

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ З ПІДКЛЮЧЕНИМ TELEGRAM-
БОТОМ З ВИКОРИСТАННЯМ LARAVEL ТА MYSQL**

**DEVELOPMENT OF AN ONLINE STORE WITH AN INTEGRATED
TELEGRAM BOT USING LARAVEL AND MYSQL**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-42
Нищий Назар Ігорович

Керівник:
Самчук Людмила Михайлівна

Кваліфікаційну роботу
допущено до захисту

«10» 06 2025 р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

« 20 » 12 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Нищому Назару Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка інтернет-магазину з підключенням Telegram-ботом з використанням Laravel та MySQL»

Керівник роботи: к.т.н., доцент Самчук Л. М.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

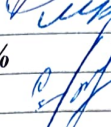
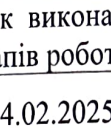
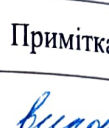
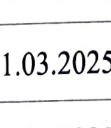
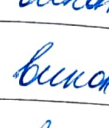
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: PHP 8.3, Laravel 10, MySQL, редактор Visual Studio Code, Tailwind CSS, DaisyUI, Livewire, Telegram Bot API, Google Cloud Platform, методичні вказівки до виконання кваліфікаційної роботи бакалавра

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): Аналіз предметної області і існуючих програмних рішень, визначення вимог до розробки веб-додатка, вибір засобів для реалізації веб-додатка, розробка веб-додатка, тестування з обраними засобами

5. Перелік графічного матеріалу: 23 рисунка, 2 таблиці, 1 додаток

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видано	завдання прийнято
Аналіз предметної області	Самчук Л. М.		
Специфікація вимог до розробленої системи	Самчук Л. М.		
Розробка об'єкта проектування	Самчук Л. М.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		105%	
Академічна доброчесність	Самчук Л. М.		

7. Дата видачі завдання «20» грудня 2024 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	викон.
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 01.03.2025 р.	викон.
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	викон.
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	викон.
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	викон.
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	викон.
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 10.06.2025 р.	викон.

Здобувач вищої освіти



Назар НИЩИЙ

Керівник кваліфікаційної роботи



Людмила САМЧУК

АНОТАЦІЯ

Нищий Н. І. Розробка інтернет-магазину з підключеним Telegram-ботом з використанням Laravel та MySQL. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

У першому розділі здійснено аналіз предметної області і існуючих вже програмних рішень і сформульовано завдання. В другому розділі була розроблена діаграма класів для створення бази даних, були визначені цілі для створення інтернет-магазину і були вибрані технології для реалізації. У третьому розділі описано процес реалізації інтернет-магазину, Telegram-бота і тестування програмного коду. У висновках описано коротко результат написання програмного забезпечення, перспективи і ідеї для подальшого покращення.

Ключові слова: інтернет-магазин, Laravel, MySQL, Telegram-бот, панель адміністратора, бази даних, Tailwind CSS, VS Code, Livewire, міграція, контролер, API, хостинг.

ABSTRACT

Nyshchyi N. Development of an online store with an integrated Telegram bot using Laravel and MySQL. Manuscript. Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's qualification thesis consists of an introduction, three chapters, conclusions, a list of references, and appendices.

The first chapter presents an analysis of the subject domain and existing software solutions and formulates the objectives. The second chapter presents a class diagram for the database, defines the goals for creating the online store, and selects the technologies for its implementation. The third chapter describes the implementation process of the online store, the Telegram bot, and the testing of the program code. The conclusions summarize the results of the software development, as well as prospects and ideas for further improvement.

Keywords: online store, Laravel, MySQL, Telegram bot, admin panel, databases, Tailwind CSS, VS Code, Livewire, migration, controller, API, hosting.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	8
1.1 Аналіз сучасного стану проблеми	8
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	17
Висновки до розділу 1	17
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ... 19	
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення	19
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання ...	27
Висновки до розділу 2.....	35
РОЗДІЛ 3 РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ З ПІДКЛЮЧЕНИМ TELEGRAM-БОТОМ З ВИКОРИСТАННЯМ LARAVEL ТА MYSQL	37
3.1 Практична реалізація об'єкта проєктування.....	37
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	47
3.3 Розробка бази даних	51
3.4 Створення Telegram-бота.....	52
3.5 Захист інформаційно-комп'ютерної системи	55
Висновки до розділу 3	57
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61
ДОДАТКИ	63

ВСТУП

Актуальність теми. Сучасний розвиток електронної комерції в Україні характеризується стрімким зростанням кількості онлайн-покупців, впровадженням інноваційних технологій та інструментів автоматизації бізнес-процесів. Одним із перспективних напрямів є інтеграція месенджерів, зокрема Telegram. Використання фреймворку Laravel у поєднанні із системою управління базами даних MySQL дає можливість розроблювати сучасні інтернет-магазини.

Мета роботи – розробити інтернет-магазин із підключеним Telegram-ботом з використанням Laravel та MySQL, який забезпечить зручну взаємодію користувачів із каталогом товарів, оформленням замовлень.

Об’єкт роботи – технологічні та програмні засоби створення інтернет-магазинів і інтегрування бота месенджера.

Предмет роботи – методи та інструменти проектування, розробки й тестування веб-додатків на основі Laravel і MySQL з використанням Telegram Bot API.

Для досягнення поставленої мети були визначені такі завдання:

- провести аналіз предметної області, проаналізувати існуючі програмні рішення;
- визначити вимоги до розроблюваного веб-додатка;
- вибрати засоби для реалізації веб-додатка;
- розробити веб-додаток завдяки вибраними засобами;
- проаналізувати найпопулярніші засоби тестування проєкту, провести тестування обраними серед них;
- розробити структуру і створити бази даних;
- створити Telegram-бота для інтернет-магазину;
- реалізувати захист веб-додатка.

Апробацію результатів роботи здійснено шляхом участі в X Міжнародній науково-практичній конференції з проблем вищої освіти і науки «Інформаційні технології в освіті, науці і виробництві (ІТОНВ-2025)» (додаток А).

РОЗДІЛ 1

АНАЛІЗ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Інтернет-магазини нині відіграють ключову роль у сучасному світі, адже споживачі мають можливість здійснити покупки онлайн. Завдяки таким покупкам зникає потреба постійно відвідувати різні магазини заради пошуку потрібного товару, так як великий спектр різних товарів може бути доступним в одному місці.

Ця сфера активно розвивається, і тенденції засвідчують стабільне зростання як кількості користувачів так і доходу в цьому напрямку відповідно. Суттєво впливає на розвиток електронної комерції наявність інтернету, збільшення кількості мобільних пристроїв, а також зміна поведінкових звичок споживачів, які дедалі частіше обирають онлайн-платформи для здійснення покупок. Впровадження технологій Big Data дозволяє краще розуміти потреби клієнтів, передбачати тренди й приймати бізнес-рішення. Крім того, важливу роль відіграє цифровий маркетинг, який забезпечує таргетовану рекламу та просування товарів, тим самим підвищуючи об'єми продажів. Згідно з дослідженням, проведеним в статті [1], наведена діаграма доходу ринку (рис. 1.1) електронної комерції в Україні.

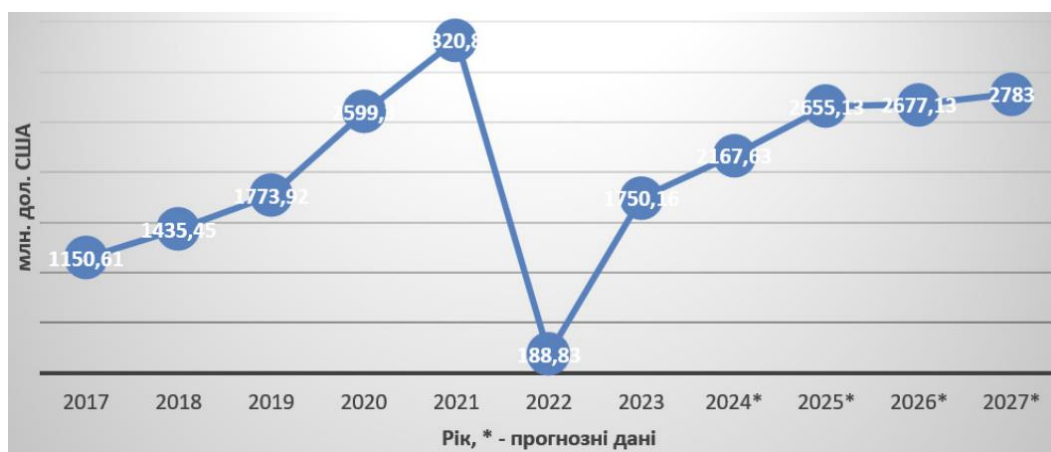


Рисунок 1.1 – Діаграма доходу ринку електронної комерції в Україні [1]

На цій діаграмі можна побачити стрімкий і стабільний ріст прибутків, наприклад в 2019 році дохід складав 1773,92 мільйона доларів США, тоді як в 2021 році вже рекордні 3320,80 мільйонів доларів США. Хоча через важку ситуацію в Україні, після сильного падіння доходів в 2022 році, 2023 рік показав позитивно, що символізує відновлення. Далі прогнозується стабільне зростання в електронній комерції у наступні роки. Дослідження, наведене у статті [1], підкреслює, що збільшення доходів в електронній комерції в Україні зумовлене не лише збільшенням кількості інтернет-користувачів, але й активним впровадженням інноваційних та передових технологій, таких як штучний інтелект, машинне навчання та аналітика великих даних. Приміром, використання алгоритмів рекомендацій, що базуються на аналізі поведінки користувачів, дозволяє інтернет-магазинам пропонувати товари, які відповідають індивідуальним потребам клієнтів. Це значно підвищує вірогідність покупки та сприяє збільшенню середнього чека. Окрім цього, важливим фактором є розвиток логістичної інфраструктури, зокрема співпраця з кур'єрськими службами та поштовими операторами, що забезпечують швидку доставку навіть у віддалені куточки країни.

Ще одним аспектом, що впливає на розширення ринку, є активне включення соціальних мереж у процеси електронної комерції. Соціальні платформи, такі як Instagram, TikTok і Facebook, стали не лише каналами просування, але й повноцінними торговими майданчиками, де користувачі мають можливість здійснювати покупки безпосередньо через інтерфейс додатків. В Україні, де значна частина населення активно використовує соціальні мережі, цей тренд набуває особливого значення. Наприклад, чимало малих і середніх бізнесів використовують Instagram Shops для продажу товарів, що дає змогу їм уникати витрат на створення власних веб-сайтів. Водночас великі гравці ринку, на кшталт Rozetka чи Prom.ua, інтегрують функції соціальної комерції у свої платформи.

Зважаючи на позитивні зрушення, електронна комерція в Україні зустрічається з деякими перешкодами, що можуть стримувати її розвиток.

Однією з них є невисокий рівень цифрової обізнаності у деяких прошарках населення, зокрема старшого покоління, що обмежує їхню участь в онлайн-покупках. Для вирішення цієї проблеми компанії активно вкладають кошти у розробку інтуїтивно зрозумілих інтерфейсів та навчальних матеріалів, які допомагають користувачам освоювати онлайн-покупки. Крім того, на часі залишається питання кібербезпеки, адже зростання обсягів онлайн-транзакцій супроводжується збільшенням числа кібератак. Застосування сучасних стандартів шифрування, таких як SSL/TLS, та використання двофакторної аутентифікації стають критичними для забезпечення довіри клієнтів до платформ електронної комерції.

Сучасні покупці очікують високу якість обслуговування, миттєвого доступу до даних про продукти, зручного інтерфейсу, а також різноманітних способів комунікації із продавцем. Безперечно, ці вимоги зумовлюють ряд викликів для розробників програмного забезпечення.

На сьогодні однією з ключових проблем у сфері інтернет-торгівлі є конкуренція за увагу покупців, через що компанії змушені постійно вдосконалювати свої розробки, впроваджувати новий функціонал для утримання клієнтів. В умовах перенасичення ринку онлайн-магазинами, користувачі стають більш вимогливими, тому навіть незначні дефекти у роботі сайту можуть спричинити втрату клієнта. Конкуренція відчувається не лише в ціні чи різноманітності товарів, але й у зручності навігації, швидкості завантаження сторінок та в загальному досвіді взаємодії з платформою. Саме тому компанії впроваджують такі методи, як персоналізація контенту, а також використання UX/UI-досліджень для вдосконалення функціоналу. Інтернет-магазини повинні гарантувати зручність використання на різних пристроях, а також підтримку кількох мов, що є надзвичайно важливим для регіональних і глобальних ринків.

Багатомовність, у свою чергу, відкриває нові можливості для розширення аудиторії. В Україні, де значна частина населення є двомовною, підтримка кількох мов на сайті стає не просто питанням зручності, але й конкурентною перевагою. Скажімо, додавання англійської мови може зацікавити іноземних

клієнтів, особливо на тлі зростання експорту української продукції. Для реалізації багатомовності розробники вдаються до спеціалізованих плагінів, таких як WPML для WordPress або вбудованих інструментів локалізації в Laravel, які дозволяють легко перемикатися між мовами без втрати функціональності. До того ж, важливо враховувати культурні особливості під час перекладу контенту, щоб уникнути непорозумінь чи некоректного сприйняття інформації на різних ринків.

Ключовим аспектом сучасного інтернет-магазину це є автоматизація процесів, наприклад, опрацювання замовлень, керування складськими запасами, відстеження доставок та підтримка клієнтів. Автоматизація охоплює не лише технічну складову, але й бізнес-процеси, що стосуються управління клієнтською базою, розподілу споживачів, розсилками електронною поштою та сповіщень. До того ж, слід впроваджувати системи CRM (Customer Relationship Management), які дозволяють зберігати історію покупок клієнтів, проводити аналіз продажів та формувати індивідуальну взаємодію з кожним покупцем. Цей підхід сприяє зростанню прихильності до бренду та стимулює повторні покупки. Платіжні системи є важливою інтеграцією серед зовнішніх сервісів, також до них можна додати логістичні та аналітичні сервіси. Останнім часом дедалі більше поширюється інтеграція проєктів з месенджерами, включаючи Telegram, з метою поліпшення комунікації з клієнтами та автоматизації робочих процесів.

Логістичні сервіси також відіграють ключову роль у гарантуванні позитивного досвіду клієнтів. В Україні взаємодія з перевізниками, такими як Нова Пошта, Укрпошта чи Meest, дозволяє онлайн-магазинам пропонувати клієнтам гнучкі варіанти доставки, включаючи кур'єрську доставку, самовивіз із відділень або поштоматів. Інтеграція логістичних API дає змогу автоматично відстежувати статус доставки, надсилати клієнтам сповіщення про зміну статусу замовлення та навіть розраховувати вартість доставки в реальному часі залежно від регіону. Скажімо, Нова Пошта надає API для інтеграції, який дозволяє магазинам автоматично формувати накладні, відстежувати посилки та надавати клієнтам актуальну інформацію про доставку.

Telegram-боти здатні запропонувати клієнтам зручний спосіб взаємодії клієнта з магазином. Вони дають змогу шукати товари, замовляти, відслідковувати доставку та ставити запитання. Це особливо важливо в умовах зростання популярності мобільних гаджетів та месенджерів як основний спосіб комунікації. Окрім Telegram, популярними каналами комунікації є також Viber, WhatsApp, Facebook Messenger, кожна з яких має власну аудиторію та набір технічних особливостей. Вибір конкретного месенджера для інтеграції залежить від цільової аудиторії інтернет-магазину. Враховуючи високий рівень відкритості повідомлень у месенджерах, ці платформи можуть бути ефективним інструментом для маркетингу, опитувань, зворотного зв'язку та навіть прийому оплат, якщо дозволяє API обраної платформи.

Слід пам'ятати, що водночас існує певна кількість проблем, які потенційно потребують уваги під час реалізації таких рішень. Серед них можна виділити: гарантування захисту даних, особливо під час обробки особистих відомостей клієнтів, а також під час здійснення онлайн-платежів. Ще одним важливим завданням є підтримка масштабованості системи, щоб вона могла витримувати високі навантаження в період пікового споживання, як-от під час святкових розпродажів. Тут слід враховувати архітектурні рішення, такі як мікросервіси, кешування, використання CDN (Content Delivery Network), а також горизонтальне масштабування серверів. Наприклад, зростання кількості одночасних користувачів під час акцій або розпродажів може спричинити перевантаження бази даних чи падіння продуктивності веб-сайту. Використання черг завдань (queue systems), на кшталт Laravel Queue, дає змогу ефективно керувати асинхронними процесами й уникнути затримок у роботі сайту.

Авторами статті [2] проаналізовано різні підходи науковців щодо визначення сутності електронної торгівлі, а також відмінності між категоріями «електронна комерція» та «е-торгівля». Встановлено передумови та прослідковано еволюцію е-торгівлі в глобальній економіці та розвитку електронного бізнесу. Визначено основні організаційно-економічні моделі електронної комерції та уточнено місце е-торгівлі в системі електронної

комерції. Також автори статті охарактеризували етапи розробки стандартів для організації електронного обміну даними між підприємствами (EDI). Можна сказати, що було досліджено специфіку електронної комерції як джерела конкурентних переваг для виробників, продавців та ринкових партнерів. Згідно з матеріалами статті, можна дійти до того що є відмінності між е-торгівлею та традиційною торгівлею. Визначено обмеження та бар'єри, пов'язані з функціонуванням електронної комерції та електронної торгівлі. В результаті дослідження в статті був зроблений висновок, що новим етапом розвитку електронної комерції є прогрес, заснований на принципах сучасного цифрового маркетингу, виведення товарів і послуг на ринок через онлайн-продажі, досягаючи тим самим лідерства, сталий розвиток і висока конкурентоспроможність у глобальному конкурентному середовищі. Визначено основні тенденції розвитку електронної комерції у світі та в Україні та внесено пропозиції щодо вдосконалення системи онлайн-продажів. Це виявилось величезною перевагою для покупців і продавців, які здійснюють транзакції через віртуальні торгові платформи. Ці платформи знижують вхідний поріг для малого бізнесу, дозволяючи навіть без фізичної точки продажу вести ефективну торговельну діяльність. Завдяки платформам типу Shopify, Prom.ua або Rozetka Marketplace багато підприємців отримують доступ до великої клієнтської бази. Разом з тим, власна розробка інтернет-магазину дозволяє повністю контролювати дизайн, функціонал і взаємодію з клієнтами, що є важливою перевагою в порівнянні з готовими рішеннями.

В роботі [3] перераховуються переваги електронної комерції. Останнім часом споживачі все більше віддають перевагу онлайн-покупкам завдяки їх зручності та доступності. Це сприятиме збільшенню продажів в онлайн-секторі, включаючи продукти харчування та напої. Удосконалення методів аналізу даних і інтелектуальних систем рекомендацій дозволить платформам електронної комерції надавати ще більш персоналізовані пропозиції споживачам у цих ринкових нішах, що також позитивно впливатиме на обсяги продажів.

Важливо усвідомлювати потенціал конкурентоспроможності України в цифровій комерції, і яка роль міжнародних а також європейських організацій. У статті [4] акцентовано, що завданням України повинна бути адаптація законодавчої бази до визнаних світових стандартів, зокрема законодавства Європейського Союзу, в якості мети можна назвати гармонізації українського ринку в електронній комерції. Рекомендації міжнародних організацій були б корисними для бізнесу.

Для успішного створення ефективного інтернет-магазину необхідним є вивчити існуючі рішення, які вже реалізовані на ринку. Одним із прикладів такого ресурсу є веб-сайт Style Print [5], який пропонує широкий асортимент функцій для онлайн-торгівлі а також одягу як для чоловіків так і для жінок. Зовнішній вигляд сторінки каталогу зображено на рисунку 1.2.

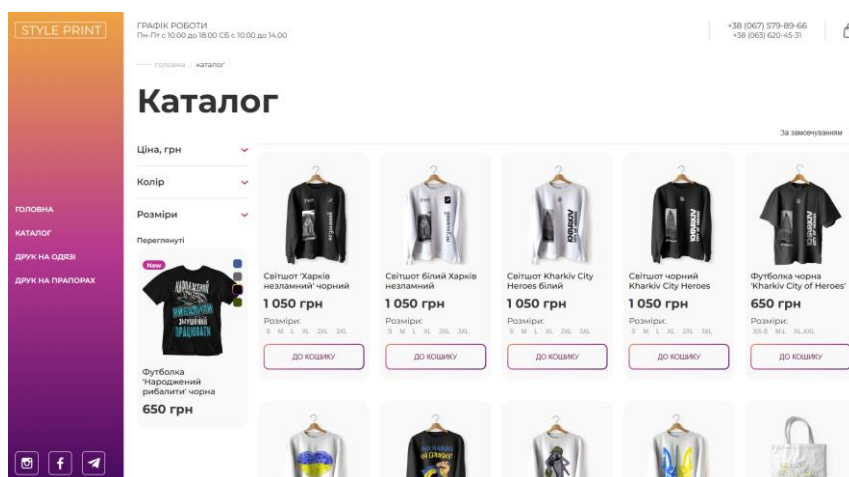


Рисунок 1.2 – Сторінка каталогу на сайті Style Print [5]

На перший погляд, сайт привертає увагу витонченістю та гарним дизайном, що значно полегшує взаємодію з ним. Зрозуміла структура навігації та чітке розміщення продукту дозволяють швидко знайти потрібний продукт.

Основною перевагою цього сайту є наявність усіх необхідних контактів для спілкування з адміністраторами, що зручно у разі виникнення питань чи проблем. Крім цього, процес вибору товару на веб-сайті простий та інтуїтивно зрозумілий завдяки чітким фільтрам і категоріям, що дозволяє легко знайти

потрібний товар за різними критеріями. Однак для підвищення ефективності пошуку можна реалізувати вдосконалений фільтр за характеристиками, застосування синонімів у пошукових запитах, а також реалізувати «розумний пошук» з автодоповненням, який аналізує популярні запити. Додатковим покращенням є сортування товарів за релевантністю, популярністю або рейтингом. Важливо також забезпечити повну відповідність сторінок фільтрів SEO-вимогам, що позитивно впливає на індексацію в пошукових системах.

Водночас, попри ці численні переваги, сайт має деякі слабкі місця, які можна було б покращити. Однією з ключових причин є відсутність автентифікації користувача та створення облікового запису. Це обмежує покупців зберігати історію своїх замовлень, отримувати персоналізовані рекомендації або здійснювати повторні покупки без повторного введення інформації. Авторизація не тільки спрощує процес покупки для відвідувачів, але й відкриває ширші можливості для взаємодії з користувачем через персоналізовані пропозиції. Крім того, наявність облікового запису створює передумови для застосування програми лояльності, накопичення бонусів, використання знижок і купонів. Персоналізований підхід також дозволяє надсилати індивідуальні пропозиції на електронну пошту або через месенджери. Система рекомендацій на основі переглядів та покупок користувача підвищує ймовірність здійснення нових покупок і покращує загальний користувацький досвід.

До того ж, сайт дозволяє спілкуватися через Telegram, і як ідея для власної розробки реалізувати Telegram-бота, який значно полегшить процес обробки замовлення. Бот може автоматизувати декілька процесів, наприклад, сповіщати менеджерів про нові замовлення, перевіряти деталі замовлення і редагувати їх. Це зменшує навантаження на адміністраторів.

Варто розглянути ще один аналог проєктованої системи, Konfiskat [6] – популярний інтернет-магазин, який спеціалізується на продажу одягу та інших товарів. Сайт пропонує широкий асортимент товарів за доступними цінами та

має сильну орієнтацію на ринок України, на рисунку 1.3 зображено головну сторінку сайту.

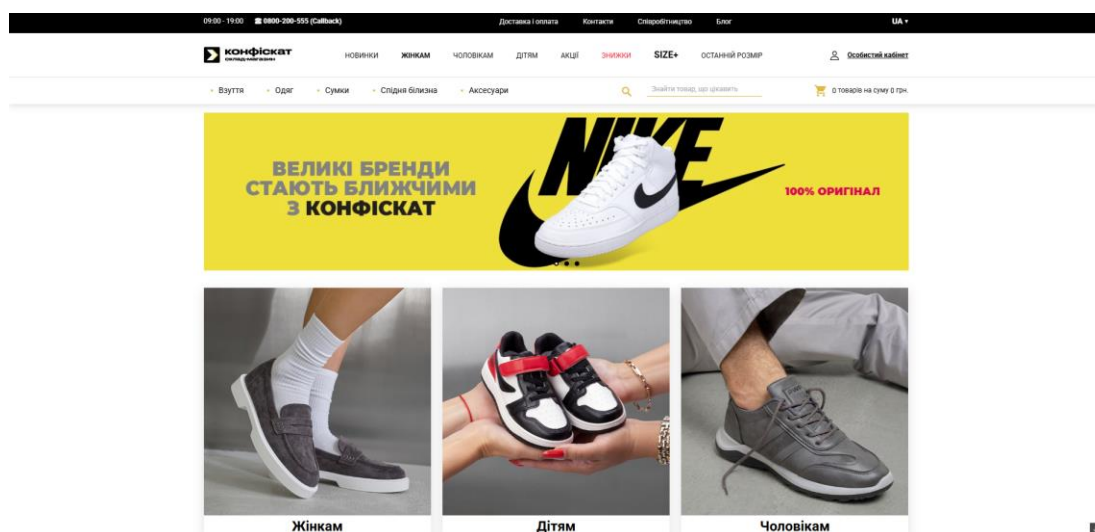


Рисунок 1.3 – Головна сторінка Konfiskat [6]

Серед переваг даного сайту можна виділити те, що є можливість створення особистого акаунта. Це дає змогу користувачам зберігати дані про попередні покупки, а також миттєво оформлювати нові. Є певний перелік варіантів зворотнього зв'язку, що теж є добре. Інтернет-магазин також пропонує великий вибір одягу, а ще взуття і різних аксесуарів. Зручна структура каталогу може допомогти користувачам швидко знаходити потрібні товари.

Варто також виділити недоліки цього сайту, хоча вони не критичні, але присутні. Першим недоліком є надмірна кількість тексту на головній сторінці внизу. Було б краще винести інформацію в окремі сторінки. Також поле пошуку розташоване вкрай непомітно й не пропонує підказок (autocomplete).

Відштовхуючись від здійсненого раніше аналізу сайтів, є сенс зосередитись на сильних та слабких сторонах, що допоможе у досягненні максимальної якості кваліфікаційної роботи. Незважаючи на стрімкий розвиток електронної комерції та збільшення обсягів ринку за останні періоди, існуючі інтернет-магазини нерідко характеризуються застарілим дизайном інтерфейсу, недостатньою адаптивністю та слабкої персоналізації, що призводить до втрати клієнтів та зниження конкурентоспроможності. Створення нового інтернет-

магазину з сучасним UX/UI-дизайном, інтегрованими інструментами соціальної комерції дасть змогу не тільки краще відповідати потребам користувачів різних пристроїв, але й підвищити рівень довіри та прихильності клієнтів, що в умовах високої конкуренції є визначальним чинником успіху.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

У процесі виконання кваліфікаційної роботи бакалавра необхідно досягти низки важливих цілей, які спрямовані на створення веб-додатка з інтеграцією сучасних технологій. Ці завдання включають в себе:

- 1) провести аналіз предметної області, проаналізувати існуючі програмні рішення;
- 2) визначити вимоги до розроблюваного веб-додатка;
- 3) вибрати засоби для реалізації веб-додатка;
- 4) розробити веб-додаток завдяки вибраними засобами;
- 5) проаналізувати найпопулярніші засоби тестування проєкту, провести тестування обраними серед них;
- 6) розробити структуру і створити бази даних;
- 7) створити Telegram-бота для інтернет-магазину;
- 8) реалізувати захист веб-додатка.

Висновки до розділу 1

Розглядаючи нинішній стан розвитку інтернет-магазинів та ключові тренди електронної комерції, можна зробити деякі важливі висновки.

По-перше, стрімкий розвиток електронної комерції відкриває нові горизонти для бізнесу, зокрема, забезпечуючи комфортні покупки, автоматизуючи процеси та інтегруючи передові рішення, як-от месенджери та чат-боти. Ринок демонструє стійке зростання, незважаючи на зовнішні виклики, а прогнози відновлення після економічного спаду в Україні підтверджують

великий потенціал цієї сфери.

По-друге, користувачі висувають високі вимоги до функцій інтернет-магазину, таких як швидкий доступ до даних, зручність користування, підтримка різних платформ і пристроїв, а також багатомовність. Це ставить перед розробниками нові, складні завдання. Для забезпечення конкурентної позиції продукту, необхідно враховувати як технічні, так і маркетингові аспекти.

По-третє, автоматизація становить важливу складову ефективної роботи сучасного інтернет-магазину. Застосування Telegram-бота та інтеграція з месенджером є ефективним способом покращити взаємодію з клієнтами та оптимізувати робочий процес. Telegram-бот може бути інтегрований із CRM або ERP системою підприємства, що дає змогу менеджерам оперативно отримувати оновлення, реагувати на замовлення та вносити зміни в режимі реального часу. Наприклад, бот може автоматично надавати інформацію про товари, їхні залишки, надходження нових партій, що суттєво прискорює процес обслуговування. Також передбачена можливість підключення платіжних сервісів для здійснення оплати безпосередньо через бот, що ще більше спрощує процес покупки для клієнта.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення

Моделювання програмних систем – це процес створення абстрактного образу або моделі системи. Вони допомагають нам збагнути, створити та описати структуру, функціональність та взаємодію різних компонентів системи. Це дасть можливість розробникам, аналітикам та іншим учасникам проєкту коректно зрозуміти, як діє система та як користувачі її використовують.

До основних видів моделей для моделювання програмних систем можна віднести:

1) UML (Unified Modeling Language), набір загальноприйнятих діаграм, які застосовують для моделювання об'єктно-орієнтованих систем; UML містить діаграми класів, діаграми варіантів використання, діаграми послідовностей і діаграми діяльності;

2) ERD (Entity-Relationship Diagram), застосовується для моделювання даних, вона допомагає спроектувати базу даних, визначаючи сутності та зв'язки між ними;

3) Domain-Driven Design (DDD), підхід до проєктування програмних систем, який акцентується на усвідомленні взаємозв'язку між предметною областю та технічними аспектами, він орієнтований на створення моделей, що відображають реальні бізнес-процеси та домени;

4) RESTful API модель, вона забезпечує взаємодію через API, створюється REST API-модель, вона дає змогу визначити, які кінцеві точки API можна викликати, які HTTP-методи (GET, POST, PUT, DELETE) підтримуються та яку інформацію відправляє або отримує запит.

Таким чином, моделювання є досить важливим етапом у розробці програмної системи та допомагає спростити процес проєктування, зменшити кількість помилок і забезпечити ясність процесу для всіх учасників проєкту.

Даний матеріал роз'яснює, як різні аспекти моделі даних Ed-Fi (Unitified Data Model) виражаються через вищезгадані моделі UML, REST API [7].

В ході аналізу моделей, свій вибір було обрано на користь UML, зважаючи на просту структуру діаграм. Однією з основних діаграм UML є діаграма класів, яка дозволяє моделювати структуру системи, відображаючи класи, їх атрибути і методи та взаємозв'язки між класами, що дуже схоже зі структурами бази даних.

На рисунку 2.1 зображено UML діаграму класів бази даних, розроблена для реалізації веб-додатку.

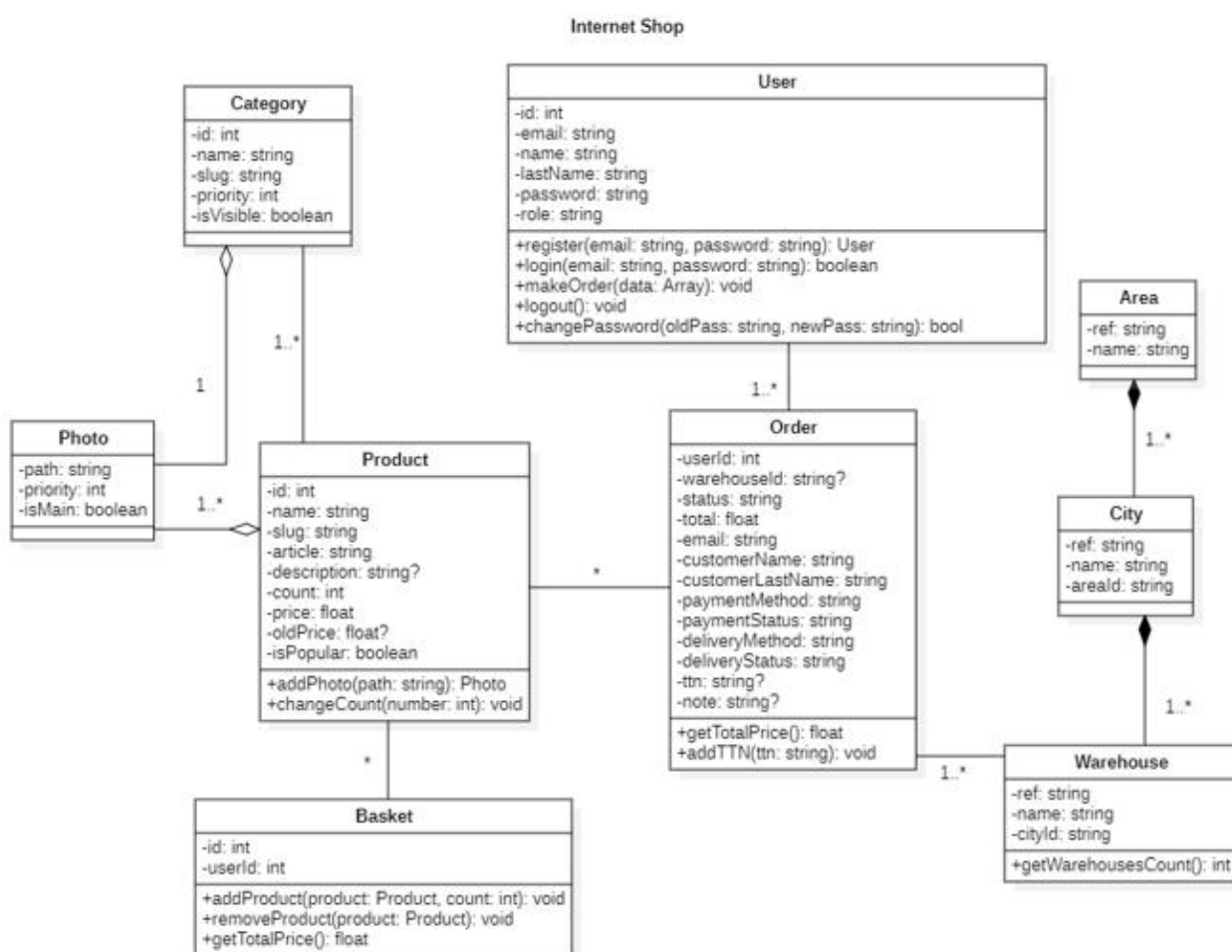


Рисунок 2.1 – Діаграма класів з моделюванням бази даних

Слід розібрати діаграму класів, котра змальовує базу даних. Класи, такі як: Area, City, Warehouse, вони описують області, міста та відділення, куди можна буде доставити придбаний в інтернет-магазині товар. Ключовими даними є їхні назви і зв'язки один з одним.

Безумовно в базі даних треба зберігати дані користувачів, що будуть користуватися інтернет-магазином, тому в діаграмі класів є описаний клас User, який містить ключові відомості, такі як: прізвище, ім'я, номер телефону, електронна пошта, роль користувача в інтернет-магазині та пароль, за допомогою якого користувач зможе зайти в особистий обліковий запис.

Отож, головним класом в інтернет-магазині є клас Product, що вміщує інформацію про товарів. До того ж, товари можна відносити до певних категорій, для цього створений клас Category в діаграмі. Як в товарах, так і в категоріях повинні бути зображення, яке буде асоціюватись, тому доданий зв'язок з класом Photo. Слід зазначити, що в товарів зазвичай більше одного зображення, відповідно використовується зв'язок «один до багатьох», тобто в базі даних багато зображень будуть посилатись на один конкретний товар.

Ключовим моментом інтернет-магазину є можливість додавання товарів у кошик, а тому потрібно зберігати інформацію про те, який товар до якого кошика належить, і також, який кошик належить до якого користувача. В такій ситуації використовується зв'язок «багато до багатьох», і практичним рішенням є реалізація додаткової таблиці.

Користувач має змогу замовляти продукцію, і для цього необхідно фіксувати інформацію про замовлення. Клас Order включає такі дані: загальна сума до оплати, статуси доставки і оплати, методи доставки і оплати, примітка, яку покупець міг залишити у формі. В класі Order є зв'язок з класом Product, і тут зв'язок «багато до багатьох», отже також потрібна проміжна таблиця, що буде комбінувати два ці класи.

Відтак, головне призначення цієї діаграми класів можна назвати представлення структури бази даних в зрозумілій формі. Розробнику буде значно простіше розібратись в структурі, чи навіть практичної її реалізації. Також, інтуїтивно зрозуміле візуальне представлення дозволяє поширювати її клієнтам для узгоджень і виправлення потенційних нюансів.

Також серед діаграм UML часто використовується діаграма варіантів використання (Use Case Diagram). В даному випадку, розроблена діаграма

використання описує взаємодію користувачів з системою інтернет-магазину, діаграма зображена на рисунку 2.2.

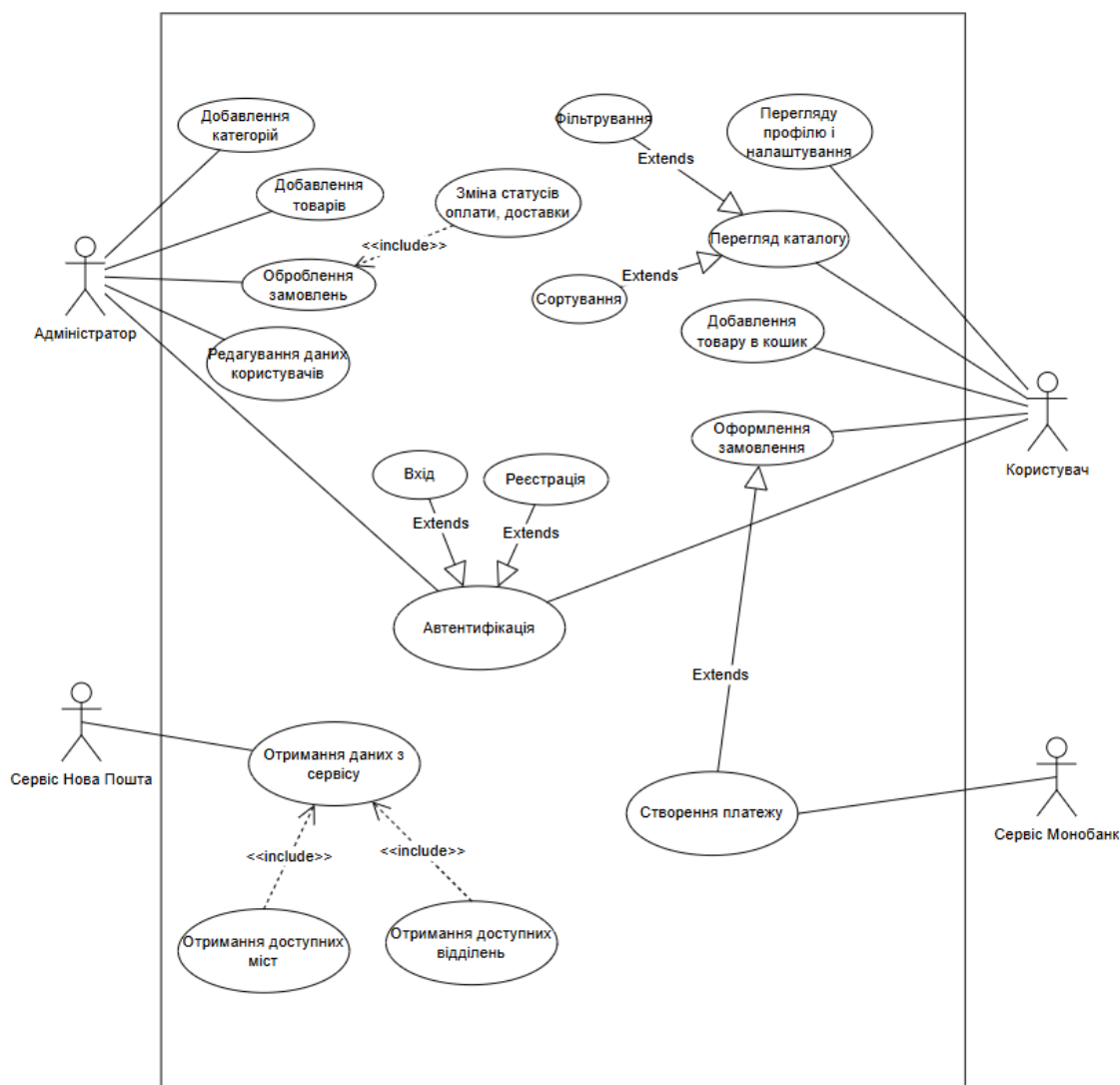


Рисунок 2.2 – Діаграма варіантів використання системи інтернет-магазину

Діаграма відображає варіанти використання для системи інтернет-магазину, показуючи тим самим взаємодію користувачів з доступними їм функціями. В системі є два типи користувачів: звичайний користувач та адміністратор.

Адміністратор наділений розширеними можливостями для керування системою. Він може додавати категорії товарів, що дає змогу групувати продукцію для зручнішої навігації каталогом. Також адміністратор має право додавати товари, введення нових позицій в систему разом з їх описом, ціною, та

іншими характеристиками. Крім того, його відповідальність включає обробку замовлень. Це значить, що він може підтверджувати, скасовувати або редагувати статуси замовлення, які були зроблені користувачами. Буває іноді потреба в редагуванні даних користувачів, коли користувач ввів неправильно ім'я, прізвище чи не може згадати пароль. Тому хорошим функціоналом буде можливість адміністратору змінити ці дані при потребі, наприклад відновити новий пароль щоб користувач зміг зайти у свій обліковий запис.

Звичайний користувач має свої можливості в системі. Він може переглядати каталог товарів, де представлені всі товари. Важливо реалізувати фільтри та сортування для кращого пошуку товару під вимоги користувача. Якщо певний товар зацікавив, він може додати його у кошик для подальшого оформлення замовлення. Після того як користувач вибере всі необхідні товари, він може перейти до оформлення замовлення, що включає вибір способу доставки, оплати, введення персональних даних, при бажанні введення примітки, яку може переглянути адміністратор під час оброблення замовлення. Також користувач має можливість перегляду свого профілю та налаштувань, де він зможе змінювати особисті дані, наприклад ім'я, прізвище, пароль, номер телефону.

Окремо на діаграмі показаний процес аутентифікації. Зважаючи на дослідження вже готових рішень та їх мінусів, серед котрих була відсутність облікового запису, хорошим рішенням є реалізації автентифікації і можливість мати облікового запису в інтернет-магазині, який розробляється. Для користування всіма можливостями покупець матиме змогу зареєструватися, а якщо вже є обліковий запис то увійти в нього. Також, для доступу до керування інтернет-магазином адміністратор має пройти авторизацію. Вхід і реєстрація є частиною загального процесу автентифікації, який гарантує, що доступ до персональних даних та функцій системи отримують лише авторизовані користувачі. Отже, діаграма відображає логічну структуру функцій в інтернет-магазині. Розмежовуються можливості між адміністратором та рядовими користувачами. Вона демонструє ключові процеси, що відбуваються в системі.

На рисунку 2.3 зображено дизайн головної сторінки проєктованого веб-додатку у світлій темі.

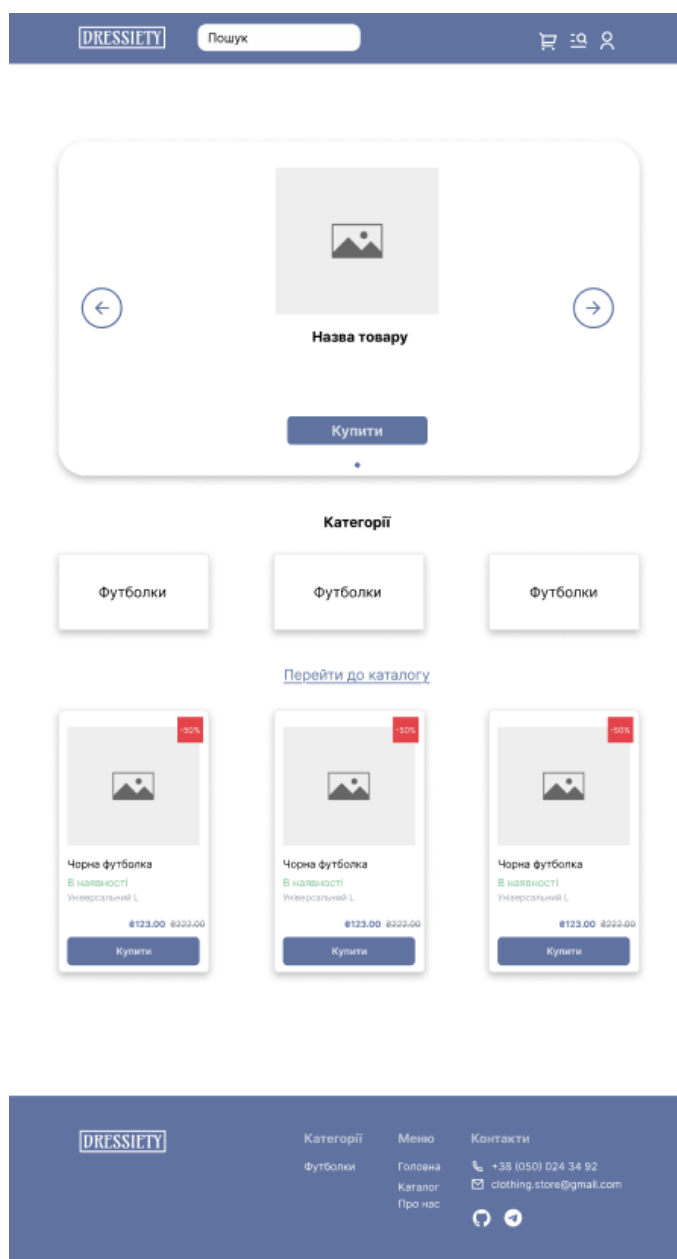


Рисунок 2.3 – Дизайн головної сторінки із світлою темою

Для інтернет-магазину є певні вимоги, що впливають на позитивне сприйняття віду користування. Дуже важлива підтримка темної теми на сайті. Реалізація важлива, оскільки вона зменшує навантаження на зір користувачів, особливо у вечірній час, що робить перегляд товарів значно комфортнішим. Варто також врахувати, що на пристроях з OLED та AMOLED дисплеями темна

тема сприяє економії заряду акумулятора. Значна частина користувачів віддає перевагу саме темній темі, тому її наявність робить сайт зручнішим і привабливішим для ширшої аудиторії. На рисунку 2.4 зображено головну сторінку з темною темою. У даній статті розкажується про всі нюанси реалізації темної теми, поради, які варто знати для реалізації хорошого інтерфейсу [8].

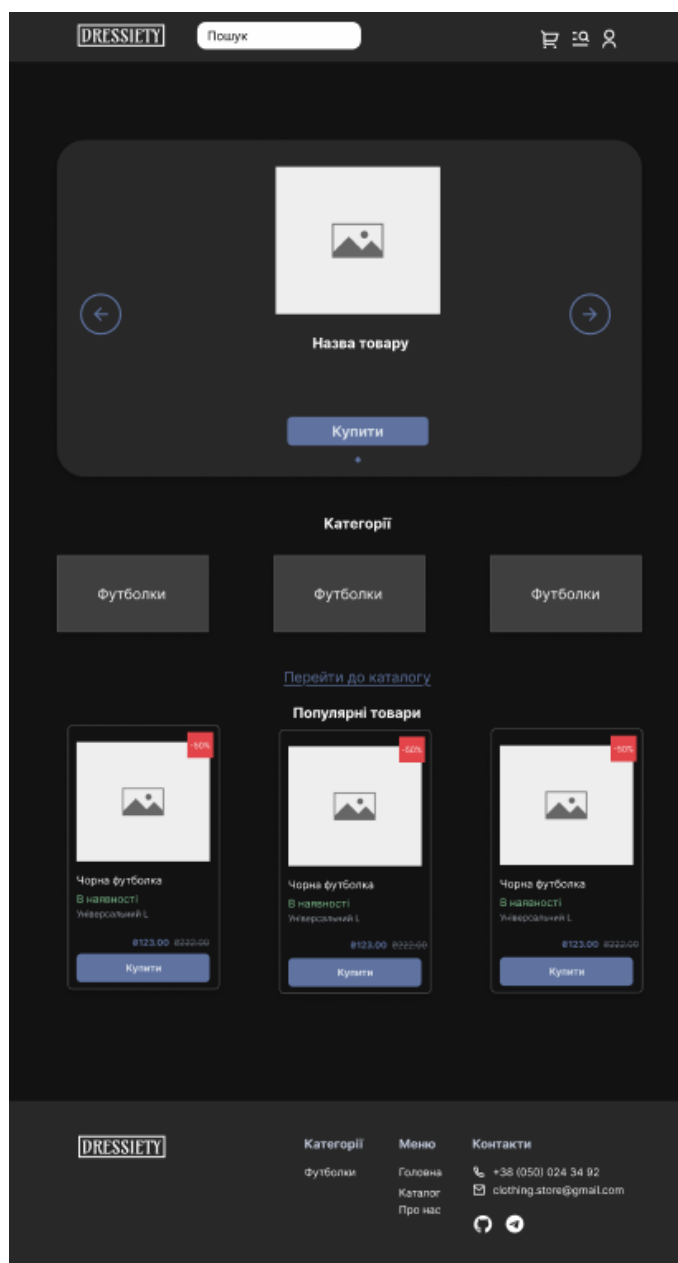


Рисунок 2.4 – Дизайн головної сторінки з темною темою

Окрім реалізації темної теми, також важлива адаптивність сайту, адже більшість користувачів відвідують його зі смартфонів, планшетів, лептопів і

десктопів, кожен з яких має власні габарити екрану. У випадку, коли сайт не адаптований, перегляд може стати надзвичайно незручним, наприклад, текст може бути надмірно великим або зовсім крихітним, або ж взагалі буде відображатись за межами екрана. Адаптивний дизайн гарантує коректне відображення контенту на будь-якому пристрої, тим самим покращуючи користувацький досвід. Також, це позитивно впливає на SEO.

Каталог в інтернет-магазині є основною частиною сайту, адже саме через нього покупці мають змогу відшукувати та вибирати потрібні речі. Зручність та функціональність каталогу прямо впливають на обсяги продажів. Якщо в каталозі представлено багато різних товарів, без правильних фільтрів та сортування користувачам буде складно швидко знайти потрібний продукт. Фільтрація дає можливість звужити вибір за певними параметрами, такими як ціна, розмір, що значно економить час покупця. Сортування допомагає впорядкувати товари за популярністю, назвою, ціною. Запровадження цих функцій суттєво полегшує пошук потрібних товарів в інтернет-магазині.

На рисунку 2.5 зображено дизайн сторінки каталогу з фільтраціями і сортуванням.

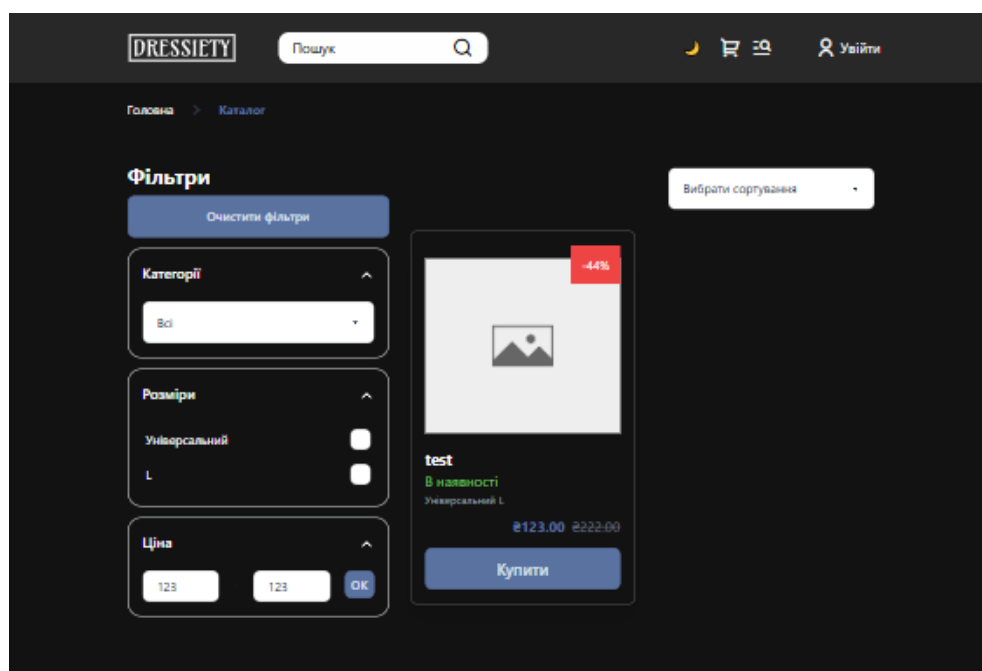


Рисунок 2.5 – Дизайн сторінки каталогу з фільтраціями і сортуванням

Автентифікація через пошту та номер телефону є важливою для зручності та безпеки користувачів в інтернет-магазині. Деякі користувачі віддають перевагу реєстрації через email, оскільки це традиційний спосіб, що дозволяє легко відновити доступ і отримувати повідомлення про замовлення. Але коли користувач бажає постійно заходити в обліковий запис через номер телефону, в нього має бути можливість прикріпити номер і заходити з ним а не через пошту, хоча можливість завжди є і з поштою. Завдяки цьому, коли в користувача прив'язана як пошта так і номер телефону, відновити аккаунт буде значно легше.

Сповіщення в інтернет-магазині відіграють важливу роль у взаємодії з користувачем. Вони інформують клієнтів про те, що замовлення створено, товар доставлено, оплата успішна і інші важливі події.

Під час аналізу існуючих рішень, було вирішено розробити Telegram-бота для інтернет-магазину, для автоматизації і спрощення завдань. Для реалізації потрібно вибрати доступну бібліотеку для створення бота, який забезпечить базову взаємодію. Для початку для інтернет-магазину хвататиме обробки замовлень адміністратором, коли буде потреба в розширенні функціоналу, завдяки обраній бібліотеці можна буде вирішити нові поставлені завдання.

Після реалізації сайту, потрібно буде його розмістити на хостингову платформу, вона повинна бути зручною. Також добре буде, якщо платформа має додаткові інструменти для вирішення базових потреб розміщення проєкту на сервер. Для кращого SEO, сайт повинен підтримувати SSL/TLS шифрування для безпеки даних, також для сайту потрібно призначити домен, щоб пошук був зручним і також для кращого SEO.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Веб-розробка – це процес створення веб-сайтів і веб-додатків, які працюють завдяки Інтернету. Розробка включає кілька етапів: дизайн, програмування, тестування та впровадження.

З-поміж найпопулярніших серверних мов програмування для розробки – PHP, Python, C#, Java. PHP є одним з найстаріших рішень, але все ще залишається найпопулярнішою мовою програмування для створення серверної частини програми, особливо в поєднанні з CMS, скажімо, WordPress, Drupal чи Joomla. Python набирає обертів завдяки легкості у вивченні та зрозумілості, читабельності коду, але краще підходить для розробки штучного інтелекту та аналізу даних. Java, а також C#, частіше застосовуються у великих корпоративних рішеннях через їх надійність та високу продуктивність.

У таблиці 2.1 зображено порівняння популярних мов, які використовуються у веб-розробці на серверній стороні.

Таблиця 2.1 – Порівняння популярних серверних мов програмування у веб-розробці

Критерій	PHP	Python	Java	C#
Простота вивчення	Простий для початківців, особливо з CMS.	Дуже простий, логічний синтаксис.	Складний для початківців.	Відносно складний через суворий синтаксис.
Популярні фреймворки	Laravel, Symfony, CodeIgniter.	Django, Flask, FastAPI.	Spring, Hibernate.	ASP.NET, Entity Framework.
Продуктивність	Залежить від конфігурації сервера.	Достатня, але менш ефективна в багато-поточкових задачах.	Висока продуктивність у великих системах.	Висока продуктивність, особливо у Windows-середовищі.
Популярність	Висока у сфері веб-розробки.	Широка популярність у різних галузях.	Висока в корпоративному секторі.	Популярний для корпоративних ігор та веб.
Затребуваність	Висока для CMS і середнього бізнесу.	Дуже висока через універсальність.	Висока у великих компаніях.	Висока, особливо у .NET-екосистемі.
Переваги	Легкість у вивченні, величезна кількість готових рішень.	Зрозумілий синтаксис, широкі можливості в різних галузях.	Висока продуктивність, стабільність, кросплатформеність.	Потужна інтеграція у Windows, ефективність у розробці додатків.

Продовження таблиці 2.1

Критерій	PHP	Python	Java	C#
Недоліки	Обмежена гнучкість поза веб-розробкою, відсутність деяких сучасних функцій.	Вища споживаність ресурсів у порівнянні з іншими мовами, проблема з багатопоточністю.	Складність налаштування середовища, високі вимоги до ресурсів для великих проєктів.	Залежність від платформи Windows, менша кількість інструментів для деяких типів програм.

Через простоту у вивченні, дуже великої популярності і затребуваності і великого асортимента фреймворків в даній реалізації буде використовуватись PHP.

PHP – це дуже поширена мова програмування для веб-розробки. PHP розшифровується як Hypertext Preprocessor, тобто PHP виконує свій код на сервері і відправляє веб-сторінку написану на HTML. Ця мова програмування є скриптованою, тобто виконує команди без попередньої компіляції в машинний код.

З'явилась мова програмування у світ в 1995 році розробником Рамусом Лердорфом як інструмент для генерації динамічних веб-сторінок. Призначенням спочатку була суто обробка простих веб-запитів. З часом мова розвивалась і стала одним із найпопулярніших інструментів для створення веб додатків.

Код виконується інтерпретатором PHP, це програма читає файл із PHP-кодом, аналізує і виконує. Інтерпретатор написаний на мові програмування C, вона забезпечує високу продуктивність і гнучкість. Також програма працює разом із серверними програмами через модулі, наприклад `mod_php` для Apache. Підтримується мультиплатформенність, завдяки чому працює на різних операційних системах, серед них: Linux, MacOS, Windows. Можна використовувати різні версії PHP-інтерпретатора для сумісності з певними додатками.

Фреймворком можна назвати набором інструментів, певних правил і бібліотек, завдяки чому спрощується розробка програмного забезпечення, зменшується кількість рутинних задач. Головна ідея фреймворків – це економія

часу, підвищення продуктивності програмного забезпечення, оскільки вони можуть запропонувати готові рішення для найпоширеніших задач, таких як робота з базами даних, автентифікація, маршрутизація, управління сесіями. Варто зазначити що PHP-фреймворки підтримують принципи MVC, що допомагає структурувати код, робить код більш читабельним та підтримуваним.

Аналізуючи статтю «PHP Framework for Web Application Development» можна вияснити, що серед найпопулярніших PHP-фреймворків є Laravel, Symfony, CakePHP, CodeIgniter [9]. Звісно кожен з них має переваги і недоліки, в таблиці 2.2 відображається порівняння цих фреймворків.

Таблиця 2.2 – Порівняння популярних PHP-фреймворків

Критерій	Laravel	Symfony	CodeIgniter	CakePHP
Дата виходу	2011	2005	2006	2005
Популярність	Дуже популярний	Популярний	Середня популярність	Низька популярність
Легкість освоєння	Легкий	Складний	Дуже легкий	Середній
Переваги	Простота, велика екосистема пакетів, хороша документація	Гнучкість, масштабованість, компонентний підхід	Простота, швидкість	Легка документація, вбудовані функції безпеки
Недоліки	Вимогливий до ресурсів, може бути повільним	Високий поріг входу	Обмежений функціонал, менша спільнота	Застарілі підходи, менша спільнота

На основі таблиці можна зрозуміти, що кожен PHP-фреймворк має свою унікальну нішу і підходить для різних типів проєктів. Однак Laravel вигідно виділяється своєю простотою, багатою екосистемою та активною спільнотою. Документація дуже детально описана [10], і проста в розумінні, завдяки чому початківець може легко почати розробляти своє програмне забезпечення. Велика кількість проєктів, які пишуться на Laravel фреймворку, є саме інтернет-магазини, через велику кількість готових інструментів, а саме автентифікації, доступу до бази даних тощо.

Незважаючи на те, що він може бути трохи вибагливим до ресурсів і менш оптимізованим для масштабованих проєктів порівняно з Symfony або Phalcon,

серед великої кількості розробників Laravel стає першим вибором завдяки своїй доступності та гнучкості.

Досліджуючи можливості сучасних фреймворків для розробки веб-додатків, важливим є досвід застосування архітектурного шаблону MVC у Laravel, описаний у даній статті [11]. Автори демонструють, як використання Laravel спрощує створення електронних комерційних платформ завдяки структурному розділенню моделі, представлення та контролера (MVC), інтеграції з RESTful API, а також інструментам командного рядка Artisan. У статті детально висвітлено етапи розробки, включаючи UML-діаграми, ER-модель бази даних та реалізацію ключових функцій. Це дослідження підкреслює ефективність Laravel у контексті швидкої розробки проєктів.

Головним аспектом в програмному забезпеченні часто буває база даних. Вона дозволяє зберігати, організовувати і також швидко отримувати інформацію, необхідну для програмного забезпечення. Без бази даних складно уявити багатофункціональні веб-додатки, не було б можливості зберігати користувацькі дані, транзакції, історії активності, або налаштування, особливо враховуючи безпеку і конфіденційність. Бази даних забезпечують дуже ефективну обробку даних а також дозволяються працювати з великими обсягами даних, обсяги можуть сягати цілими гігабайтами, без сильного зниження продуктивності.

Для інтернет-магазину база даних має особливу важливість. Завдяки ній ми можемо зберігати інформацію про товари, їхні характеристики, ціни, наявність на складі, дані користувачів, замовлення тощо. Це дозволяє додавати, редагувати, видаляти товари, адміністратор зможе відслідковувати статуси замовлень, виконувати їх обробку, користувачі зможуть переглядати свою історію покупок. Інтернет-магазин без бази даних був би неефективним а більшість операцій були б неможливими.

Дослідження в статті «Database Management Education in MySQL» детально аналізує функціональні можливості MySQL як системи управління реляційними базами даних, підкреслюючи його роль у сучасних інформаційних

системах [12]. Автори описують історію розвитку MySQL, його ключові переваги – відкритість коду, мультиплатформеність, високу продуктивність при обробці великих обсягів даних, а також інтеграцію з різними мовами програмування. У статті зазначено, що MySQL підтримує транзакційні та нетранзакційні операції, що робить його придатним як для веб-додатків так і для бізнес-систем.

Важливо визначитись в якому ПЗ буде розроблятися проєкт, чи це текстовий редактор, чи повноцінне середовище. Хорошим варіантом є VS Code, оскільки він один з найпопулярніших текстових редакторів для розробки. Одна з основних переваг – це його легкість та швидкість. Він запускається майже миттєво і працює без значного навантаження на систему, це робить його ідеальним навіть для менш потужних комп'ютерів. На рисунку 2.6 зображено вигляд даного редактора.

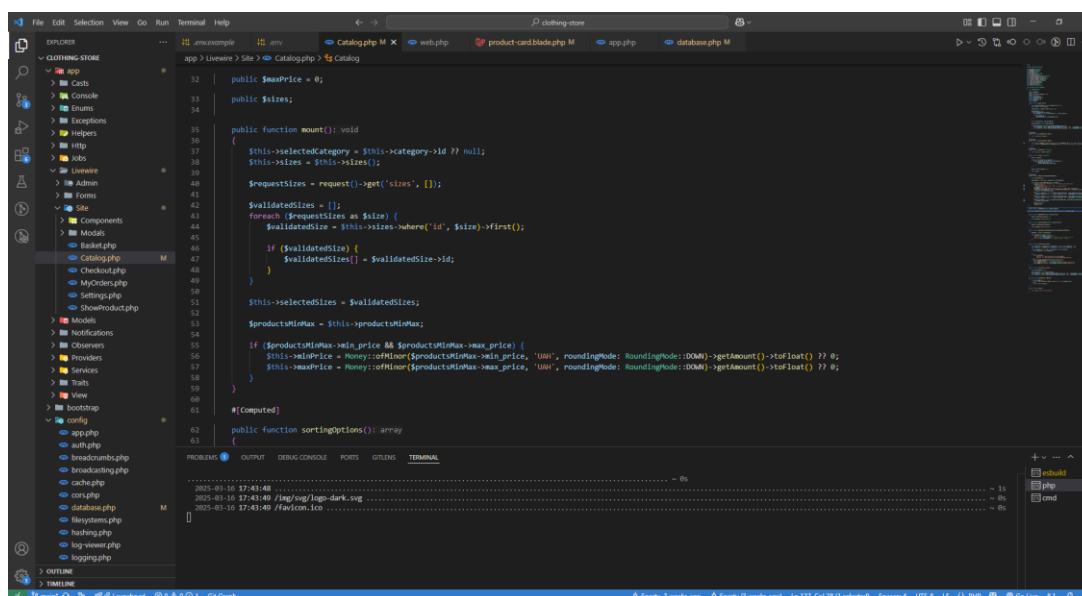


Рисунок 2.6 – Вигляд редактора VS Code

VS Code також має велику кількість розширень для розробки Laravel додатків з використанням PHP. Завдяки підтримці таких розширень, як PHP Intelephense, можна отримати автодоповнення коду, навігацію по класах і методах, а також підсвічення помилок в коді.

Ще однією перевагою є вбудована підтримка Git, що дозволяє працювати з системою контролю версій безпосередньо з редактора. Також можна створювати різні типи терміналів і запускати команди також безпосередньо з редактора.

Таким чином, VS Code забезпечує зручне, швидке середовище для розробки Laravel додатків.

У розробці програмного забезпечення буде використовуватися Livewire для реалізації інтерактивних компонентів на стороні клієнта без необхідності писати складний JavaScript код. Завдяки цьому значно можна спростити розробку динамічних елементів інтернет-магазину, таких як кошик, фільтрація товарів та оновлення замовлень у реальному часі.

Livewire – це бібліотека для Laravel, вона надає можливість створювати сучасні інтерактивні інтерфейси. Працює шляхом взаємодії з сервером через AJAX-запити, оновлюючи тільки необхідні частини сторінки. Завдяки цьому можна отримати динамічність без повного перезавантаження сторінки, це покращує користувацький досвід. Livewire також дозволяє дуже легко створювати компоненти, які містять як HTML, так і логіку PHP.

Як зазначає автор у статті, Livewire є потужним інструментом для створення динамічних веб-інтерфейсів у рамках екосистеми Laravel [13]. Автор підкреслює, що Livewire дозволяє розробникам реалізовувати інтерактивні функції, такі як двостороннє зв'язування даних, оновлення DOM у реальному часі та управління станом компонентів, використовуючи переважно PHP, що значно зменшує залежність від JavaScript. Стаття розкриває архітектурні особливості Livewire, зокрема взаємодію між клієнтом і сервером через AJAX-запити, а також механізми автоматичного перерендерингу компонентів. Додатково обговорюються переваги інтеграції з Laravel, зокрема використання Blade-шаблонів та безпечну обробку даних.

Laravel має вбудовану підтримку Vite, це сучасний інструмент для зборки фронтенду, який Laravel використовує за замовчуванням починаючи з версії 9, раніше використовувався Laravel Mix. Vite забезпечує швидку розробку, гарячу

оновлення модулів і продуктивну збірку для продакшну. Миттєве перезавантаження завдяки оновлення модулів змінює CSS, JS без перезавантаження сторінки. Підтримується проста інтеграція з TypeScript, Vue, React, Tailwind CSS без складної конфігурації.

Для стилізації сторінок буде використовуватись CSS фреймворк Tailwind CSS. Він дозволяє швидко стилізувати інтерфейси, використовуючи утилітарні класи без написання власного CSS.

На рисунку 2.7 зображено використання класів Tailwind CSS в коді HTML, а також зображено результат коду.

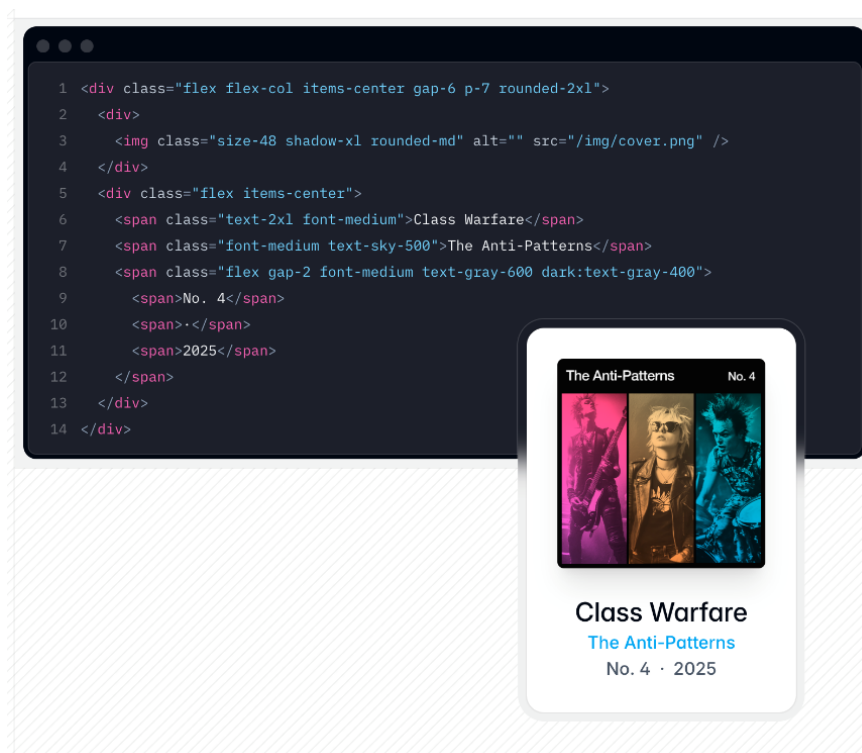


Рисунок 2.7 – Зображення використання Tailwind CSS класів у коді і дизайн в результаті коду [14]

Основні переваги Tailwind CSS:

- 1) замість попередньо створених стилів надаються класи, які можна комбінувати для будь-якого дизайну;
- 2) немає потреби створювати окремі CSS файли, усі стилі додаються прямо в HTML теги;

3) завдяки PurgeCSS видаляються всі стилі які не використовуються, тому у продакшні файли вагнуть значно менше;

4) можна змінювати кольори, відступи, тіні і інші параметри завдяки файлу конфігурації `tailwind.config.js`.

Telegram служить хорошим інструментом для сповіщення клієнта, або місцем для реалізації певного функціоналу для бізнесу. Хоча повноцінний функціонал займає не малі ресурси, мінімальний функціонал для даного програмного забезпечення зможе відобразити можливість і перспективу даного напрямку. Вибрано для розробки «Telegram Bot SDK for PHP» [15] для інтеграції бота в інтернет-магазин обумовлений його оптимальною сумісністю з екосистемою Laravel та широкими функціональними можливостями. SDK надає зручний інтерфейс для роботи з Telegram API, що дозволяє швидко реалізовувати ключові задачі. Важливою перевагою є підтримка вебхуків, яка забезпечує миттєву реакцію бота на події без необхідності використовувати Long Polling. Бібліотека також інтегрується з Eloquent ORM, що спрощує роботу з базою даних, а детальна документація та активна спільнота значно знижують поріг входження для розробників. Цей вибір дозволяє розробити простий але зручний телеграм-бот, з можливістю подальшого розширення.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи здійснено поглиблений аналіз, обґрунтування та створення плану технологічного стеку, який використовуватиметься для розробки інтернет-магазину з вбудованим Telegram-ботом.

Laravel був обраний як ключовий фреймворк для розробки бекенду через його архітектуру MVC, зрозумілу документацію, великій спільноті та готовим рішенням для автентифікації, взаємодії з базами даних та маршрутизації. В порівнянні з Symfony чи CodeIgniter, Laravel демонструє більш сучасний підхід

до розробки, що є вирішальним для швидкої реалізації складних функцій, зокрема, обробки замовлень або інтеграції з API.

MySQL було обрано як СУБД через її сумісність з Laravel, чудову продуктивність під час операцій з транзакціями, підтримку реляційних відносин та забезпечення безпеки даних. Нормалізація структури бази даних використовується для уникнення надлишковості даних.

Livewire було використано для створення динамічних компонентів, що дозволяє уникнути необхідності використовувати складні JavaScript-фреймворки. Це сприяє зниженню навантаження на сервер і спрощує процес синхронізації між фронтендом та бекендом.

Tailwind CSS було обрано задля гнучкості, модульності та здатності будувати адаптивні інтерфейси. Застосування утилітарних класів прискорює процес розробки та полегшує втілення темних/світлих тем.

Для розробки Telegram-бота обрано бібліотеку «Telegram Bot API for PHP», яка легко інтегрується з Laravel.

Ці технічні засоби гарантують високу ефективність, безпеку функціонування, комфорт розробки та здатність до масштабування. Береться до уваги недоліки деяких вже наявних рішень. Застосовані UML-діаграми (класів, випадків використання) дозволяють ретельно спроектувати архітектуру системи, що знижує вірогідність помилок на стадії кодування. Майбутній розвиток проекту може передбачати інтеграцію з міжнародними платіжними сервісами, впровадження двофакторної авторизації та локалізацію.

РОЗДІЛ 3

РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ З ПІДКЛЮЧЕНИМ TELEGRAM-БОТОМ З ВИКОРИСТАННЯМ LARAVEL ТА MYSQL

3.1 Практична реалізація об'єкта проєктування

У рамках кваліфікаційної роботи реалізовано інтернет-магазин з тематикою одягу для чоловіків. Для реалізації програмної частини обрано мову програмування PHP та фреймворк Laravel, а для зберігання даних – реляційну систему управління базами даних MySQL.

На рисунку 3.1 зображено структуру проєкта даної розробки з використанням фреймворку Laravel.

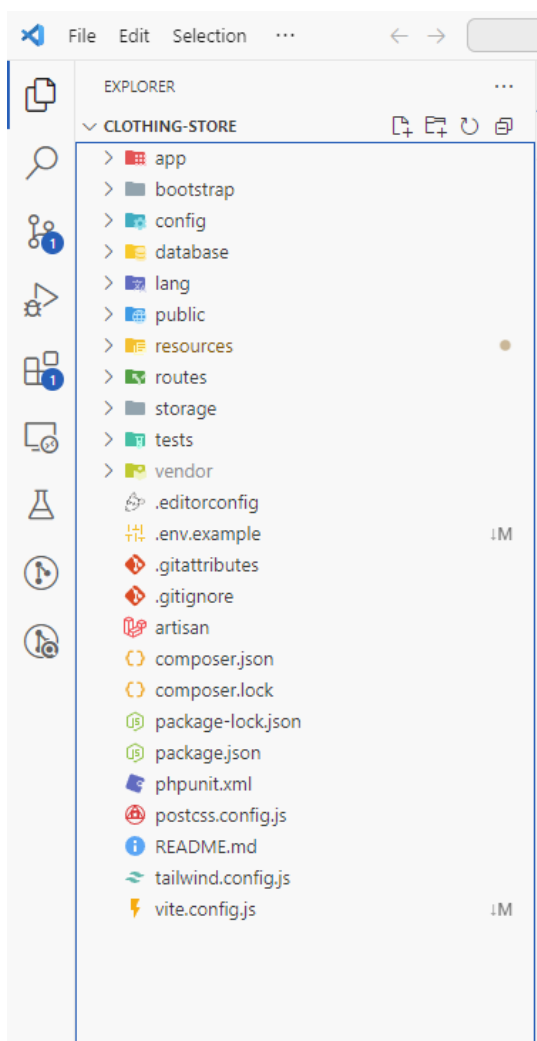


Рисунок 3.1 – Структура проєкту з фреймворком Laravel

Laravel містить пару ключових папок, які використовуються найчастіше:

- App – основна логіка застосунку, моделі, які представляють дані таблиць. Контролери, які містять ендпоінти (англ. endpoint) сайту;
- Services, які містять бізнес логіку, код у цій папці виконує основну роботу із запитами, базою даних та логікою застосунку;
- Routes містить файли для переліку веб-маршрутів по яким можна буде звертатись з браузера;
- Resources містить Blade-шаблони (HTML з поєднанням PHP), незкомпільовані CSS і JS для фронтенду. Цей каталог тісно пов'язаний із представленням у шаблоні MVC;
- Database призначена для роботи з базою даних. Тут розміщуються міграції (структура таблиць), сидери (початкові дані для тестування) та фабрики (для генерації тестових записів). Це ядро для наповнення бази. «.env» – це прихований файл у корені проєкту, який містить конфіденційні змінні середовища, так як доступ до бази даних, ключі API і інші.

Щоб були сторінки для панелі адміністратора, був створений файл `admin.php` в каталозі `routes`. Там вказуються всі потрібні ендпоінти. Так як до панелі адміністратора може мати доступ лише довірений користувач, був добавлений «middleware» на всі ендпоінти, які знаходяться у файлі `admin.php`. Даний middleware перевіряє чи має користувач роль адміністратора. Оголошений middleware знаходиться у файлі `Kernel.php` в каталозі `Http`, а використаний в `RouteServiceProvider.php`.

Щоб запити були обробленими потрібні контролери, які містять ендпоінти. Зазвичай в панелі адміністратора контролери слугують для реалізації CRUD (create, read, update, delete), тобто створювати, отримувати, оновлювати чи видаляти дані.

На рисунку 3.2 зображено структуру проєкту із переліком контролерів, також зображено вміст `ProductController.php` контролера, який містить 3 ендпоінта.

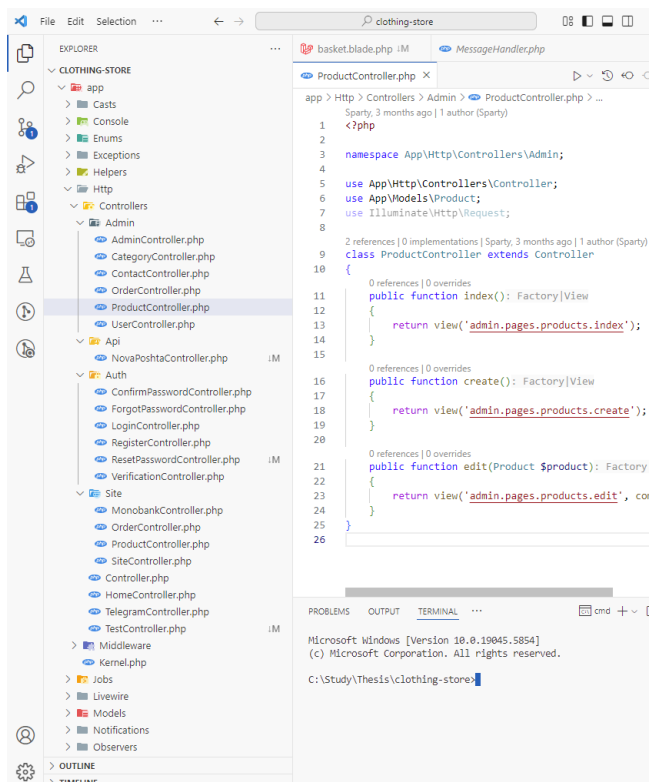


Рисунок 3.2 – Перелік контролерів в проєкті і зміст ProductController.php

Index метод в контролері дає можливість перегляду, зазвичай в ній знаходиться таблиця, можливо з фільтруванням і сортуванням для пошуку конкретних даних. В create методі вертається сторінка форми для заповнення даних, щоб створити модель, а в edit методі форма, в яку загружається вже існуюча модель і дається змога редагувати ці дані. В даній ситуації ми досягаємо цих методів через HTTP запит типом GET. В чистому Laravel можна реалізувати методи до яких можна досягти іншими типами HTTP запитів, але в даній кваліфікаційній роботі використовується Livewire, в якій прописується бекенд логіка, і тому в тих компонентах є логіка для видалення даних, занесення в базу даних чи редагування.

У процесі розробки інтернет-магазину було реалізовано доволі зручну систему автентифікації користувачів, вона дозволяє входити в обліковий запис як за допомогою пошти електронної, так і через номер телефону. Такий підхід забезпечує більшу гнучкість для користувачів, вони можуть обирати зручний для них спосіб авторизації. На рисунку 3.3 зображено зовнішній вигляд сторінки входу в обліковий запис.

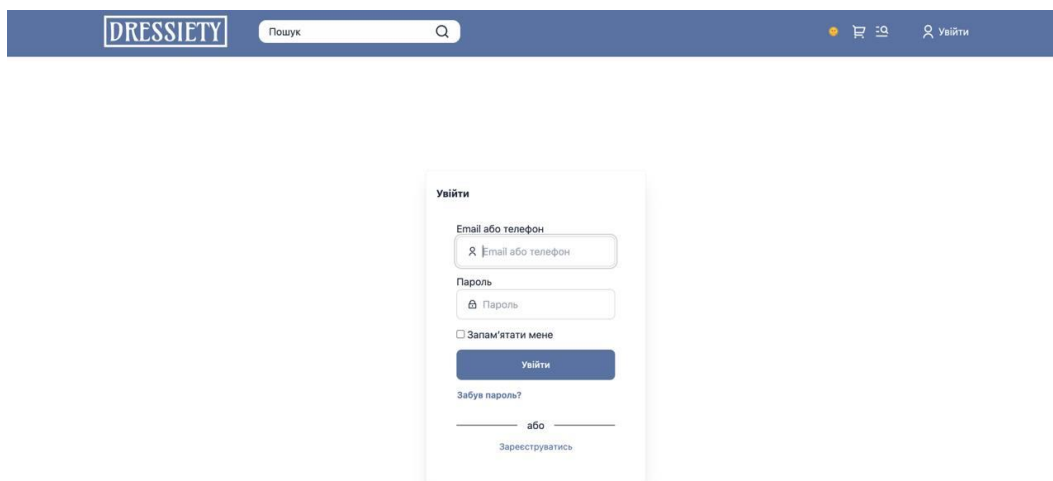


Рисунок 3.3 – Зовнішній вигляд сторінки входу в обліковий запис

Під час входу система автоматично визначає, чи введене значення є електронною поштою чи номером телефону, відповідно до цього виконується перевірка користувача в базі даних. Для цього створено власну логіку автентифікації, яка обробляє обидва варіанти в одному полі вводу. Це значно спрощує інтерфейс і робить процес входу більш інтуїтивним.

В лістингу 3.1 зображено код кастомної автентифікації, яка обробляє обидва варіанти.

Лістинг 3.1 – Код обробки обох варіантів автентифікації

```
protected function attemptLogin(Request $request)
{
    $result = Auth::attempt($this->credentials($request), $request-
    >boolean('remember'));

    if ($result) {
        Log::info(auth()->user()->createToken('auth_token')-
        >plainTextToken);
    }
}

protected function credentials(Request $request)
{
    $login = $request->input('login');
```

```

        $field = filter_var($login, FILTER_VALIDATE_EMAIL) ? 'email' :
        'phone';

        if ($field === 'phone') {
            $login = new PhoneNumber($login, 'UA');
        }

        return [
            $field => $login,
            'password' => $request->input('password'),
        ];
    }

```

кінець лістингу 3.1

Також реалізована функція відновлення доступу до облікового запису. Якщо користувач не пам'ятає свій пароль, він може ініціювати процедуру відновлення через відповідну форму. Система надсилає код підтвердження або на вказану електронну пошту, або у вигляді SMS на номер телефону, залежно від того, яку контактну інформацію було використано при реєстрації. Після введення правильного коду користувач отримує можливість створити новий пароль.

У проєкті використовується Livewire для динамічного оновлення сторінок, для того щоб прискорити реалізацію фронтенд частини, використовуючи PHP і мінімум JavaScript коду.

Дуже важлива сторінка в інтернет-магазині це каталог. Саме на цій сторінці користувач зможе зручно знайти той товар, який він хоче. Повинні бути фільтрації і сортування по певним атрибутам. Отже було створено компонент Catalog, в catalog.php файлі прописана бекенд логіка: фільтрація, сортування, вибір категорії. У файлі catalog.blade.php зберігається HTML код з підтримкою PHP, на основі макету був написаний відповідний код, але також були прописані спеціальні директиви завдяки якому ми можемо використовувати PHP і звертатись до бекенд логіки Livewire. На справді це інструкції як звертатись, коли користувач клікне на кнопку, JavaScript код від Livewire зробить AJAX запит до сервера, виконається бекенд логіка і відправиться відповідь. JavaScript отримає цю відповідь і оновить сторінку при потребі. Отже, в catalog.blade.php

файлі добавлений дизайн фільтрацій, сортування, карток товарів, і завдяки директивам і Livewire будуть виводитись дані. На рисунку 3.4 зображено зовнішній вигляд реалізованої сторінки каталогу.

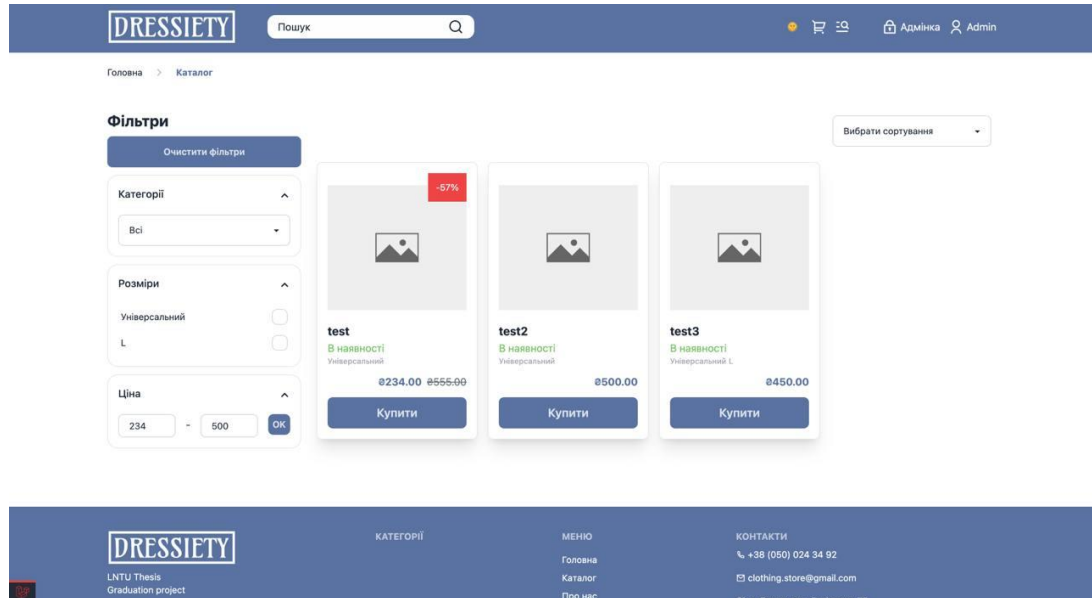


Рисунок 3.4 – Зовнішній вигляд сторінки каталогу

В лістингу 3.2 зображено головний метод з алгоритмом фільтрування і сортування для отримання потрібних товарів.

Лістинг 3.2 – Алгоритм отримання товарів на сторінці каталогу

```
#[Computed()]
public function products()
{
    $this->selectPrices();

    $selectedSort = SortProduct::tryFrom($this->selectedSort);

    return Product::searchByCategory($this->category)
        ->join('product_size', 'products.id', '=',
        'product_size.product_id')
        ->select('products.*', DB::raw('SUM(product_size.count) as
        total_count'))
        ->groupBy('products.id')
        ->orderByRaw("(CASE WHEN total_count > 0 THEN 1 ELSE 0 END)
        DESC")
        ->when($this->selectedSizes, function ($builder, $value) {
            $builder->filterBySizes($value);
        })
}
```

```

        ->when($this->minPrice, function ($builder, $value) {
            $builder->where('price', '>=', Money::of($value, 'UAH',
roundingMode: RoundingMode::DOWN)->getMinorAmount()->toInt());
        })
        ->when($this->maxPrice, function ($builder, $value) {
            $builder->where('price', '<=', Money::of($value, 'UAH',
roundingMode: RoundingMode::DOWN)->getMinorAmount()->toInt());
        })
        ->when($selectedSort, function ($builder, $value) {
            $builder->sort($value);
        })
        ->with('mainPhoto')
        ->paginate(12);
    }

```

кінець лістингу 3.2

Таким підходом була реалізована вся клієнтська частина. На головній сторінці відображаються популярні товари, які адміністратор сам призначає. Тобто завдяки Livewire і Tailwind легко створювався дизайн, а бекенд логіка, із запитам до бази даних писалась в Livewire компонентах. Також створено модальне вікно в якому виводяться товари, які клієнт додавив у кошик. Livewire компонент проводить обрахунки, скільки вибрано товару по якій ціні, і знизу виводить суму до сплати. Там же добавлено кнопку, яка просто пересилає користувача на сторінку оплати.

Tailwind CSS – це утилітарний CSS-фреймворк, який дозволяє створювати дизайн, використовуючи класи в HTML коді, без необхідності створювати свої власні. Наприклад існують такі класи: grid, flex, gap, mt-3, p-4, rounded-xl. Можна стилізувати колір кнопок, фон або сам текст, існують такі команди: bg-blue-500, text-white. А адаптивність реалізується завдяки префіксам до класів наприклад «sm:mt-2» означає що буде використаний клас, який знаходиться зразу після двокрапки, коли ширина екрану буде більше 640 пікселів.

На сторінці оплати користувач може вибрати метод доставки: Нова Пошта чи самовивіз. Якщо Нова Пошта, то користувач може вибрати ті населені пункти і відділення, які отримуються з Нова Пошта API. Laravel, раз в тиждень робить запити і оновлює актуальні міста і відділення. Також є можливість вибрати спосіб оплати: Монобанк, при отриманні або на інший рахунок. Якщо клієнт

використовує Монобанк, тоді інтернет-магазин взаємодіє з API від Монобанку для створення платежу, потім інтернет-магазин перенаправляє користувача на створений платіж. На рисунку 3.5 сторінка оформлення замовлення.

The screenshot displays the 'DRESSIETY' website's order form. The header includes the brand name, a search bar, and navigation links for 'Адмінка' and 'Admin'. The main content is divided into two panels. The left panel, titled 'Форма замовлення', contains fields for 'Ім'я' (Name), 'Прізвище' (Surname), 'Телефон' (Phone), and 'Метод доставки' (Delivery Method) with radio buttons for 'Нова Пошта' (selected) and 'Самовібіз'. It also includes a dropdown for 'Населений пункт' (Settlement) and a section for 'Метод оплати' (Payment Method) with radio buttons for 'Монобанк' (selected), 'При отриманні' (On receipt), and 'На рахунок'. A 'Додаткова інформація' (Additional information) field is at the bottom. The right panel, titled 'Ваше замовлення', shows the order details: 'Футболка унісекс SURF, жовта' (Unisex SURF T-shirt, yellow), 'Артикул: 1002', 'Розмір: Універсальний', and a price of '750.00 грн'. The total amount to be paid is '750.00 грн'.

Рисунок 3.5 – Сторінка оформлення замовлення

Були створені сторінки профілю користувача, налаштування, щоб користувач міг змінити при бажанні. Також зручна сторінка «Мої покупки», там користувач може переглядати свої попередні замовлення, і повторити при бажанні.

Інтерфейс адміністратора – це окрема частина інтернет-магазину, в основному вона призначена, щоб модератор керував товарами, замовленнями, користувачами. Його реалізація була побудована з врахуванням принципів зручності і інтуїтивності, також присутня адаптивність, яка дає змогу використовувати на різних девайсах. Серед технологій для реалізації були використані також Tailwind CSS та DaisyUI.

DaisyUI – це бібліотека компонентів, яка є створеною на базі Tailwind CSS класів. Вона в основному забезпечує готові компоненти інтерфейсу, що дуже сильно пришвидшує розробку. Дуже часто в панелі адміністратора використовуються компоненти такі як: table, modal, btn. Завдяки цьому, час на

реалізації красивого дизайну значно зменшується, і завжди є можливість редагування.

На рисунку 3.6 зображено зовнішній вигляд панелі адміністратора, а саме сторінки замовлень.

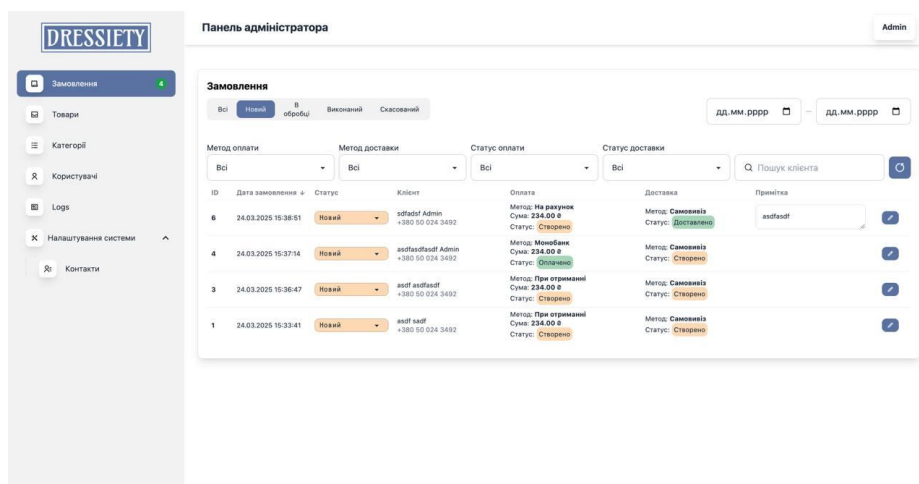


Рисунок 3.6 – Зовнішній вигляд сторінки замовлень в панелі адміністратора

Завдяки цим технологіям, інтернет-магазин повністю адаптивний, всі таблиці, меню та форми правильно відображаються на телефонах і планшетах.

На рисунку 3.7 зображено зовнішній вигляд сторінки, використовуючи інструмент Google DevTools для перевірки адаптивності.

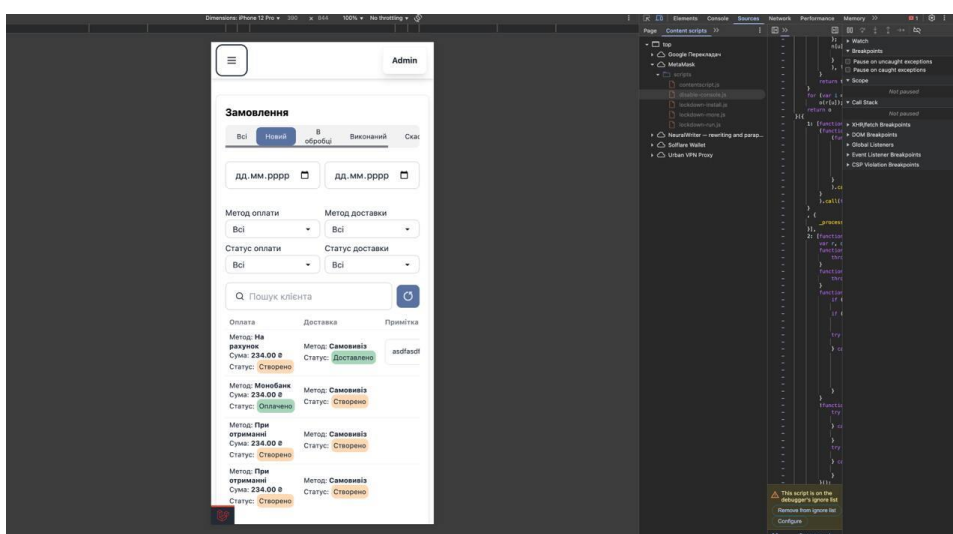


Рисунок 3.7 – Використання Google DevTools для перевірки адаптивності

В лістингу 3.3 зображено частину коду реалізації сторінки замовлень з використанням класів Tailwind CSS і компонентів DaisyUI.

Лістинг 3.3 – Код з використанням Tailwind CSS та DaisyUI

```
<div class="mt-2 w-full xl:max-w-[320px] flex items-center gap-2">
  <label class="input input-bordered flex items-center gap-2 w-full">
    <i class="ri-search-line"></i>
    <input wire:model.live.debounce.200ms="form.searchText"
class="w-full" type="text" class="grow" placeholder="Пошук клієнта" />
  </label>

  <button wire:click="resetFilter" class="btn btn-sm btn-primary px-1
min-w-[40px] min-h-[40px]" aria-label="Reset filter">
    <i class="ri-reset-left-line ri-lg"></i>
  </button>
</div>
```

кінець лістингу 3.3

У процесі розробки інтернет-магазину важливим етапом є розгортання його на продакшн (англ. production), вибрана хостингова (англ. hosting) платформа має забезпечити стабільну, безпечну і масштабовану роботу сайту. На ринку існує багато рішень, серед яких найбільш популярними є Microsoft Azure, Amazon Web Services, Google Cloud Platform, DigitalOcean, а також традиційні віртуальні хостинги на зразок Hostinger, Bluehost.

Google Cloud Platform поєднує в собі гнучкість і потужність корпоративного рівня з інтерфейсом, що є досить зручним навіть для індивідуальних розробників. GCP пропонує інструменти, необхідні для розгортання Laravel-додатків, зокрема сервіси Compute Engine для створення VPS-серверів, Cloud SQL для управління базами даних MySQL, а також Cloud Storage для зберігання статичних файлів. Однією з ключових переваг GCP є доступ до безкоштовного рівня, який дозволяє розгорнути невеликий інтернет-магазин без додаткових витрат під час розробки та тестування.

На рисунку 3.8 зображено користувацький інтерфейс Google Cloud Platform, де відображається інформацію по серверу на якому розгортається проєкт.

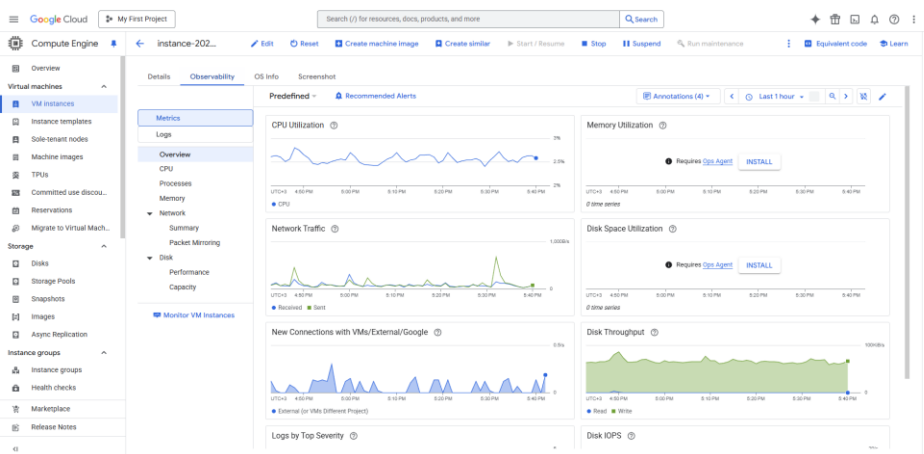


Рисунок 3.8 – Користувацький інтерфейс Google Cloud Platform [15]

Отже, даний проєкт не потребує великої кількості пристроїв, чи великої продуктивності. Був створений сервер з 2 ядрами в процесорі, 8 ГБ оперативної пам'яті і 20 ГБ пам'яті на диску.

Я клонував репозиторій свого проєкту у домашній каталог віртуального сервера, налаштував .env файл, вказавши параметри підключення до бази даних. Також я створив обліковий запис користувача та надав доступ до бази зі сторони віртуального сервера, налаштувавши відповідні правила фаєрволу.

Також я налаштував доменне ім'я через NIC.UA [16], де було орендоване доменне ім'я, прив'язавши його до IP-адреси мого сервера. Для забезпечення захищеного з'єднання я встановив SSL-сертифікат за допомогою «Let's Encrypt» через утиліту Certbot.

Оскільки мій застосунок інтегрує Telegram-бота для взаємодії з адміністрацією магазину, я окремо налаштував вебхук (англ. webhook), вказавши адресу мого сайту з відповідним маршрутом, через який бот отримує повідомлення. GCP дозволяє відкривати потрібні порти, тому я без проблем активував необхідні налаштування мережі.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Варто розказати про середовище розробки на якому розроблявся проєкт. Використовувався персональний комп'ютер на базі операційної системи

Windows. Для кращої продуктивності системи встановлено 16 ГБ оперативної пам'яті. Для розгортання MySQL використовувався локальний сервер XAMPP, який слухає на базовому порту 3306 запити, які будуть надсилатись до MySQL.

На рисунку 3.9 зображено панель керування локальним сервером XAMPP.

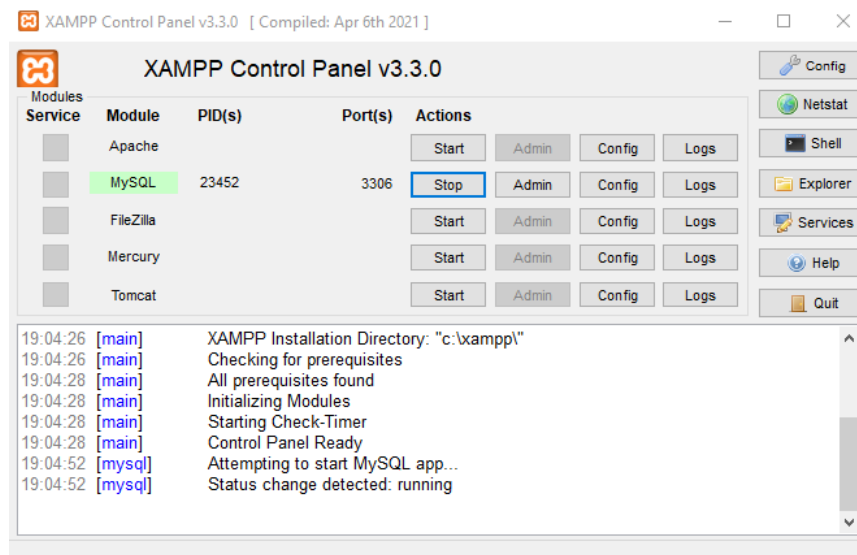


Рисунок 3.9 – Вигляд панелі керування локального сервера XAMPP [17]

Для базового функціонування системи в умовах невеликого навантаження (наприклад, тестування або запуск MVP-версії проєкту) достатньо таких параметрів:

- процесор (CPU) з 1 або 2 віртуальними ядрами;
- оперативна пам'ять має бути мінімум 4 ГБ;
- місце на диску має бути від 10 до 20 ГБ (залежно від кількості медіа);
- операційна система Ubuntu 20.04 або новіша.

Такі характеристики дозволяють підтримувати коректну роботу Laravel-застосунку, обробку запитів від Telegram-бота, базову взаємодію з базою даних і забезпечення простого користувацького інтерфейсу.

У процесі розробки важливе місце займає тестування та налагодження, адже саме ці етапи дозволяють переконатись у правильності роботи застосунку, виявити помилки та оптимізувати логіку роботи окремих функцій. Для мого проєкту, реалізованого з використанням Laravel, Livewire, MySQL та Tailwind

CSS, я застосував різні підходи та інструменти тестування – як вручну, так і за допомогою вбудованих або сторонніх засобів.

На початкових етапах найбільш часто використовуваним методом перевірки правильності роботи коду була функція «dd()» (dump and die), яка дозволяє вивести значення змінних у будь-якій точці коду та миттєво зупинити подальше виконання скрипта. Це дуже зручно, коли потрібно швидко відслідкувати, що саме передається в контролер, у компонент Livewire чи модель, або які значення зчитуються з бази даних. Laravel також надає альтернативу у вигляді функції «dump()», яка не зупиняє виконання програми, що корисно при послідовному виведенні кількох значень.

Окрім цього, у проєкті активно використовувався пакет Laravel «Debugbar» є одним з найпопулярніших інструментів налагодження у світі Laravel.

На рисунку 3.10 зображено інтерфейс пакета, який використовується для тестування.

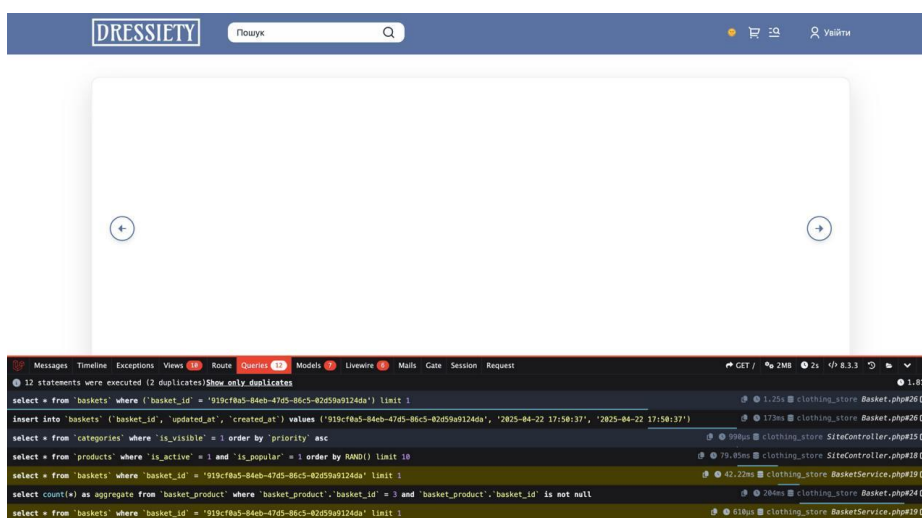


Рисунок 3.10 – Інтерфейс інструменту налагодження «Debugbar»

Цей пакет додає панель знизу сторінки, в якій відображається велика кількість корисної інформації: список виконаних SQL-запитів, маршрути, поточний контролер та метод, дані сесії, змінні середовища, повідомлення логів, а також інформація про використання пам'яті, час виконання запиту та інше. Це особливо корисно, коли потрібно оптимізувати продуктивність, наприклад,

побачити надмірну кількість запитів до бази даних або перевірити, чи всі змінні формуються правильно.

Для складніших сценаріїв Laravel надає потужну систему логування на базі Monolog. Усі події можна логувати вручну через метод «`Log::info()`», «`Log::error()`» або інші. Це дозволяє відслідковувати, наприклад, процес авторизації, отримання повідомлень від Telegram-бота чи обробку виключень (exceptions). Логи зберігаються у файлі `storage/logs/laravel.log`, і можуть бути особливо корисними у продакшн-оточенні, де немає доступу до `dd()` чи `Debugbar`.

У рамках налагодження Telegram-бота також були застосовані методи логування повідомлень, що приходять, і результатів, які віддає бот у відповідь. Крім того, Laravel дозволяє налаштовувати окремі канали логування, наприклад, зберігати інформацію в окремих файлах відокремлених по днях.

Щодо автоматичного тестування, Laravel підтримує PHPUnit – інструмент для модульного тестування. Хоч у рамках цього проєкту основна перевірка проводилась вручну, середовище повністю готове до написання тестів і для моделей, і для HTTP-запитів, і навіть для Livewire-компонентів. Для останніх існує бібліотека Livewire Testing, яка дозволяє симулювати взаємодію з компонентами, перевіряти зміни стану, рендеринг та навіть виконання валідації. На етапі фінального тестування я вручну перевіряв усі основні сценарії: реєстрацію, вхід у систему, роботу фільтрів і сортування на сторінці каталогу, додавання, редагування та видалення товарів з панелі адміністратора, роботу Telegram-бота, редагування замовлень. Було протестовано адаптивність інтерфейсу на різних пристроях, коректність відображення повідомлень та поведінку системи у разі помилкових дій користувача, наприклад, неправильного введення даних.

Отже, Laravel надає потужний інструментарій для налагодження: як під час розробки, так і на етапі фінального тестування. Застосування `dd()`, логування, `Debugbar`, ручного тестування та перевірки бізнес-логіки у компонентах Livewire дозволило досягти стабільної та прогнозованої роботи системи. Усі критичні

помилки були виявлені й усунуті до моменту розгортання, що забезпечило високий рівень надійності проєкту.

3.3 Розробка бази даних

Для реалізації інтернет-магазину потрібно зберігати дані, а значить створити таблиці так як використовується реляційна база даних MySQL. Для створення таблиць, введення змін до них або відкати використовуються міграції. На основі діаграми класів, рисунок 2.1, були написані міграції в Laravel для створення таблиць. Одна з міграцій зображена на лістингу 3.4, коли вона визивається, то створюється таблиця товарів з колонками відповідними до діаграми класів і містить додаткові.

Лістинг 3.4 – Міграція для створення таблиці products

```
public function up(): void
{
    Schema::create('products', function (Blueprint $table) {
        $table->id();
        $table->foreignId('category_id')->nullable()
->constrained('categories', 'id')->onDelete('cascade');
        $table->string('name');
        $table->string('slug');
        $table->string('article');
        $table->text('description')->nullable();
        $table->text('short_description')->nullable();
        $table->integer('count');
        $table->bigInteger('price');
        $table->bigInteger('old_price')->nullable();
        $table->boolean('is_discount')->default(false);
        $table->boolean('is_active')->default(true);
        $table->boolean('is_popular')->default(false);
        $table->timestamps();
    });
}
```

кінець лістингу 3.4

Після створення міграцій, і виклику команди, яка запускає їх (php artisan migrate), була створена база даних на основі даних в файлі .env і таблиці на основі

міграцій. Також на додачу можна створити сидери (англ. seeder), які заповняють тестовими даними таблиці, але доволі легко і швидко можна реалізувати керування моделями в Laravel, і заповнити дані вже на сайті.

3.4 Створення Telegram-бота

Для реалізації функціоналу Telegram-бота було досліджено офіційну документацію Telegram Bot API [18]. Також була використана популярна бібліотека в PHP «Telegram Bot SDK for PHP» [19], яка дозволяє зручно працювати з усіма можливостями API Telegram у рамках Laravel додатка. Завдяки цій бібліотеці вдалось організувати отримання оновлень від Telegram, обробку повідомлень та відповіді користувачам.

На боці сервера було створено окремий маршрут на котрий будуть приходити входні вебхуки (англ. webhook) від Telegram. Цей маршрут є захищеним, так як містить у своєму URI унікальний токен бота, що унеможливорює несанкціонований доступ. При отриманні запиту Telegram, викликається контролер TelegramController, який відповідає за обробку оновлення. У контролері знаходиться базова логіка розрізнення типів повідомлення, якщо оновлення є текстовим повідомленням, воно передається на обробку у MessageHandler, якщо ж мова йде про callback-запит (наприклад, при натисканні на кнопку), тоді використовується CallbackHandler.

Головна логіка обробки текстових повідомлень реалізована у класі MessageHandler. У цьому класі описано, як бот має реагувати на команду «/orders», що є тригером для виводу списку останніх замовлень. Система виконує перевірку, чи є користувач адміністратором, що налаштовано через конфігураційний файл. У разі успішної авторизації бот формує текстове повідомлення з інформацією про замовлення. Разом із текстом надсилається клавіатура з кнопками для редагування кожного замовлення та кнопками для переходу між сторінками, тобто реалізована повноцінна пагінація в межах Telegram.

На рисунку 3.11 зображено вивід Telegram-бота у відповідь на команду «/orders».

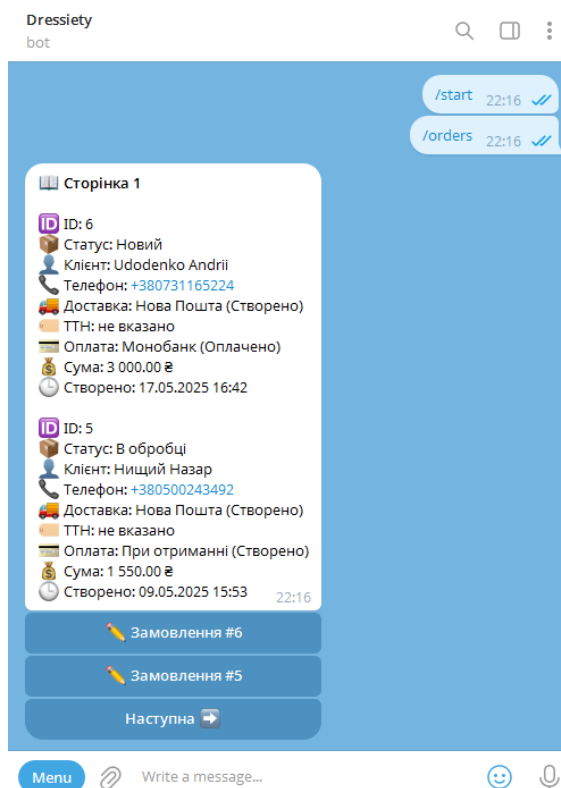


Рисунок 3.11 – Вивід Telegram-бота у відповідь на команду

У лістингу 3.5 зображено код, який створює пагінацію замовлень у повідомленні від Telegram-бота, завдяки чому адміністратор зможе переглядати інші замовлення не переглядаючи всі одночасно.

Лістинг 3.5 – Код формування пагінації у відповіді від Telegram-бота

```
public static function buildPaginationKeyboard($orders, int
$currentPage): Keyboard
{
    $keyboard = Keyboard::make()->inline();

    foreach ($orders as $order) {
        $keyboard->row([
            Keyboard::inlineButton([
                'text' => "✎ Замовлення #{ $order->id }",
                'callback_data' => "edit_order_{ $order-
>id }_{ $currentPage }"
            ])
        ])
    }
}
```

```

    });
}

$paginationRow = [];
if ($currentPage > 1) {
    $paginationRow[] = Keyboard::inlineButton([
        'text' => '⬅️ Попередня',
        'callback_data' => 'orders_page_' . ($currentPage - 1)
    ]);
}

if ($orders->hasMorePages()) {
    $paginationRow[] = Keyboard::inlineButton([
        'text' => 'Наступна ➡️',
        'callback_data' => 'orders_page_' . ($currentPage + 1)
    ]);
}

if (!empty($paginationRow)) {
    $keyboard->row([...$paginationRow]);
}

return $keyboard;
}

```

кінець лістингу 3.5

На рисунку 3.12 зображено вивід деталей для редагування замовлення, знизу під повідомленням знаходяться кнопки для взаємодії.

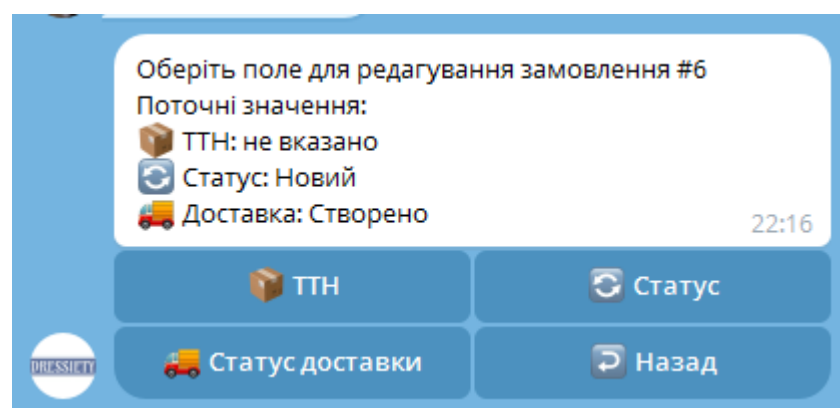


Рисунок 3.12 – Деталі для редагування замовлення в Telegram-ботові

На рисунку 3.13 зображено перелік статусів серед яких можна вибрати, якщо адміністратор хоче змінити статус замовлення.

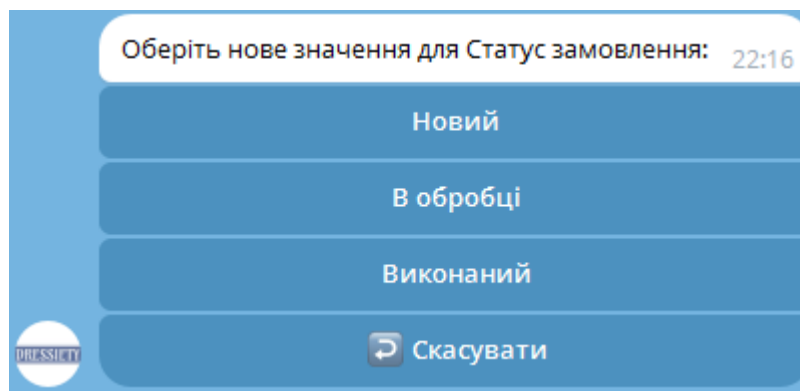


Рисунок 3.13 – Перелік статусів для редагування статусу замовлення

3.5 Захист інформаційно-комп'ютерної системи

Кожен елемент системи – від реєстрації користувача до шифрування даних на зберіганні – розглядався з погляду інформаційної безпеки, щоб гарантувати конфіденційність, сукупність та доступність даних навіть за інтенсивних навантажень або під час спланованих кібер-атак.

Усім користувачам надається можливість входити до свого облікового запису або через електронну адресу, або через мобільний номер телефону, що значно підвищує зручність і гнучкість автентифікації. При введенні електронної пошти чи телефону разом із паролем відбувається перевірка відповідності введених даних із хешем у базі даних за допомогою алгоритму bcrypt. «Сіль» довжиною 128 біт генерується випадково для кожного пароля, що унеможливорює попередню атаку «rainbow-tables».

Також доступ до панелі адміністратора є тільки для користувачів із ролям відповідним. Middleware при кожному запиті буде перевіряти, і якщо немає відповідної ролі, користувач не зможе отримати доступ і відкрити сторінку.

Laravel із коробки надає широкий набір інструментів і механізмів для забезпечення безпеки веб-додатків, що дозволяє розробникам зосередитися на бізнес-логіці, не витрачаючи зайвий час на базову захисну інфраструктуру. По-перше, фреймворк автоматично генерує для всіх форм на сайті захист від міжсайтової підробки запитів (CSRF). Це реалізовано через вставку у форму

прихованого інпута з токеном, який перевіряється при обробці запиту, і у випадку невідповідності Laravel одразу відхиляє запит із помилкою 419.

По-друге, Blade (власний шаблонізатор Laravel) за замовчуванням екранує всі виведені змінні, що мінімізує ризик впровадження шкідливого HTML чи скриптів (XSS). Якщо потрібно вивести «сирий» HTML, розробнику доводиться спеціально викликати методи на кшталт «{!! \$variable !!}», тим самим явно беручи відповідальність за безпеку.

ORM Eloquent та побудова запитів через Query Builder використовують PDO з підготовленими виразами, тому у стандартному коді практично неможливо випадково піддати додаток ін'єкціям SQL. Навіть у разі динамічної побудови запиту через функцію whereRaw розробник повинен явно вказати, що бере на себе відповідальність за безпеку переданих параметрів.

Для шифрування даних Laravel пропонує взаємодію з компонентом OpenSSL через фасад Crypt. Він забезпечує AES-256-CBC шифрування з додаванням HMAC для автентичності даних. Достатньо викликати Crypt::encrypt(\$value) та Crypt::decrypt(\$payload), щоб безпечно зберігати конфіденційну інформацію.

Механізми аутентифікації та управління сесіями у Laravel побудовані на модулі Guards і Providers. За замовчуванням доступні сесійний та токен-базований Guard, а для зберігання сесій – різні драйвери (файли, база даних, Redis). Сесійні куки автоматично шифруються та підписуються вмістом ключів із конфігурації app.key, а рівень захисту можна посилити через опції secure, http_only і same_site у файлі config/session.php.

Крім того, у Laravel реалізовано механізм обмеження кількості запитів (Rate Limiting) через фасад RateLimiter. Він дозволяє задавати політики для окремих маршрутів або груп маршрутів, наприклад, не більше 60 запитів на хвилину з однієї IP-адреси, що допомагає захиститися від простих форм DoS-атак і brute-force спроб.

Окремо варто згадати підтримку безпечного хешування паролів через фасад Hash. За замовчуванням використовується bcrypt із фінтюнером cost із конфігурації, але можна обирати Argon2 для ще більшого рівня захисту.

Нарешті, Laravel регулярно оновлює та підтримує свій код, випускаючи патчі безпеки й попереджаючи про вразливості. Завдяки чіткій структурі та активній спільноті виправлення інтегруються просто й оперативно через Composer.

Усі HTTP-запити обгорнуті протоколом TLS 1.3, сертифікати для якого керуються безпосередньо GCP або автоматично оновлюються через «Let's Encrypt». Під час TLS-handshake відбувається обмін еліптичними ключами, із яких генерується спільний секрет і похідні ключі для шифрування та автентифікації повідомлень. Таким чином, захищено з'єднання між клієнтом і сервером.

Усе це в сукупності дозволяє початково захищати додаток від найпоширеніших загроз, залишаючи за розробником свободу додавати кастомні рішення там, де це необхідно.

Висновки до розділу 3

В цьому розділі розглянуто процес створення інтернет-магазину чоловічого одягу з використанням сучасних технологій та інструментів розробки. Основою проєкту став фреймворк Laravel, що надав структуровану архітектуру за моделлю MVC, а також спростив маршрутизацію, перевірку даних, взаємодію з базою даних та інші рутинні операції. Застосування реляційної СУБД MySQL дозволило ефективно структурувати та зберігати дані, а міграції та сидери забезпечили гнучкість у керуванні структурою бази даних.

Розроблена система авторизації з опцією входу через електронну пошту чи номер телефону покращила комфорт користування, а middleware для адміністраторської панелі гарантував безпеку від нелегального проникнення.

Додатково передбачена функція відновлення пароля через електронну пошту або телефон.

Інтеграція з Telegram-ботом виглядає актуальною та перспективною концепцією. Вона дала змогу адміністраторам контролювати замовлення за допомогою інтуїтивного інтерфейсу месенджера, що суттєво пришвидшило обробку запитів.

Для фронтенду застосовано Livewire, Tailwind CSS та DaisyUI, що полегшило розробку адаптивного та зручного інтерфейсу, як для користувачів, так і для адміністраторів.

Розгортання проєкту на Google Cloud Platform гарантує стабільність та безпеку завдяки інструментам Compute Engine та автоматичному резервному копіюванню. Встановлення SSL-сертифікату та доменного імені додатково збільшило надійність системи.

Реалізований інтернет-магазин демонструє успішне впровадження сучасних технологічних інструментів для створення повноцінного комерційного продукту. Він також враховує слабкі сторони наявних рішень та містить перспективні нововведення, серед яких реалізація Telegram-бота виглядає особливо цікавою.

Проєкт повністю готовий до експлуатації, має потенціал для подальшого розвитку та масштабування. Це може бути досягнуто шляхом інтеграції нових сервісів платіжних систем, служб доставки або шляхом розширення функціоналу Telegram-бота, наприклад, додавши можливість безпосередньо оформляти замовлення через нього. Для підвищення безпеки проєкту можливо використовувати додаткові інструменти від Google Cloud Platform.

ВИСНОВКИ

У процесі виконання дипломної роботи було створено актуальний інтернет-магазин з інтегрованим Telegram-ботом, що відповідає актуальним стандартам електронної комерції.

Дослідження охоплювало детальний аналіз ринкових трендів, зокрема стрімке збільшення популярності онлайн-покупок, необхідність автоматизації робочих процесів та інтеграцію месенджерів для підвищення ефективності обробки замовлень.

За результатами проведеного дослідження були сформовані вимоги до системи, які включають інтуїтивно зрозумілий інтерфейс, адаптивність під різні пристрої, забезпечення безпеки даних та автоматизацію основних операцій.

Архітектура системи спроектована за допомогою UML-діаграм класів і варіантів використання, що сприяло конкретному окресленню взаємодії складових. Бекенд було вирішено розробити з використанням фреймворку Laravel через його гнучкість, структуру MVC та інструменти для швидкої розробки. База даних на основі MySQL гарантувала надійне збереження та обробку транзакцій. Фронтенд було реалізовано з використанням Livewire та Tailwind CSS, що дало змогу створити динамічні інтерфейси з мінімальним залученням JavaScript. Інтеграція Telegram-бота через Telegram Bot API автоматизувала керування замовленнями.

Практична частина роботи передбачала розробку повноцінного інтернет-магазину. Він мав каталог товарів, систему кошика, автентифікацію за допомогою електронної пошти або номера телефону, а також адаптивний дизайн і підтримку темної та світлої тем. Система була розгорнута на платформі Google Cloud Platform, а саме на Compute Engine, що гарантувало стабільність, масштабованість і високий рівень захисту від зовнішніх атак.

Особливу увагу було приділено тестуванню та відлагодженню системи. В процесі розробки застосовувались інструменти Laravel Debugger, а також вбудовані функції «dd()» і «dump()».

База даних MySQL проєктувалась на основі UML-діаграми класів, що полегшило та прискорило її створення.

Інтеграція Telegram-бота, реалізована через PHP-бібліотеку з використанням вебхуків, надає адміністраторам можливість обробляти замовлення безпосередньо у месенджері, що дозволяє значно прискорити процес обробки.

Забезпечення безпеки інформаційної системи через Laravel та його функціонал дозволяє безперешкодно здійснювати онлайн операції, коректно автентифікувати користувачів, та звести ризик несанкціонованого доступу до мінімуму. Використовувались засоби, зокрема, хешування, шифрування, middleware з перевіркою прав та інші рішення.

Майбутнє проєкту включає впровадження нових платіжних методів та збагачення можливостей бота. Виконана робота засвідчила результативність вибраного шляху: синергія передових технологій гарантувала високу ефективність, простоту у використанні для клієнтів та адміністраторів, а також потенціал для подальшого вдосконалення. Створений інтернет-магазин є конкурентоспроможним рішенням, яке відповідає вимогам цифрової епохи та сприяє розвитку електронної комерції в Україні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Крилов Д. В. Розвиток електронної комерції в Україні в сучасних умовах. 2024. № 12. URL: <https://doi.org/10.54929/2786-5738-2024-12-03-02> (дата звернення: 13.01.2025).
2. Ларіна Я., Нагорна О. Current state and development trends of e-trade in Ukraine. 2024. Т. 326, № 1. С. 373-380. URL: <https://doi.org/10.31891/2307-5740-2024-326-59> (date of access: 20.01.2025).
3. Balyk U., Stevchak Y. E-commerce market: new technologies and development prospects. 2024. URL: <https://doi.org/10.23939/smeu2024.01.083> (date of access: 23.01.2025).
4. Lesko N., Kira S. E-commerce market in Ukraine: the role of european and international e-commerce organizations. 2024. Т. 11, № 41. С. 153-158. URL: <https://doi.org/10.23939/law2024.41.153> (date of access: 27.01.2025).
5. Style Print. URL: <https://styleprint.ua/> (дата звернення: 15.02.2025).
6. Konfiskat. URL: <https://konfiskat.ua/> (дата звернення: 15.02.2025).
7. Understanding UML, REST API, and Database Expressions of the Ed-Fi Data Model – Ed-Fi Alliance. URL: <https://docs.ed-fi.org/reference/data-exchange/technical-articles/understanding-uml-rest-api-and-database-expressions-of-the-ed-fi-data-model> (date of access: 28.02.2025).
8. Редакція. Темна тема чи світла: що краще? Висновки на основі наукових публікацій. URL: <https://ux.pub/editorial/tiemna-tiema-chi-svitla-shcho-krashchie-visnovki-na-osnovi-naukovikh-publikatsii-23e4> (дата звернення: 28.03.2025).
9. Tenzin S. PHP Framework for Web Application Development. 2022. Т. 9, № 2. URL: <https://doi.org/10.17148/iarjset.2022.9218> (date of access: 28.03.2025).
10. Installation – Laravel 12.x – The PHP Framework For Web Artisans. URL: <https://laravel.com/docs/12.x> (date of access: 28.03.2025).

11. Santoso G. B., Sinaga T. M., Zuhdi A. MVC Implementation In Laravel Framework For Development Web-Based E-Commerce Applications. 2021. T. 1, № 1. URL: <https://doi.org/10.25105/itm.v1i1.7867> (date of access: 28.03.2025).

12. Database Management Education in MYSQL / J. Wahyudi and etc. URL: <https://doi.org/10.33487/edumaspul.v6i2.4570> (date of access: 28.03.2025).

13. Onyebueke E. An Overview Of How Livewire Works. URL: <https://medium.com/@developer.olly/an-overview-of-how-livewire-works-85395746d10a> (date of access: 15.03.2025).

14. Tailwind CSS – Rapidly build modern websites without ever leaving your HTML. URL: <https://tailwindcss.com/> (date of access: 10.04.2025).

15. Cloud Computing Services – Google Cloud. URL: <https://cloud.google.com/> (date of access: 19.04.2025).

16. Domain registrar NIC.UA. URL: <https://nic.ua/> (дата звернення: 18.04.2025).

17. XAMPP Installers and Downloads for Apache Friends. URL: <https://www.apachefriends.org/> (date of access: 14.03.2025).

18. Telegram Bot API. URL: <https://core.telegram.org/bots/api> (date of access: 15.04.2025).

19. The ultimate toolkit for building Telegram chatbots with Laravel support. Telegram Bot SDK for PHP. URL: <https://telegram-bot-sdk.com/> (date of access: 16.04.2025).

ДОДАТКИ

Додаток А – Підтвердження апробації результатів кваліфікаційної роботи

Міністерство освіти і науки України
 Волинська обласна рада
 Луцька міська рада
 Луцький національний технічний університет
 Національний технічний університет України «Київський політехнічний
 інститут імені Ігоря Сікорського»
 Національний університет «Львівська політехніка»
 Військовий інститут телекомунікацій та інформатизації
 імені Героїв Крут (м. Київ)
 Тернопільський національний педагогічний університет імені В. Гнатюка
 Рівненський державний гуманітарний університет
 Навчально-науковий інститут «Українська інженерно-педагогічна академія»
 Харківського національного університету ім. В.Н. Каразіна
 Люблінська політехніка (Польща)
 Фрайберзька гірнича академія (Німеччина)
 Гронінгенський університет (Нідерланди)
 Кавказький університет (Грузія)
 Політехнічний університет Браганси (Португалія)



ТЕЗИ ДОПОВІДЕЙ

**X Міжнародної науково-практичної конференції з
 проблем вищої освіти і науки «Інформаційні технології
 в освіті, науці і виробництві (ІТОНВ-2025)**

23-24 травня 2025 р.

Луцьк 2025

Кондіус І.С. Антоненко Р.В.	Автоматизація бізнес-процесів компанії з регенерації картриджів	190
Кондіус І.С. Кондіус К.Ю.	Розробка системи управління персоналом	192
Кондіус І.С. . Кубай Д.І.	Розробка системи для автоматизації процесу обліку кадрів великого підприємства	195
Кондіус І.С. Терещук М.Р.	Розробка системи автоматизації Веб-сервісу комерційного банку	198
Кухарчук М.О. Сачук В.О.	Розробка мобільного додатку для анонімних чат знайомств на основі Kotlin та хмарних технологій	201
Ліщина Н.М. Малевиц Ю.В.	Удосконалення процесу вивчення англійської мови за допомогою вебзастосунку з базою даних та особистим кабінетом користувача	205
Ліщина Н.М. Озірський Р.М.	Автоматизація обробки заявок у логістичній компанії за допомогою вебзастосунку з інтерактивною картою та базою даних	208
Мельник М.В. Самчук Л.М.	Вибір інструментів для розробки онлайн таблиці лідерів в грі «StarCraft:Remastered»	212
Нищий Н.І. Самчук Л.М.	Архітектурне моделювання інтернет-магазину через діаграму класів UML	216
Ніколайчук Є.Р. Ніколайчук С.Р. Ліщина Н.М.	Мобільний застосунок як інструмент цифрової трансформації дошкільної освіти	220
Пархоменко В.Г.	Знаходження додатного аксіально-симетричного розв'язку задачі Діріхле для еліптичного рівняння з антимонотонною нелінійністю методом двобічних наближень	223

УДК 004.41

АРХІТЕКТУРНЕ МОДЕЛЮВАННЯ ІНТЕРНЕТ-МАГАЗИНУ ЧЕРЕЗ ДІАГРАМУ КЛАСІВ UML

Нищий Назар Ігорович

Луцький національний технічний університет, студент, бакалавр інженерії програмного забезпечення, nyshchyi.n3007@lntu.edu.ua

Самчук Людмила Михайлівна

Луцький національний технічний університет, кандидат технічних наук, доцент кафедри прикладної механіки та мехатроніки, Samchuk204@gmail.com

ARCHITECTURAL MODELING OF AN ONLINE STORE THROUGH A UML CLASS DIAGRAM

Nyshchyi Nazar Ihorovych

Lutsk National Technical University, Student, Bachelor of Software Engineering, nyshchyi.n3007@lntu.edu.ua

Samchuk Liudmyla Mykhailivna

Lutsk National Technical University, PhD in Technical Sciences, Associate Professor, Department of Applied Mechanics and Mechatronics
Samchuk204@gmail.com

The rapid growth of e-commerce has led to an increasing demand for reliable and scalable online store systems. To ensure efficient system architecture, it is crucial to use modeling tools such as UML class diagrams. This paper presents the development of a class diagrams part of used on creating an online store. The use of UML modeling has proven effective in organizing the system structure and guiding fur the implementation and data base design.

Keywords: UML, class diagram, online store, system modeling, soft ware architecture, e-commerce, data base design

Інтернет-магазини є одними з найпоширеніших типів веб-застосунків у сучасному цифровому середовищі. Щодня мільйони користувачів взаємодіють із такими платформами для здійснення покупок, оформлення замовлень, оплати та отримання товарів. В умовах зростаючої конкуренції та високих вимог до якості онлайн-сервісів особливої важливості набуває правильне проєктування програмної архітектури інтернет-магазинів. Одним із ключових інструментів такого проєктування

є UML-діаграми, які дозволяють формалізувати структуру системи ще до написання коду.

Метою даної роботи було розробити UML-діаграму класів для інтернет-магазину. Основні завдання:

- визначити ключові сутності системи та їх взаємозв'язки;
- оптимізувати архітектуру для подальшої реалізації.

Розробка UML-діаграми класів почалася з аналізу предметної області. Було визначено основні сутності системи, такі як користувачі, товари, замовлення, кошики та категорії продуктів. Кожен клас отримав свої атрибути та методи, що дозволило чітко визначити їхню роль у системі. Наприклад, клас «Користувач» містив дані для автентифікації та управління профілем, а клас «Замовлення» відповідав за обробку транзакцій і доставку.

Важливим етапом стало проектування зв'язків між класами. Асоціації, агрегації та узагальнення допомогли відобразити реальні взаємодії між об'єктами. Так, зв'язок між «Кошиком» і «Товаром» був реалізований через проміжний клас, що дозволило враховувати кількість товарів у кошику. Ієрархія «Категорія» – «Підкатегорія» спростила навігацію по каталогу, а зв'язок «Замовлення» – «Користувач» забезпечив відстеження історії покупок.

Використання UML також дало змогу виявити потенційні проблеми на ранніх етапах. Наприклад, первісна структура не враховувала можливість знижок для груп товарів, що було виправлено до початку програмування. Крім того, діаграма стала зручним інструментом, оскільки наочно демонструвала логіку системи без необхідності аналізу коду.

Як зазначає Мартін Фаулер у своїй праці «UML Distilled», використання UML сприяє глибшому розумінню предметної області та значно полегшує спілкування між технічними і не технічними учасниками команди [1].

Розроблена діаграма, яка зображена на рисунку 1, охоплює ключові компоненти функціональності інтернет-магазину.

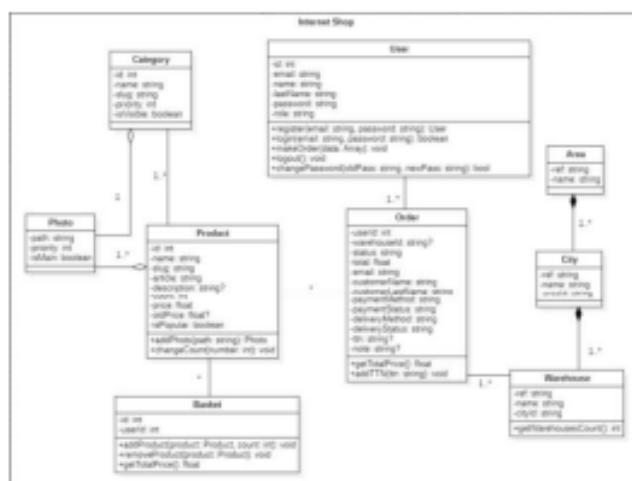


Рисунок 1 – UML-діаграма класів інтернет-магазину

На діаграмі можна виділити наступні групи класів [2]:

1. Користувач (User). Цей клас відповідає за дані зареєстрованих користувачів. Він містить атрибути для зберігання особистої інформації: електронної пошти, імені, прізвища, пароля та ролі користувача. Серед основних методів: реєстрація, авторизація, оформлення замовлення, зміна пароля тощо.

2. Категорія та продукт (Category, Product). Категорії використовуються для групування товарів. Кожна категорія має атрибути ієрархії (priority), ідентифікатор, назву, видимість тощо. Товари містять більш детальну інформацію: артикул, опис, ціну, популярність тощо. Продукт може бути пов'язаний із кількома фото.

3. Фото (Photo). Допоміжний клас, який пов'язаний із продуктом. Дозволяє прикріплювати кілька фото до одного товару, а також визначати головне фото.

4. Кошик (Basket). Кожен користувач має кошик, у якому може зберігати товари до моменту оформлення замовлення. У кошик додаються продукти з кількістю, можна їх видаляти або обраховувати загальну суму.

5. Замовлення (Order). Клас описує замовлення користувача. Містить атрибути, пов'язані зі статусом, методом

оплати та доставки, ПІБ замовника, email, номером ТТН, коментарями та загальною вартістю. Методи дозволяють обраховувати вартість замовлення та додавати ТТН.

6. Область, Місто, Склад (Area, City, Warehouse). Для доставки товарів реалізовано класи, що описують географічну структуру: область, міста та відділення (склади). Вони пов'язані між собою за ієрархічною моделлю.

На основі аналізу UML-діаграми класів інтернет-магазину можна зробити такі висновки. Система має чітку структуру з правильно визначеними зв'язками між основними сутностями: користувачами, товарами, замовленнями та системою доставки. Архітектура дозволяє ефективно керувати каталогом товарів через систему категорій, обробляти замовлення та контролювати процес доставки.

Головні переваги поточної архітектури включають розділення логіки між класами, що спрощує подальшу розробку та підтримку. Наприклад, клас Basketінкапсулює всю логіку роботи з кошиком, а Order відповідає за управління замовленнями. Система також передбачає різні ролі користувачів через атрибут role у класі User.

Для покращення системи рекомендується: усунути дублювання даних між User та Order шляхом використання посилань на об'єкт User; додати клас Discount для реалізації системи знижок та промокодів; розширити клас Product для підтримки різних варіантів товарів (кольори, розміри); додати клас Review для обробки відгуків про товари.

Список використаних джерел

1. Теорія та практика UML. – 2020. [Електронний ресурс]. Режим доступу до ресурсу: http://www.it-gost.ru/articles/view_articles/94
2. Чорний А. Бізнес-ідеї для інтернет-магазину з мінімальними вкладеннями [Електронний ресурс] / Андрій Чорний // Lamarbet. – 2019. – Режим доступу до ресурсу: <https://lemarbet.com/ua/instrument-idej/biznes-ideidlya-internet-magazinas-minimalnymi-vlozheniyami/>