): коректна генерація мета-тегів, «чисті» URL-адреси, мікророзмітка.
- Локалізація: можливість багатомовного інтерфейсу (у перспективі).

Усі зазначені вимоги будуть враховані на етапі проектування системи, формування бази даних, побудови маршрутизації та реалізації інтерфейсу. Їх правильне визначення вже на початковому етапі забезпечить структуровану розробку та мінімізує ймовірність появи критичних помилок у подальшому.

2.2 Проектування структури сайту

Структура веб-сайту інтернет-магазину визначає логіку розміщення основних сторінок, взаємозв'язок між ними та послідовність користувацьких дій. Грамотно спроектована структура дозволяє забезпечити зручну навігацію, швидкий доступ до потрібної інформації, а також оптимізувати роботу системи в цілому [29].

В основі розроблюваного інтернет-магазину лежить ієрархічна структура, що складається з публічної частини, котру бачить користувач, та адміністративної частини, що доступна лише авторизованим адміністраторам.

Основні сторінки публічного інтерфейсу:

- Головна сторінка. Містить навігацію по категоріях, популярні товари, інформацію про магазин.
- Каталог товарів. Відображає перелік продукції з можливістю фільтрації за ціною.
- Картка товару. Детальна сторінка з фотографіями товару, описом, характеристиками, ціною, кнопками «Додати в кошик» та «Купити зараз».
- Кошик. Список вибраних товарів із можливістю редагування кількості, видалення та переходу до оформлення замовлення.
- Оформлення замовлення. Форма введення персональних даних, вибору способу доставки й оплати.
- Сторінка реєстрації та входу. Форма створення акаунта або авторизації для зареєстрованих користувачів.
- Особистий кабінет користувача. Містить історію замовлень, можливість змінити контактні дані, залишити відгуки, вийти з облікового запису.
- Сторінки «Про нас», «Контакти», «Оплата й доставка». Інформаційні розділи з описом магазину, формами зв'язку, політикою обробки замовлень.

Сторінки адміністративної частини:

- Панель адміністратора. Головна сторінка.
- Управління товарами. Додавання, редагування, видалення товарів.
- Користувачі та ролі. Перегляд зареєстрованих користувачів, присвоєння ролей (покупець, адміністратор).
- Категорії та фільтри. Редагування структури каталогу, створення підкатегорій, налаштування доступних фільтрів.

Для зареєстрованих користувачів додається можливість переходу до особистого кабінету після авторизації. Система передбачає доступ до кошика й оформлення замовлення навіть без попередньої реєстрації.

Проектування такої структури дозволяє не лише полегшити розробку й тестування функціоналу, а й покращити UX завдяки логічній та зрозумілій навігації.

2.3 Архітектура та вибір технологій

Архітектурне планування є ключовим етапом розробки веб-сайту інтернет-магазину, оскільки визначає логіку взаємодії між компонентами системи, забезпечує зручність у підтримці та масштабуванні проєкту. У межах цієї роботи для створення сайту використано архітектурну модель MVC (Model-View-Controller) та фреймворк Laravel, що підтримує цю модель на рівні ядра.

Модель MVC дозволяє розділити логіку веб-застосунку на три незалежні компоненти:

- Model – відповідає за роботу з даними, взаємодію з базою, логіку обробки бізнес-процесів (наприклад, створення замовлення).
- View – відповідає за відображення інформації користувачу. В рамках Laravel для цього використовується шаблонізатор Blade.
- Controller – приймає запити від користувача, викликає відповідну модель, обробляє логіку і повертає результат у вигляді View.

Така архітектура забезпечує чистоту коду, модульність, зменшення дублювання та дає змогу незалежно розвивати кожен із компонентів системи.

Використання Laravel та MVC-підходу дозволяє збудувати гнучку, масштабовану й захищену систему, адаптовану до вимогу сучасної електронної комерції [15, 24]. Вибрані технології забезпечують не лише швидкість розробки, а й відповідність професійним стандартам веб-програмування.

2.4 Проектування бази даних

Одним із ключових етапів створення інтернет-магазину є проектування бази даних, адже вона зберігає всю критично важливу інформацію: користувачів,

товари, замовлення тощо. База даних повинна бути структурованою, логічно цілісною, підтримувати зв'язки між сутностями та забезпечувати ефективний пошук, фільтрацію і обробку запитів.

База даних реалізуватиметься за допомогою реляційної СУБД MySQL із використанням ORM Eloquent, що входить до складу фреймворку Laravel. Для проєктування використано ER-модель, яка описує основні сутності та зв'язки між ними.

Основні сутності баз даних:

Таблиця Users:

- Id – первинний ключ
- Name – ім'я користувача
- Email – електронна пошта
- Password – пароль
- Phone – номер телефону
- Address – адреса доставки
- Role – (admin, customer)

Таблиця Products:

- Id – первинний ключ
- Title – назва товару
- Description – опис товару
- Price – ціна
- Size – розмір
- Color – колір
- Image – фото товару
- Category – категорія товару (зовнішній ключ)

Таблиця: Categories

- Id – первинний ключ
- Name – назва категорії

Таблиця Orders:

- Id – первинний ключ
- User_id – зовнішній ключ
- Total_price – сума замовлення
- Status – статус обробки
- Payment_method – спосіб оплати
- Shipping_address – адреса доставки

Таблиця Order_Items:

- Id – первинний ключ
- Order_id – зовнішній ключ
- Product_id – зовнішній ключ
- Quantity - кількість
- Price (на момент замовлення)

Таблиця Reviews:

- Id – первинний ключ
- User_id – зовнішній ключ
- Product_id – зовнішній ключ
- Rating, comment, created_at

Типові зв'язки між таблицями:

- Один користувач може мати багато замовлень – зв'язок 1:N
- Одне замовлення містить багато позицій – зв'язок 1:N
- Один товар належить до однієї категорії – зв'язок N:1
- Один товар може мати багато відгуків – зв'язок 1:N
- Один користувач може залишати багато відгуків – зв'язок 1:N
- Таблиця order_items – типова для реалізації M:N між товарами та

замовленнями

Така модель дозволяє реалізувати всі базові сценарії інтернет-магазину: реєстрацію користувача, створення замовлення, обробку кошика, формування каталогу з фільтрами, підрахунок залишків, обробку знижок, перегляд історії

покупок та збереження відгуків. База даних адаптована для роботи з Eloquent ORM, що спрощує створення та обробку запитів до неї [22].

2.5 Ролі користувачів та система авторизації

Однією з базових вимог для сучасного інтернет-магазину є наявність механізму аутентифікації (встановлення особи користувача) та авторизації (визначення рівня доступу до функціоналу системи). Це забезпечує захист персональних даних, контроль доступу до ресурсів сайту та гнучке розмежування прав користувачів.

У межах розробки сайту передбачено дві основні категорії користувачів, що наведено у таблиці 2.1:

Таблиця 2.1 Категорії користувачів

Звичайні користувачі (покупці)	Адміністратори
Мають змогу реєструватися, входити в акаунт, переглядати каталог товарів, додавати товари в кошик, оформлювати замовлення.	Мають доступ до адміністративної панелі.
У особистому кабінеті можуть змінювати свої дані, переглядати історію замовлень, залишати відгуки на товари.	Можуть керувати товарами (проводити CRUD-операції), переглядати та обробляти замовлення, управляти категоріями товарів, переглядати відгуки користувачів.

Фреймворк Laravel надає потужні інструменти для реалізації безпечної системи входу [16] та розмежування прав доступу (див. таблицю 2.2):

Таблиця 2.2 Аутентифікація та авторизація на сайті

Аутентифікація	Авторизація
Використовується вбудована система Laravel Breeze або Laravel UI, яка забезпечує реєстрацію, вхід, вихід, скидання пароля. Усі паролі шифруються за допомогою bcrypt.	Для визначення доступу до певних розділів або маршрутів використовуються: – Middleware (наприклад, auth, admin); – Політики доступу (Policies); – Перевірка ролі в контролерах (if(auth()->user()->role === 'admin')).

Ролі зберігаються у таблиці users у вигляді role, де можливі значення 'customer', 'admin'. У майбутньому можливе розширення з виділенням ролей у окрему таблицю roles із проміжною role_user для реалізації зв'язку M:N.

Система авторизації реалізується до принципів безпеки:

- Обов'язкове підтвердження email;
- Захист від CSRF (вбудовано в Laravel);
- Обмеження доступу до особистих даних через маршрути.

Наявність ролей користувачів та правильно реалізованої авторизації – це не лише зручність у роботі з сайтом, а й основа для масштабованості, безпеки та правової відповідальності при роботі з персональними даними користувачів.

2.6 Доменне ім'я та хостинг

Розробка веб-сайту передбачає дуже важливий етап – забезпечити його постійну доступність для кінцевих користувачів у мережі Інтернет. Для цього необхідно виконати два взаємопов'язані кроки: зареєструвати доменне ім'я та обрати хостинг. Правильний вибір цих компонентів має значення як для

технічної стабільності сайту, так і для його комерційного успіху: від швидкодії ресурсу та безпеки до зручності запам'ятовування адреси та брендингу.

Домен – це унікальне символічне ім'я сайту, за яким користувачі можуть знайти його через браузер. Наприклад, unismag.shop – це доменне ім'я, де unismag є назвою бренду, а .shop – доменна зона, що вказує на приналежність до сфери електронної торгівлі.

Хостинг – це послуга з надання простору на сервері для зберігання файлів сайту (HTML, PHP, CSS, зображень тощо) і забезпечення його безперебійної роботи в мережі. Саме хостинг відповідає за те, щоб користувачі могли відкривати сайт у будь-який час і з будь-якого місця.

У межах проєкту зареєстровано доменне ім'я unismag.shop через одного з провідних українських провайдерів – Ukraine.com.ua (див. рисунок 2.1). Це домен у міжнародній зоні .shop, яка є релевантною для сайтів електронної комерції. Обране ім'я відповідає ряду важливих критеріїв:

- Коротке та запам'ятовуване – легко вводиться вручну і добре читається.
- Релевантне зонування - .shop явно вказує на онлайн-продажі, що зручно для пошукових систем.

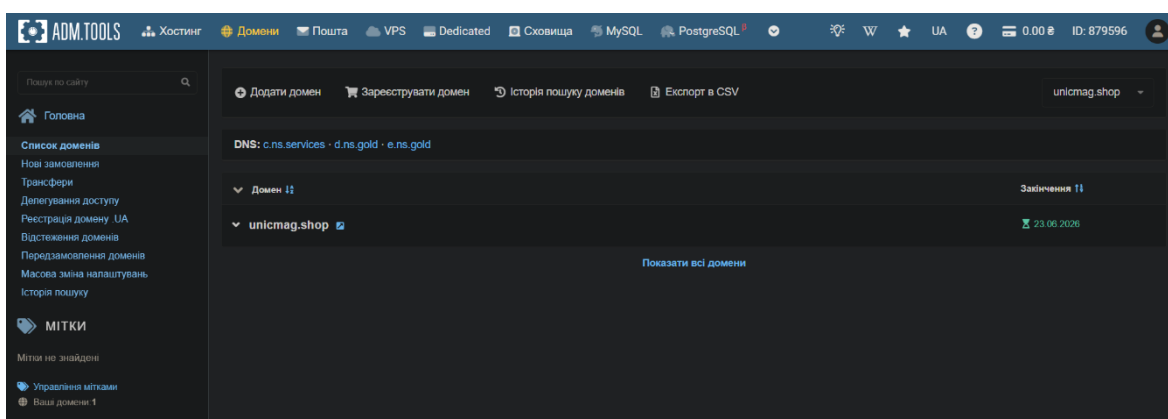


Рисунок 2.1 – Доменне ім'я на хостингу

Для розміщення проєкту [30] також використано послуги Ukraine.com.ua, які надають оптимальні технічні параметри для роботи сайту на PHP-фреймворку Laravel:

- Підтримка PHP 8.1+ та MySQL – необхідні компоненти для стабільної роботи Laravel.
- SSH-доступ – дозволяє підключення до сервера для керування проєктом через термінал.
- Підтримка Composer та Artisan – для роботи із залежностями та консоллю Laravel.
- Безкоштовний SSL-сертифікат – забезпечує захищене з'єднання (HTTPS).
- Зручна панель керування (cPanel або ISPmanager) – дозволяє керувати поштою, доменами, файлами.
- Регулярні резервні копії = захист від втрати інформації.

Обрано VPS-хостинг, що забезпечує достатню гнучкість, стабільність і контроль над середовищем виконання [31]. Це важливо для роботи з Laravel, котрий потребує більше ресурсів і гнучких налаштувань у порівнянні зі звичайним статичним сайтом.

Для публікації сайту реалізуються такі кроки [32]:

- Завантаження Laravel-проєкту на сервер через SFTP.
- Налаштування віртуального хосту на директорію public/.
- Створення та підключення бази даних через .env файл.
- Генерація ключа застосунку (php artisan key:generate).
- Кешування конфігурації (php artisan config:cache).
- Установлення SSL та налаштування редиректу з HTTP на HTTPS.

2.7 Висновок до другого розділу

В другому розділі кваліфікаційної роботи було визначено функціональні та нефункціональні вимоги до веб-сайту інтернет-магазину, обґрунтовано вибір

архітектури MVC, фреймворку Laravel, СУБД MySQL та інших інструментів. Спроектовано структуру сайту, базу даних із ключовими сутностями, реалізовано поділ ролей користувачів і описано процес розміщення проєкту на хостингу.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЄКТУ

3.1 Налаштування середовища розробки

Для розробки веб-сайту інтернет-магазину на базі мови PHP було вибрано фреймворк Laravel у поєднанні з локальним середовищем Laragon, яке забезпечує зручну, швидку та гнучку роботу зі стеком веб-технологій.

Етапи встановлення середовища:

- Завантаження Laragon – завантаження останньої версії з офіційного сайту laragon.org та встановлення за замовчуванням.

- Налаштування PHP та Composer – Laragon автоматично встановлює PHP та Composer. У разі потреби версію PHP можна змінити у налаштуваннях меню.

- Створення проєкту Laravel – у Laragon відкривається Термінал, де виконується команда:

```
composer create-project laravel/laravel unimag
```

- У результаті створюється повноцінна Laravel структура у теці

```
C:\laragon\www\unimag.
```

- Налаштування локального домену – Laragon автоматично надає доступ до сайту за адресою: <http://unimag.test> (за допомогою внутрішнього механізму Host-модифікації).

- Створення бази даних – у phpMyAdmin або через CLI створюється база unimag, котра буде пов'язана з проєктом через файл `.env`.

- Запуск Laravel-сервера – команда `php artisan serve` або просто відкриття unimag.test у браузері запускає сайт у режимі розробки.

У конфігураційному файлі `.env` вказуються параметри підключення до бази даних (див. лістинг 3.1):

Лістинг 3.1 – Підключення до БД

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1
```

```
DB_PORT=3306
DB_DATABASE=unicmag
DB_USERNAME=root
DB_PASSWORD=
```

Також у цьому файлі пізніше налаштовуватимуться змінні для пошти, кешу, сесій тощо.

3.2 Реалізація базової структури сайту

Після налаштування середовища розробки було створено базову структуру веб-сайту інтернет магазину з використанням архітектури MVC. Цей підхід дозволяє ефективно розділяти логіку програми, відображення інтерфейсу та взаємодію з базою даних.

Основні каталоги проєкту:

- App/Http/Controllers – зберігає контролери, які обробляють запити користувачів;
- Resources/views – містить шаблони сторінок у форматі Blade (.blade.php);
- Routes/web.php – визначає маршрути сайту;
- App/Models – моделі для роботи з базою даних через ORM Eloquent;
- Public/ - публічна частина сайту (доступна ззовні), де розміщуються зображення, CSS, JS;
- .env – конфігураційний файл, що зберігає параметри підключення до БД та інші змінні середовища.

У лістингу 3.2 наведено, як у файлі routes/web.php створено базову навігацію:

Лістинг 3.2 – Базова навігація

```
Route::get('/', function () {
    Return view('home');
});
Route::get('/about', function () {
    Return view('about');
```

```
});
```

У відповідь цим маршрутам створено шаблони `resources/views/home.blade.php` та `about.blade.php` за базовою розміткою HTML та використанням компонентів Blade-шаблонізатора.

Для уніфікації стилю сайту створено головний шаблон `layouts/app.blade.php`, що містить загальну структуру сторінки (шапка, меню, футер) та підключення CSS (Bootstrap), що показано у лістингу 3.3:

Лістинг 3.3 – Загальна структура сторінки

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>@yield('title', 'UnicMag')</title>
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0/dist/css/bootst
rap.min.css" rel="stylesheet">
</head>
<body>
    @include('partials.nav')
    <div class="container mt-4">
        @yield('content')
    </div>
</body>
</html>
```

Цей шаблон використовується всіма сторінками сайту через конструкцію `@extends('layouts.app')`.

У файлі `resources/views/partials/nav.blade.php` розміщено меню з посиланням на основні сторінки (див. лістинг 3.4):

Лістинг 3.4 – Меню сайту

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
        <a class="navbar-brand" href="/">UnicMag</a>
        <div class="collapse navbar-collapse">
            <ul class="navbar-nav me-auto">
                <li class="nav-item"><a class="nav-link"
href="/">Головна</a></li>
```

```

        <li class="nav-item"><a href="/about">Про нас</a></li>
    </ul>
</div>
</div>
</nav>

```

3.2.1 Головна сторінка

Головна сторінка відіграє роль першого візуального контакту користувача з веб-сайтом, тому її оформлення повинне бути привабливим, зрозумілим і зручним для навігації. Вона реалізована у вигляді Blade-шаблону `home.blade.php`, що містить основні елементи: логотип магазину, інтерактивне меню категорій товарів, секцію з описом компанії та блок із контактною інформацією.

Графічні елементи головної сторінки (категорії товарів: «Домашнє взуття», «Косметички», «Органайзери») реалізовані у вигляді зображень із текстовими підписами, що з'являються при наведенні (див. рисунок 3.1). Таке рішення дозволяє інтуїтивно зрозуміти тематику розділів і забезпечує швидкий перехід до каталогу обраної категорії. Всі стилі підключаються через функцію `{{ asset('css/styles.css') }}` і використовують сучасну типографіку з Google Fonts.

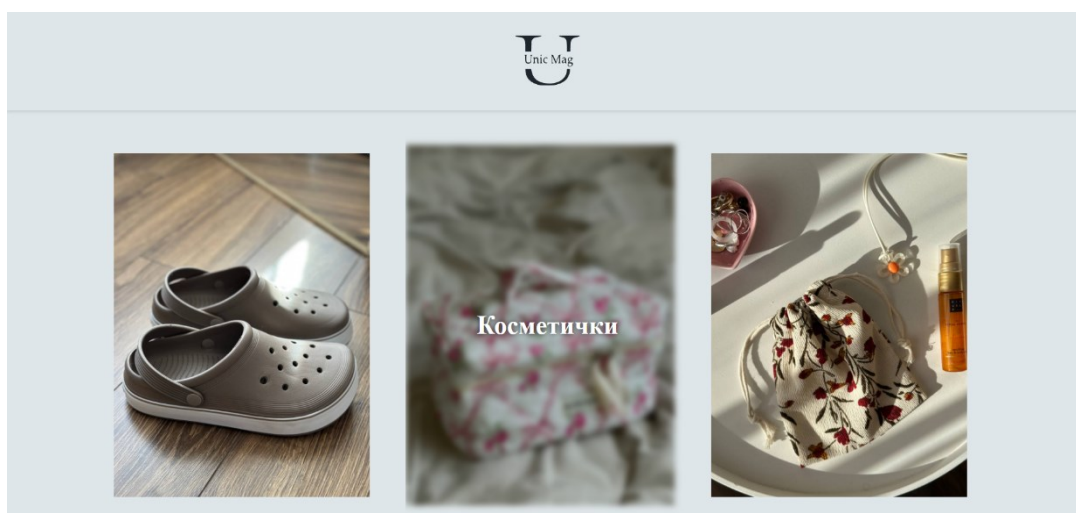


Рисунок 3.1 – Головна сторінка сайту

Адаптивність сторінки забезпечується за допомогою CSS Flexbox і медіа-запитів, що дозволяє відображати сторінку коректно на різних пристроях — як на моніторах, так і на мобільних телефонах [28].

3.2.2 Сторінка каталогу товарів

Сторінка каталогу (`catalog.blade.php`) відповідає за відображення товарів відповідно до обраної категорії. Реалізація цієї сторінки включає:

- динамічне отримання товарів з бази даних за допомогою контролера `'CatalogController'`;
- передачу змінної `'$categoryName'` для відображення назви категорії у заголовку сторінки;
- відображення кожного товару у вигляді окремої картки з фото, назвою, ціною та статусом наявності.

Для виводу карток товарів застосовано CSS-сітку (`'display: grid'`), що дозволяє рівномірно розмістити елементи на сторінці та забезпечити гнучке масштабування (див. рисунок 3.2). Кожна картка має закруглені кути, тіньове оформлення та анімацію при наведенні. Це підвищує візуальну привабливість та інтуїтивну взаємодію користувача з інтерфейсом.

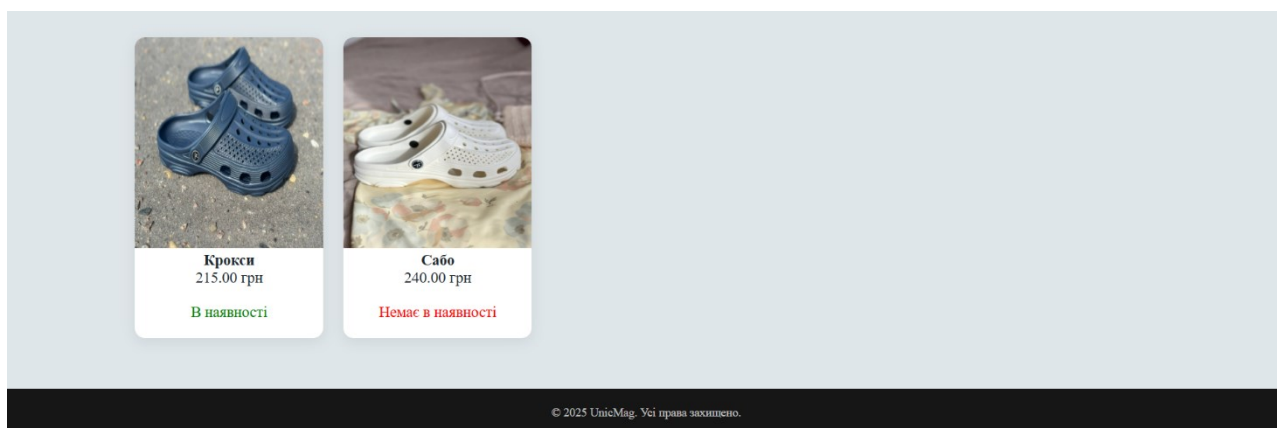


Рисунок 3.2 – Картки товарів у категорії Взуття

При відсутності товарів у категорії передбачено виведення повідомлення «Товари не знайдено у цій категорії» (див. рисунок 3.3), що реалізовано через конструкцію `@forelse ... @empty ... @endforelse` у шаблоні.



Рисунок 3.3 – Відображення відсутності товарів

3.3 Реалізація функціоналу користувача

У сучасному інтернет-магазині одним із ключових елементів є особистий кабінет користувача, що дозволяє зручно керувати замовленнями, переглядати історію покупок та змінювати персональні дані. У межах проєкту реалізовано базовий функціонал реєстрації, авторизації та управління профілем користувача за допомогою вбудованого пакета Laravel Breeze.

Laravel Breeze є мінімалстичним стартовий пакетом, що надає просту систему аутентифікації із використанням Blade-шаблонів. Для його встановлення виконано такі команди (див. лістинг 3.5):

Лістинг 3.5 – Команди для встановлення Laravel Breeze

```
composer require laravel/breeze --dev
php artisan breeze:install
npm install && npm run dev
php artisan migrate
```

У результаті система отримала набір маршрутів та шаблонів для:

- Реєстрації (/register);
- Входу (/login);
- Особистого кабінету (/dashboard);
- Виходу з облікового запису (/logout).

Laravel автоматично створює модель User, яка відповідає за зберігання інформації про користувачів у БД.

Доступ до персонального кабінету обмежено через middleware auth, що гарантує безпеку та авторизований доступ. При спробі відкрити сторінку без входу система перенаправляє на /login.

Інтерфейс користувача побудовано на основі Blade-шаблонів із підтримкою Tailwind CSS (вбудовано в Breeze). Основні шаблони мають адаптивний дизайн і дозволяють розширити функціонал без шкоди для стилю.

3.4 Реалізація адміністративної панелі

Адміністративна панель є важливою складовою інтернет-магазину, оскільки забезпечує управління товарами, категоріями, замовленнями та користувачами. Для цього створюється окремий інтерфейс для адміністратора з обмеженим доступом для інших ролей.

У таблиці users додано поле role, яке визначає рівень доступу:

- User – звичайний покупець;
- Admin – адміністратор.

Це поле додається через команду:

```
php artisan make:migration add_role_to_users_table --table=users
```

У тілі міграції (див. лістинг 3.6):

Лістинг 3.6 – Додавання поля role

```
public function up()
{
    Schema::table('users', function (Blueprint $table) {
        $table->string('role')->default('user');
    });
}
```

```
}
```

У маршрутах створено окрему групу, доступну лише для адміністраторів (див. лістинг 3.7):

Лістинг 3.7 – Група адміністраторів

```
Route::middleware(['auth', 'admin'])->prefix('admin')->group(function () {
    Route::get('/dashboard', [AdminController::class, 'index'])->name('admin.dashboard');
});
```

Middleware admin перевіряє роль користувача (див. лістинг 3.8):

Лістинг 3.8 – Перевірка ролі користувача

```
public function handle($request, Closure $next)
{
    if (auth()->user()->role !== 'admin') {
        abort(403, 'Доступ заборонено');
    }

    return $next($request);
}
```

Контролер адміністратора:

```
php artisan make:controller AdminController
```

У методі index (див. лістинг 3.9):

Лістинг 3.9 – Метод index

```
public function index()
{
    return view('admin.dashboard');
}
```

У Blade-шаблоні `resources/views/admin/dashboard.blade.php` відображаються доступні адміністративні дії.

У рамках реалізації адміністратор може:

- Переглядати список зареєстрованих користувачів;
- Додавати, редагувати й видаляти товари;
- Керувати категоріями;
- Переглядати список замовлень та їх статуси.

Панель адміністратора

Додати новий товар

Назва товару:

Ціна (грн):

Наявність:

Категорія:

Фото товару:

Додати товар

Рисунок 3.4 – Сторінка додавання товару через адмін-панель

3.5 Робота з базою даних через Eloquent ORM

Laravel використовує об'єктно-реляційну модель Eloquent ORM, яка забезпечує зручну роботу з базою даних шляхом взаємодії з таблицями через PHP-класи (моделі). Це дозволяє розробнику оперувати даними не за допомогою SQL-запитів, а через методи об'єктів, що спрощує розробку, покращує читаність коду та підвищує безпеку застосунку.

Для кожної сутності бази даних створено відповідну модель із міграцією. Наприклад, для товарів:

```
php artisan make:model Product -m
```

У результаті створюються:

- `app/Models/Product.php` — модель;
- `database/migrations/xxxx_xx_xx_create_products_table.php` — міграція.

У файлі міграції описано структуру таблиці `products`, з якою можна ознайомитись у лістингу 3.10, а також продемонстрована таблиця на рисунку 3.5.

Лістинг 3.10 – Файл міграції таблиці `products`

```
Schema::create('products', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->decimal('price');
    $table->boolean('availability');
    $table->string('category');
    $table->string('image')->nullable(); // шлях до
    $table->timestamps();
});
```

unicmag.products: 3 всього >> Далі

Показати всі | Сортування | Поля (8/8) | Фільтр

#	id	name	price	availability	category	image
1	1	Крокси	215,0	1	shoes	products/rcYampJuj3TFIkzTgoSryZpIlhB4NU9ef...
2	2	Косметичка	200,0	1	cosmetic	products/zpqPzB8eL7u0w0WgzCci0KrhphGi63Ir...
3	4	Сабо	240,0	0	shoes	products/4hCKkcJUTIZR4Jjc7XBTq8vP3GhIqW0...

Рисунок 3.5 – Таблиця `products`

У Laravel реалізуються зв'язки між моделями, які відповідають зв'язкам між таблицями:

Один до багатьох:

- Категорія має багато товарів

```
public function products() {
    return $this->hasMany(Product::class);
}
```

- Товар належить до однієї категорії

```
public function category() {
    return $this->belongsTo(Category::class);
}
```

Багато до багатьох:

- Товари та замовлення через проміжну таблицю `order_product`.

Для заповнення таблиць тестовими даними застосовуються сидери:

```
php artisan make:seeder ProductSeeder
```

У методі `run()` створюються записи (див. лістинг 3.11):

Лістинг 3.11 – Метод run()

```
Product::create([
    'name' => 'Домашні капці',
    'description' => 'Зручні капці з хутром',
    'price' => 399.99,
    'stock' => 20,
]);
```

Сидери запускаються командою:

```
php artisan db:seed
```

Приклад отримання списку товарів у контролері (див. лістинг 3.12):

Лістинг 3.12 – Отримання списку товарів

```
$products = Product::where('stock', '>', 0)->orderBy('name')-
>get();
return view('shop.index', compact('products'));
```

Це значно зручніше, ніж формування SQL-запитів вручну, та дозволяє легко комбінувати фільтри, сортування, пагінацію.

3.6 Захист і валідація даних

Захист веб-застосунку є критичним аспектом будь-якого інтернет-магазину [23]. У Laravel реалізовано набір вбудованих механізмів безпеки, що дозволяють уникнути найпоширеніших вразливостей: XSS, CSRF, SQL-ін'єкцій, а також забезпечити коректну валідацію вхідних даних.

Laravel надає зручні способи перевірки даних, які надходять із форм. Наприклад, при додаванні товару в контролері (див. лістинг 3.13):

Лістинг 3.13 – Валідація додавання товару

```
$request->validate([
    'name' => 'required|string|max:255',
    'price' => 'required|numeric',
    'availability' => 'required|boolean',
    'category' => 'required|string',
    'image' => 'required|image|max:2048',
]);
```

Якщо дані не відповідають правилам, користувача буде автоматично повернено назад із повідомленням про помилки.

Додати новий товар

Назва товару:

Ціна (грн):

Заповніть це поле.

Рисунок 3.6 – Попередження про обов’язкове заповнення поля

Laravel автоматично додає CSRF-токен [17] до кожної форми через Blade-директиву `@csrf`, що запобігає підробленим запитам (див. лістинг 3.14):

Лістинг 3.14 – CSRF-токен у формі

```
<form method="POST" action="/products">
    @csrf
    <!-- Поля форми -->
</form>
```

Використання Eloquent ORM гарантує безпечну обробку запитів до БД [21]. Наприклад, цей запит:

```
Product::where('name', $request->input('name'))->first();
```

автоматично «екранує» вхідні дані, що унеможливорює SQL-ін’єкцію.

Також Laravel автоматично екранує всі змінні, які виводяться у шаблонах через `{{ }}`. Наприклад:

```
<p>{{ $product->description }}</p>
```

У разі наявності шкідливого JavaScript-коду він буде перетворений у безпечну текстову форму.

Для розмежування прав доступу використовується middleware. Наприклад, сторінки адміністратора доступні лише тим, у кого роль `admin`. Це реалізовано

через клас AdminMiddleware, який перевіряє роль користувача перед наданням доступу до функціоналу.

3.7 Інтеграція з поштою та повідомленнями

Дуже зручною є функція автоматичного інформування користувачів про події, пов'язані з їх діями на сайті – підтвердження реєстрації, створення або зміна статусу замовлення тощо [33]. Для цього налаштовано інтеграцію з електронною поштою через SMTP-сервер хостинг-провайдера та реалізовано базові поштові сповіщення [34].

Проект розміщено на платформі, яка надає SMTP-доступ для надсилення листів із корпоративного домену unicmag.shop. Налаштування мають такий вигляд (див. лістинг 3.15):

Лістинг 3.15 – Налаштування надсилення листів

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.unicmag.shop
MAIL_PORT=465
MAIL_USERNAME=info@unicmag.shop
MAIL_PASSWORD=*****
MAIL_ENCRYPTION=ssl
MAIL_FROM_ADDRESS=info@unicmag.shop
MAIL_FROM_NAME="UnicMag"
```

Завдяки Laravel можна легко реалізувати через вбудовану систему надсилення сповіщень про нові замовлення:

```
php artisan make:notification OrderCreated
```

У методі toMail() формується шаблон листа (див. лістинг 3.16):

Лістинг 3.16 – Шаблон листа про початок обробки замовлення

```
public function toMail($notifiable)
{
    return (new MailMessage)
        ->subject('Ваше замовлення прийнято')
        ->greeting('Дякуємо за покупку в UnicMag!')
```

```

        ->line('Ми отримали ваше замовлення і починаємо його
обробку.')
        ->action('Переглянути замовлення', url('/orders'))
        ->line('З повагою, команда UnicMag.');
```

Повідомлення надсилається одразу після створення замовлення:

```
$user->notify(new OrderCreated($order));
```

Для перевірки надсилання пошти під час розробки можна використовувати:

- Mailtrap.io — для тестування;
- Mailhog (локальний SMTP-сервер);
- або реальну пошту від Ukraine.com.ua — на продакшені.

Переваги реалізованого підходу:

- Централізоване форматування повідомлень;
- Можливість розширити сповіщення на SMS, Telegram тощо;
- Шаблони можуть бути кастомізовані Blade-компонентами.

3.8 Публікація сайту на хостингу

Після завершення розробки сайт було розгорнуто на реальному хостингу для забезпечення постійного доступу користувачів до ресурсу. Для цього використано послуги вже орендованого хостинг-провайдера, на якому зареєстроване доменне ім'я та виділено простір для розміщення файлів сайту.

Перед завантаженням сайту на сервер виконано такі дії:

- Згенеровано production-версію CSS та JS через `npm run build` (для Vite);
- Скопійовано всі файли проєкту, окрім папок `node-modules`, `vendor`, `.git`;
- У корені проєкту перевірено файл `.env`, оновлено підключення до віддаленої БД;
- Виконано команду `composer install` без dev-залежностей (опція `--no-dev`);
- Запущено міграції: `php artisan migrate`.

У панелі керування Ukraine.com.ua створено:

- Базу даних MySQL (дані внесено у .env);
- Домен unicmag.shop, який вказує на підкаталог /public;
- Підключено SSL-сертифікат для захищеного з'єднання HTTPS;
- Налаштовано поштову скриньку info@unicmag.shop для обробки повідомлень.

Оскільки Laravel зберігає публічні файли в папці public, коренева папка домену має бути спрямована саме на неї. Для цього в панелі керування хостингом змінено «кореневу директорію» сайту.

Також були надані права на запис для папок storage та bootstrap/cache, які використовуються для логів та кешу.

Після завантаження файлів і запуску міграцій сайт було успішно перевірено за посиланням:

<https://unicmag.shop>

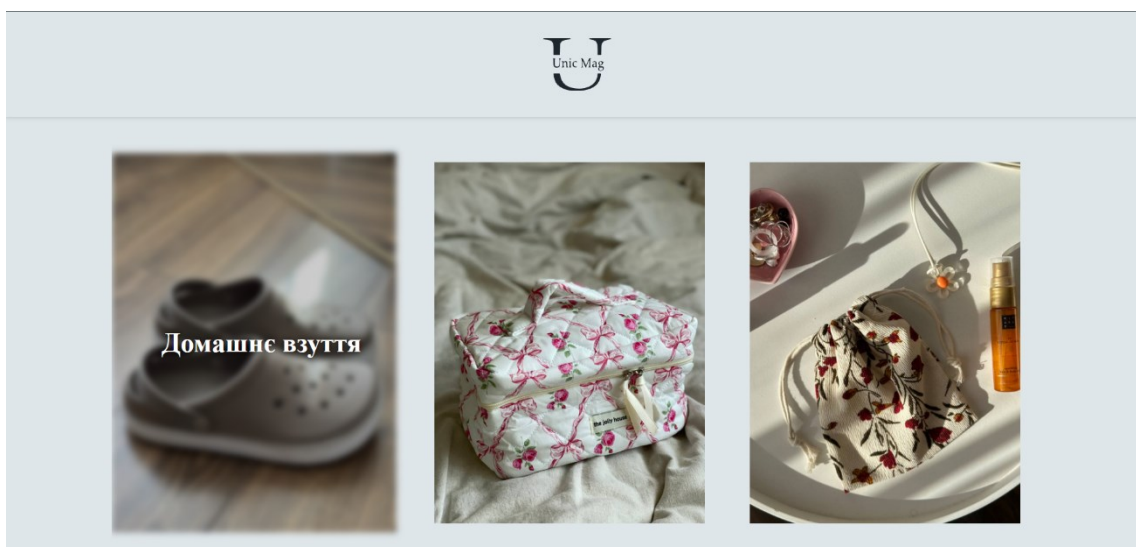


Рисунок 3.7 – Веб-сайт інтернет-магазину «Unic_Mag»

Він коректно відображається на всіх типах пристроїв, працює система авторизації, перегляду товарів, оформлення замовлень і надсилення поштових повідомлень.

3.9 Висновок до третього розділу

У третьому розділі реалізовано інтернет-магазин на базі PHP та Laravel: налаштовано середовище, створено структуру сайту, реалізовано функціонал для користувачів і адміністратора, забезпечено захист і валідацію даних, налаштовано електронну пошту та виконано публікацію проєкту на хостингу під доменом unicmag.shop.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Ергономічні проблеми безпеки життєдіяльності

Ергономіка (грец. «закон праці») — наукова дисципліна, що вивчає взаємодію людини, машини та середовища з метою забезпечити ефективність, безпеку та комфорт роботи [37, 40]. Головна мета — підвищити ефективність функціонування системи «людина-машина-середовище» та забезпечити безпечні умови праці [25].

До основних завдань належить:

- Розробка принципів та алгоритмів для діяльності оператора з урахуванням якості роботи техніки й зовнішніх факторів;
- Дослідження закономірностей взаємодії людини з технічними засобами та середовищем;
- Проектування перспективних форм праці всередині систем «людина-машина-середовище»;
- Розробка методик для створення, аналізу та експлуатації таких систем з акцентом на безпеку людини та продуктивність.

Ергономіка вивчає трудові процеси в умовах впливу на людину різноманітних факторів середовища, а також є частиною інженерної психології – науки, що аналізує інформаційні процеси оператора в технічній системі.

Робоче місце розглядається як зона, де людина взаємодіє з апаратурою, предметами праці та допоміжними засобами, оснащена екранами, органами керування та інструментами.

Організація такого простору спрямована на:

- оптимізацію умов праці,
- гарантію безпеки працівника,
- досягнення максимальної ефективності,
- забезпечення комфортності.

Основні вимоги включають:

- Забезпечення достатнього вільного простору для рухів;
- Забезпечення фізичного, зорового та слухового контакту з обладнанням і між працівниками;
- Належне освітлення;
- Контроль рівня шуму, вібрацій та інших шкідливих впливів;
- Наявність засобів індивідуального захисту;
- Рациональне розташування робочих зон та безпечні проходи.

4.2 Економічне значення заходів щодо покращення умов та охорони праці

Охорона праці має не лише соціальну, а й вагомую економічну цінність. Своєчасне виявлення небезпечних умов, профілактика нещасних випадків і захист працівників від шкідливих чинників дозволяють зменшити втрати для підприємства [38]. Економічний аналіз у сфері безпеки праці спрямований на виявлення фінансових збитків, пов'язаних із травматизмом і професійними захворюваннями, та на оцінку ефективності впроваджених профілактичних заходів.

Основними завданнями економічного аналізу охорони праці є:

- визначення розмірів економічних збитків від виробничих травм;
- обчислення втрат від плинності кадрів і зниження продуктивності;
- оцінка ефективності вкладених коштів у заходи з охорони праці;
- розрахунок окупності профілактичних інвестицій.

У практиці управління безпекою праці широко використовуються ключові показники, зокрема:

- Показник частоти травматизму (Кч) – кількість нещасних випадків на 1000 працівників;
- Показник тяжкості травматизму (Кт) – середня тривалість непрацездатності в днях на одного потерпілого;

– Рівень витрат на охорону праці – загальна сума витрачених коштів на заходи безпеки протягом звітного періоду.

Ці показники дозволяють виявити динаміку змін, порівняти ситуацію з попередніми періодами чи з іншими організаціями, а також вчасно приймати управлінські рішення.

Для покращення безпеки умов праці підприємство може реалізовувати наступні економічно доцільні заходи:

- придбання засобів індивідуального захисту (окуляри, антивідблискові фільтри, стільці з ортопедичною підтримкою тощо);
- модернізація технічного обладнання (наприклад, більш ергономічні монітори, клавіатури);
- інвестування в навчання працівників з техніки безпеки;
- створення комфортних побутових умов (кімнати відпочинку, зона харчування).

Важливо також враховувати економічний ефект від впроваджених заходів, який може проявлятися у:

- скороченні витрат на лікування та компенсації;
- зниженні кількості лікарняних і прогулів;
- підвищенні продуктивності праці;
- зменшенні кількості помилок і аварійних ситуацій.

На завершення слід зазначити, що грамотне фінансування заходів з охорони праці є інвестицією, що швидко окупається. Якщо коефіцієнт ефективності перевищує нормативний рівень, підприємство отримує не лише безпечні умови для працівників, а й відчутний фінансовий результат.

4.3 Планування робіт щодо охорони праці

На сучасних підприємствах безпека праці забезпечується через перспективне та поточне планування заходів, спрямованих на покращення виробничого середовища, гігієни та безпеки праці [39].

При формуванні планів із охорони праці враховуються такі документи та дані :

- Стратегічний і соціально-економічний розвиток підприємства.
- План організаційно-технічних і технологічних нововведень.
- Результати паспортизації цехів і робочих місць, та атестації умов праці.
- Висновки комплексних перевірок щодо нормативів безпеки, гігієни та санітарії.
- Матеріали розслідувань нещасних випадків, аварій, профзахворювань.
- Приписи контролюючих органів і служби охорони праці.
- Рекомендації комісії з охорони праці та пропозиції трудового колективу.
- Відповідні рішення профспілки, адміністрації, державних і галузевих органів.

Особливе значення надається розділу «Охорона праці» в колективному договорі між роботодавцем і трудовим колективом (за участі профспілки) [41]. У ньому визначаються інженерно-технічні заходи щодо досягнення нормативів безпеки, включно з:

- Проектуванням та встановленням сучасних інженерних засобів захисту (огорожень, сигналізації, запобіжних пристроїв);
- Виготовленням і монтажем ефективнішого обладнання;
- Легалізацією технічної документації для реалізації заходів охорони праці.
- Джерела фінансування
- Фінансування здійснюється роботодавцем через:
 - Виробничі витрати;
 - Амортизаційні відрахування;
 - Капітальні інвестиції;
 - Банківські кредити;
 - Бюджетні кошториси;
 - Спеціальний фонд охорони праці підприємства.

Закон України «Про охорону праці» передбачає створення фондів охорони праці на державному, галузевому, регіональному рівнях, поповнюваних коштами підприємств (0,5 % від реалізації або 0,2 % фондом оплати праці), бюджетів та штрафів.

Першочерговість реалізації заходів визначається на основі атестації робочих місць, аналізу нещасних випадків та оцінки їх соціальної та економічної значущості.

Кошти фонду охорони праці не CAN, but SHOULD NOT:

- використовуватися на ремонтні роботи чи поточне технічне обслуговування основних фондів;
- покривати компенсації чи пільги;
- фінансувати природоохоронні проекти.
- Галузевий і регіональний рівні
- Заходи, які фінансуються за кошти галузевих фондів, включають:
- Впровадження автоматизованих систем попереджувального характеру (аварійних, пожежних, охоронних);
- Програми з метрологічного, кадрового та організаційного забезпечення атестації робочих місць;
- Науково-дослідні та проектно-конструкторські роботи у сфері безпеки праці.

За рахунок регіональних фондів здійснюються:

- Оновлення приміщень для зберігання небезпечних речовин;
- Утилізація небезпечних відходів;
- Забезпечення безпеки об'єктів з масовим перебуванням людей (магазини, навчальні заклади тощо).

4.4 Висновок до четвертого розділу

У розділі було розглянуто основні аспекти охорони праці, актуальні для діяльності веб-розробника, зокрема ергономіку робочого місця, економічну доцільність створення безпечних умов праці та планування профілактичних заходів. Такі дії сприяють збереженню здоров'я працівника, зменшенню втомлюваності та підвищенню ефективності роботи над програмним продуктом.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було створено функціонуючий веб-сайт інтернет-магазину, який реалізовано з використанням мови програмування PHP та сучасних веб-технологій. Інтернет-магазин має повний набір функцій для роботи з каталогом товарів, управління замовленнями та адміністративною панеллю для керування вмістом.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- проаналізовано основні типи сайтів і їх функціональні особливості;
- наведено огляд сучасних технологій веб-розробки;
- обґрунтовано вибір мови PHP для реалізації інтернет-магазину;
- сформульовано вимоги до функціональності та дизайну майбутнього сайту;
- проведено аналіз існуючих аналогів та визначено конкурентні переваги.

В другому розділі кваліфікаційної роботи:

- розроблено загальну архітектуру сайту та структуру бази даних;
- описано логіку взаємодії між клієнтською та серверною частиною;
- реалізовано систему авторизації користувачів та ролей;
- забезпечено захист персональних даних та введено основи безпеки;
- створено зручну навігацію сайту та адаптовано інтерфейс під різні пристрої.

В третьому розділі кваліфікаційної роботи:

- реалізовано функціонал адміністративної панелі для управління товарами;
- додано можливість створення, редагування та видалення товарів за категоріями;
- створено механізм відображення товарів для користувача за категоріями;
- інтегровано простий шаблон оформлення замовлення;

- забезпечено базову валідацію введених даних.

У розділі «Безпека життєдіяльності, основи охорони праці» розглянуто основні ризики для здоров'я під час роботи за комп'ютером, зокрема навантаження на зір і спину. Проаналізовано вимоги до безпечної організації робочого місця та наведено рекомендації щодо режиму праці й відпочинку. Також відзначено значення охорони праці в ІТ-сфері.

ПЕРЕЛІК ДЖЕРЕЛ

1. Laudon, K. C., & Traver, C. G. (2023). *E-commerce 2023: Business, technology, society* (17th ed.). Pearson.
2. Statista. (n.d.). *Worldwide Retail E-commerce Sales*. Retrieved from <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/>
3. Васильєв, О. М. (2022). *Програмування мовою PHP* (навч. посіб., 368 с.). Київ: Видавництво Ліра-К. ISBN 978-617-520-322-4.
4. Терещук В. С. Розробка інтернет-магазину «BROX» з використанням PHP 7.5, MySQL 8.0 та JavaScript v8
5. Powers, D. (2014). *PHP Solutions: Dynamic Web Design Made Easy* (3rd ed.). Apress.
6. W3Techs. (n.d.). *Usage Statistics and Market Share of PHP*. Retrieved from <https://w3techs.com/technologies/details/pl-php>
7. PHP. (n.d.). <https://www.php.net>. Retrieved from <https://www.php.net>
8. Hostinger. (n.d.). *What is PHP & How Does it Work?* Retrieved from <https://www.hostinger.com/tutorials/what-is-php>
9. PHP Benchmarks. (n.d.). PHP Benchmarks. Retrieved from <https://www.phpbenchmarks.com>
10. ts2.tech. (2025). *War-Fueled E-Commerce Boom: Inside Ukraine's Online Shopping Surge in 2025*. Retrieved from <https://ts2.tech/war-fueled-e-commerce-boom-inside-ukraines-online-shopping-surge-in-2025/>
11. Laravel. (n.d.). *Laravel Documentation*. Retrieved from <https://laravel.com/docs>
12. Laragon. (n.d.). *Official website*. Retrieved from <https://laragon.org>
13. Strapi. (2025, February 21). *7 Laravel Best Practices for Developers in 2025*. Retrieved from <https://strapi.io/blog/laravel-best-practices>
14. Cloudways. (2025, April). *Core PHP vs Laravel: Which Should You Choose for Your Next Project?* Retrieved from <https://www.cloudways.com/blog/core-php-vs-laravel-performance/>

15. DM WebSoft LLP. (2024, October). *Building High-Performance E-commerce Stores with PHP Frameworks Like Laravel*. Retrieved from <https://dmwebsoft.com/blog/building-high-performance-e-commerce-stores-with-php-frameworks-like-laravel>
16. Mallow Technologies. (2025, January 3). *14 Laravel security best practices for 2025*. Retrieved from <https://mallow-tech.com/blog/14-best-security-practices-to-follow-for-your-laravel-application/>
17. DEV Community. (2025, March). *Laravel vs Nodejs: Security Comparison for Web Applications*. Retrieved from <https://dev.to/prolion/laravel-vs-nodejs-security-comparison-for-web-applications-219o>
18. DevDiggers. (n.d.). *OWASP's Guide to Mitigating Ecommerce Security Threats*. Retrieved from <https://devdiggers.com/owasp-guide-to-mitigating-ecommerce-security-threats/>
19. Svitla Systems. (2025, April). *E-commerce Security Checklist: How to Ensure Your Website is Safe*. Retrieved from <https://www.svitla.com/blog/e-commerce-security-checklist-how-to-ensure-your-website-is-safe>
20. Pixelfreestudio. (n.d.). *How to Use OWASP Guidelines for Web Security*. Retrieved from <https://pixelfreestudio.com/how-to-use-owasp-guidelines-for-web-security/>
21. GO-Globe. (2025, March 26). *Mastering Database Performance Optimization: Proven Strategies for Peak Efficiency*. Retrieved from <https://www.go-globe.com/mastering-database-performance-optimization>
22. Snowdog. (2025, March 18). *Optimizing Database Performance for High-Traffic eCommerce Stores*. Retrieved from <https://www.snow.dog/blog/optimizing-database-performance-for-high-traffic-ecommerce-stores>
23. Duda, O., Martsenko, S., Matsiuk, O., Kunanets, N., & Pasichnyk, V. (2020). Building secure urban information systems based on IoT technologies.
24. Kharchenko, A., Bodnarchuk, I., & Yatcyshyn, V. (2014). The method for comparative evaluation of software architecture with accounting of trade-offs. *American Journal of Information Systems*, 2(1), 20–25.

25. Kunanets, N., Pasichnyk, V., Bodnarchuk, I., Martsenko, S., & Matsiuk, O. (2019). Information system for visual analyzer disease diagnostics.
26. COI.UA. (2025, January 15). *UX/UI design of the online store 2025: trendy solutions from COI.UA*. Retrieved from <https://coi.ua/en/blog/DesignCo/ux-ui-design-for-e-commerce-in-2025-how-to-create-a-user-friendly-and-attractive-website/>
27. Netguru. (2025, January 22). *Essential Principles for Ecommerce Web UI Design*. Retrieved from <https://www.netguru.com/blog/principles-ecommerce-web-ui-design>
28. O8 Agency. (2025). *Agile Web Design Process: The Complete Guide (2025)*. Retrieved from <https://www.o8.agency/blog/agile-web-design-process>
29. Verpex. (2025, May 22). *Agile E-commerce*. Retrieved from <https://verpex.com/blog/ecommerce-woocommerce-hosting/agile-e-commerce>
30. DigitalOcean. (n.d.). *Laravel hosting*. Retrieved from <https://www.digitalocean.com/solutions/laravel-hosting>
31. Bitcatcha. (2025, June 11). *11 Best Laravel Hosting Providers 2025*. Retrieved from <https://www.bitcatcha.com/web-hosting/php/laravel/>
32. CloudPanel. (2025). *5 Steps to Deploy Laravel Applications in CloudPanel in 2025*. Retrieved from <https://www.cloudpanel.io/tutorial/deploy-laravel/>
33. Apurple. (2025, May 20). *eCommerce Payment Gateway Integration: A Complete Guide*. Retrieved from <https://www.apurple.co/ecommerce-payment-gateway-integration/>
34. ECORN Agency. (n.d.). *Payment Gateway Integration Guide: E-commerce Growth*. Retrieved from <https://www.ecorn.agency/blog/payment-gateway-integration-guide>
35. Dinarys. (n.d.). *8 Key Advantages of CRM for E-Commerce*. Retrieved from <https://dinarys.com/blog/7-key-advantages-of-crm-for-e-commerce>
36. BugBug.io. (2024, March 21). *Ecommerce Website Testing - Comprehensive Guide*. Retrieved from <https://bugbug.io/blog/software-testing/ecommerce-website-testing/>

37. Studfile. (n.d.). *Ергономічні основи охорони праці*.
<https://studfile.net/preview/10250869/page:9/>
38. Studfile. (n.d.). *Охорона праці в галузі*.
<https://studfile.net/preview/9364384/>
39. Бібліотека BukLib. (n.d.). *Планування заходів з охорони праці*.
<https://buklib.net/books/35174/>
40. dev.to. (2024, September). *The Importance of Ergonomics for Software Engineers: A Comprehensive Guide*. Retrieved from <https://dev.to/bostjank/the-importance-of-ergonomics-for-software-engineers-a-comprehensive-guide-2f1f>
41. OHSE.ca. (2025, June). *Workplace Safety for Software Developers: Protecting Health in Digital Environments*. Retrieved from <https://ohse.ca/blog/workplace-safety-for-software-developers>
42. Боднарчук, І., Харченко, О., Хоміцький, Б., & Шимчук, Г. (2019). Проектування архітектури програмних систем в проектах з гнучкими методами управління. Матеріали XXI Наукової Конференції Тернопільського Національного Технічного Університету Імені Івана Пулюя, 46–48.
43. Струтинська, І. В. (2015). Бренд-менеджмент: теорія і практика. Тернопільський національний технічний університет імені Івана Пулюя.