

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Інтернет-магазин військових товарів»

Виконав студент групи ПЗ-21-2
КОВАЛЬ Дмитро Андрійович

Керівник роботи:
КОНЦИДАЙЛО Андрій Михайлович

Рецензент:
ЧИЖМОТРЯ Олена Геннадіївна

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна Вакалюк

«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **КОВАЛЬ Дмитро Андрійович**

Керівник роботи: **КОНЦИДАЙЛО Андрій Михайлович**

Тема роботи: **«Інтернет-магазин військових товарів»**,

затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**

Вихідні дані для роботи: **процеси та методи розробки веб-додатку для продажу військових товарів**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Концидайло А.М.	27.02.2025	27.02.2025
2	Концидайло А.М.	10.03.2025	10.03.2025
3	Концидайло А.М.	05.04.2025	27.04.2025

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з веб-додатку інтернет-магазину військових товарів та пояснювальної записки. Пояснювальна записка містить 101 сторінку, 21 таблицю, 27 рисунків, 9 додатків, 31 використане джерело.

Метою роботи є розробка веб-орієнтованого інтернет-магазину військових товарів, що забезпечує користувачам зручний інтерфейс для перегляду, пошуку, замовлення товарів та адміністрування контенту.

У процесі дослідження було проаналізовано особливості сучасних інтернет-магазинів, вивчено функціональність аналогічних рішень, визначено архітектуру системи, обрано технології реалізації. Основу програмного забезпечення складає веб-додаток, створений з використанням HTML, CSS, JavaScript та PHP, з базою даних MySQL.

Система дозволяє адміністратору керувати товарами, категоріями, замовленнями, а покупцеві – зручно обирати та замовляти продукцію.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ-МАГАЗИН, ВІЙСЬКОВІ ТОВАРИ, ВЕБ-ДОДАТОК, ЕЛЕКТРОННА КОМЕРЦІЯ, PHP.

					ІПЗ.КР.Б-121-25-ПЗ						
Зм.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Коваль Д.А.			Інтернет-магазин військових товарів			Літ.	Арк.	Акрушів	
Керівник		Конидайло А.М.								3	
								Житомирська політехніка, група ІПЗ-21-2			
Рецензент		Чижморя О.Г.									
Зав. каф.		Вакалюк Т.А.									

ABSTRACT

The bachelor`s qualification work consists of a web application for a military goods online store and an explanatory report.

The explanatory report includes 101 pages, 21 tables, 27 pictures, 9 appendices, and 31 references.

The aim of this work is to develop a web-oriented online store for military goods that provides users with a convenient interface for browsing, searching, ordering products, and managing content.

During the research process, the features of modern e-commerce platforms were analyzed, the functionality of similar solutions was studied, the system architecture was defined, and appropriate implementation technologies were selected. The core of the software solution is a web application developed using HTML, CSS, JavaScript and PHP, with a MySQL database.

The system allows the administrator to manage products, categories, and orders, while providing users with a user-friendly experience for selecting and purchasing goods.

KEYWORDS: ONLINE STORE, MILITARY GOODS, WEB APPLICATION, E-COMMERCE, PHP.

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT	4
ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ ВІЙСЬКОВИХ ТОВАРІВ	8
1.1. Постановка задачі.....	8
1.2. Аналіз аналогів програмного продукту	10
1.3. Вибір архітектури веб-орієнтованої системи	17
1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	20
Висновки до першого розділу.....	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ ВІЙСЬКОВИХ ТОВАРІВ.....	24
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи.....	24
2.2. Розробка бази даних системи.....	29
2.3. Проектування та реалізація алгоритмів роботи системи	40
2.4. Реалізація додатку	43
Висновки до другого розділу	46
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБ-ДОДАТКОМ ІНТЕРНЕТ-МАГАЗИНУ ВІЙСЬКОВИХ ТОВАРІВ	48
3.1. Порядок встановлення та налаштування параметрів системи	48
3.2. Структура інтерфейсу веб-додатку	50
3.3. Тестування функціоналу програмного комплексу	58
Висновки до третього розділу.....	60
ВИСНОВКИ	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТКИ.....	65

ВСТУП

У наш час, коли цифрові технології охоплюють майже всі аспекти нашого життя, електронна комерція стає дуже важливою у тому, як ми купуємо речі. Інтернет-магазини дають нам можливість швидко знайти багато товарів, легко купити або продати щось, отримати персоналізоване обслуговування та легко розширювати бізнес. Це особливо важливо для товарів, які користуються великим попитом і потребують частого оновлення, наприклад, військових товарів. В умовах збройної агресії, потреба в якісному військовому спорядженні значно зростає. Таке спорядження шукають не тільки військові, але й працівники силових структур, волонтери та звичайні громадяни. Тому важливо забезпечити легкий, зручний і безпечний спосіб купівлі цих товарів, що стає важливим не тільки з економічної, а й з соціальної точки зору, особливо в умовах війни.

Актуальність теми полягає у необхідності впровадження цифрових рішень для реалізації спеціалізованої продукції, яка потребує швидкого оновлення асортименту, контролю залишків товарів та зручного замовлення. Створення інтернет-магазину дозволяє автоматизувати процеси торгівлі, забезпечити зручний доступ користувачів до інформації про продукцію [28]. Актуальність теми також обумовлена великим попитом на захисне та тактичне спорядження серед військових, волонтерів.

Метою випускної кваліфікаційної роботи є проектування та реалізація веб-орієнтованого інтернет-магазину для продажу військових товарів із забезпеченням функціоналу перегляду асортименту, фільтрації, кошика, оформлення замовлень і управління товарами з боку адміністратора.

Мета дослідження включає наступні завдання:

- дослідити особливості торгівлі військовими товарами та потреби цільової аудиторії;
- проаналізувати існуючі рішення та виділити їх переваги та недоліки;
- сформулювати вимоги до функціоналу системи;
- розробити архітектуру веб-додатку та базу даних;

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

- реалізувати веб-додаток на основі фреймворку Laravel;
- забезпечити функціонування адміністративної панелі;
- протестувати систему на відповідність технічним вимогам.

Об'єктом дослідження є електронна комерція у сфері військових товарів.

Предметом дослідження є процеси проектування та реалізації веб-орієнтованих інформаційних систем для продажу військових товарів.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ІНТЕРНЕТ-МАГАЗИНУ ВІЙСЬКОВИХ ТОВАРІВ

1.1. Постановка задачі

У сучасному світі зростаючий попит на військові, тактичні та захисні товари серед військовослужбовців, служб безпеки, волонтерських організацій створює потребу у зручному, надійному та захищеному інструменту онлайн-торгівлі. Це зумовлено як умовами війни, так і розвитком технологій онлайн-продажів. Водночас поширення мобільних пристроїв і розвиток інтернету стимулюють користувачів здійснювати покупки саме через веб-сайти. Покупці очікують на швидкий доступ до якісної продукції, можливість швидко та зручно оформити замовлення з гарантіями безпеки при оплаті й передачі персональних даних [14].

Ринок військових товарів має низку особливостей, які впливають на архітектуру інформаційної системи. По-перше, необхідна чітка класифікація товарів (тактичне спорядження, форма, рюкзаки, засоби захисту, тощо). По-друге, існують обмеження в продажу окремих категорій товарів (наприклад, холодної зброї, спецзасобів). Крім того, важливо забезпечити швидке оновлення залишків, адже в умовах воєнного часу попит може змінюватись щоденно. Критично важливим є забезпечення стабільної роботи веб-додатку під час пікових навантажень, а також захисту персональних даних клієнтів і адміністраторів магазину [18].

Саме тому виникає потреба у створенні сучасного інтернет-магазину військових товарів, який буде враховувати ці проблеми. Така система має не лише виконувати базові функції продажу товарів, а й бути зручною у користуванні.

Метою роботи є створення інтернет-магазину який би забезпечував повний цикл онлайн-продажу: від перегляду продукції до формування замовлення, управління товарною базою, обробки клієнтських даних і

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

подальшого масштабування.

У процесі виконання кваліфікаційної роботи необхідно вирішити низку ключових завдань.

Насамперед потрібно дослідити особливості торгівлі військовими товарами як в Україні, так і за кордоном, а також проаналізувати потреби користувачів. Далі варто вивчити аналогічні інформаційні системи, проаналізувати структуру, функціональні можливості та зручність користування популярними сайтами (наприклад, Abrams, Militarist, Vik Tailor). Це допоможе виділити сильні сторони і виявити недоліки конкурентів.

Наступним кроком є формування вимог до системи. Потрібно скласти повний перелік вимог, розділивши їх на функціональні, нефункціональні та технічні, а також визначити рівень доступності, вимоги до захисту інформації та створити діаграму варіантів використання, діаграми діяльності та діаграми класів.

Після цього слід перейти до проектування архітектури системи, обрати відповідний архітектурний підхід, описати логіку взаємодії між модулями, спроектувати базу даних і обґрунтувати вибір технологій.

Під час реалізації необхідно створити веб-додаток. Для цього потрібно розробити інтерфейс користувача з каталогом товарів, системою фільтрації, переглядом товарів, кошиком і оформленням замовлення. Також має бути створена адміністративна панель для управління товарами, замовленнями та категоріями. Важливо забезпечити автентифікацію та авторизацію користувачів. Завершальним етапом є тестування системи. Треба перевірити, чи відповідає веб-додаток визначеним вимогам, виявити помилки та усунути їх.

Результатом виконаної роботи є розроблений інтернет-магазин військових товарів, який забезпечує інтуїтивно зрозумілий інтерфейс для користувача, зручний кабінет адміністратора, реалізована система авторизації та реєстрації, що дозволить користувачам спростити процес купівлі товарів а адміністраторам керувати продукцією.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2. Аналіз аналогів програмного продукту

У сучасному ринку існує багато сайтів, які спеціалізуються на продажі військових товарів. Аналіз функціональності, дизайну та досвіду користування таких платформ є важливим при розробці власного інтернет-магазину, це дозволяє врахувати найкращі рішення і поширені недоліки конкурентів.

Проведемо порівняльний аналіз популярних інтернет-магазинів військових товарів, такі сайти, як Abrams, Militarist, Velmet, Punisher, а також іноземний магазин Militaria.pl.

Першим сайтом є Abrams (рис. 1.1). Це один з найпопулярніших інтернет-магазинів на ринку військових товарів, відрізняється сильним мілітаристським стилем. Інтерфейс має приємний дизайн, темну гаму, динамічні банери, популярні товари, категорії.

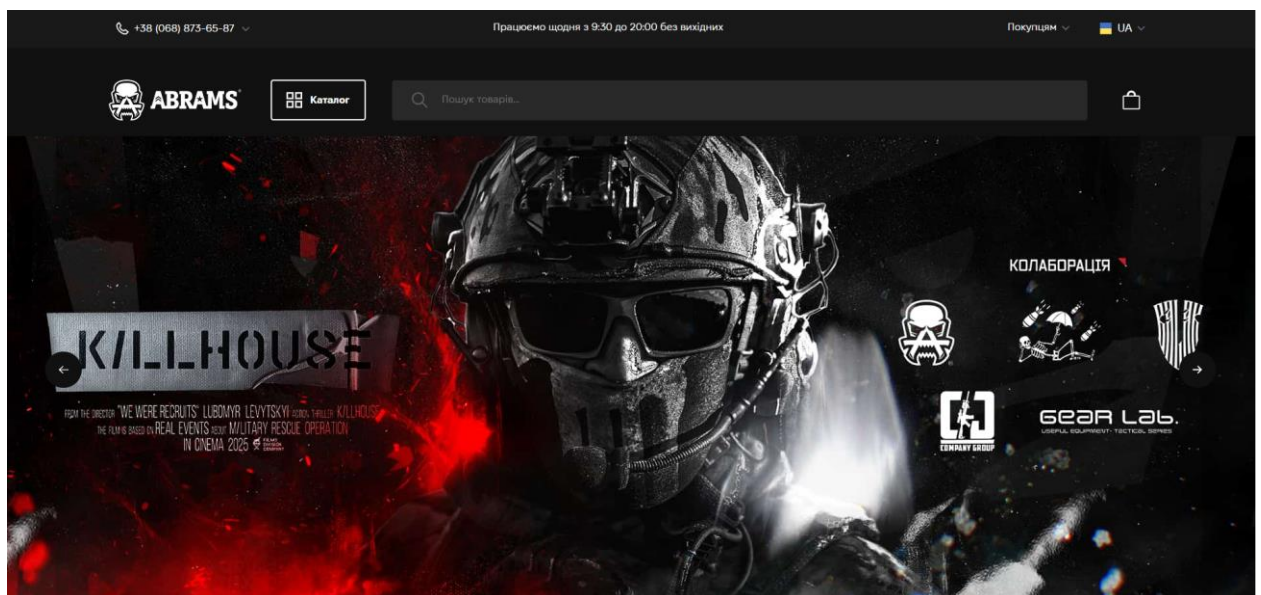


Рис. 1.1. Інтернет-магазин військових товарів Abrams[1]

Серед ключових позитивних аспектів варто зазначити сучасний дизайн, який добре передає атмосферу військової тематики, а також двомовну підтримку інтерфейсу – українською та англійською мовами. Навігація є зрозумілою для користувача: каталог товарів чітко виділений, рядок пошуку розташований у зручному місці, логіка взаємодії інтуїтивно проста. Також реалізована зручна фільтрація продукції за категоріями, що полегшує пошук.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Додатковою перевагою є окремі блоки які присвячені брендам і фірмовим лінійкам. Також варто відзначити добру адаптацію інтерфейсу до мобільних пристроїв.

Попри ці переваги, сайт має і певні недоліки. Зокрема, відсутній повноцінний особистий кабінет. Хоча при оформленні замовлення користувачу пропонується увійти в акаунт, відповідна кнопка при натисканні не функціонує, а можливість авторизації на сайті візуально відсутня. Крім того, локалізація українською мовою реалізована не повністю, деякі елементи інтерфейсу залишаються англійською. Ще одним слабким місцем є пошук, він працює лише за точними назвами товарів, не шукає співпадиння в описі, бренді або ключовими словами.

Militarist (рис. 1.2) це інтернет-магазин який продає тактичне спорядження, одяг, взуття, аксесуари, бронежилети тощо. Сайт побудовано у класичному стилі.

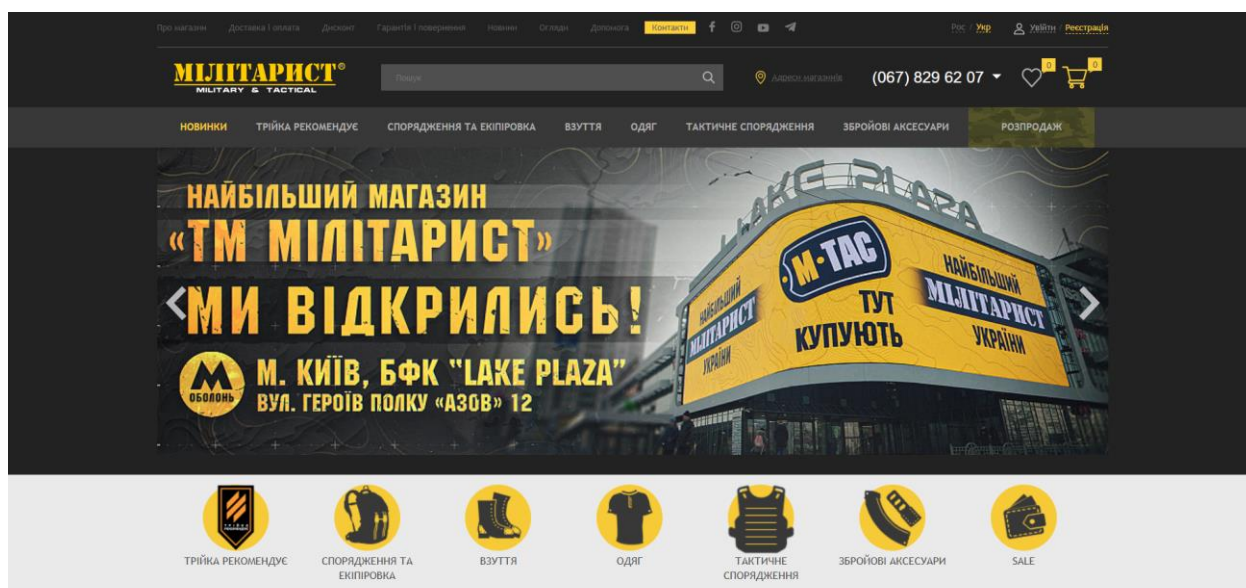


Рис. 1.2. Інтернет-магазин військових товарів Militarist[8]

Верхнє меню сайту має чітко структуровані основні категорії та розпродажі, така навігація полегшує пошук товару. Також є функціонал реєстрації та авторизації, це дає змогу переглядати історію покупок, адреси доставки та інше. Важливим є наявність оглядів майже на всі популярні

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

товари, присутній блог який активно ведеться. Каталог продукції має детальні описи, характеристики, фото та іноді відеоогляди.

Дизайн сайту хоча і є функціональним, але виглядає морально застарілим і не має сучасного вигляду. Адаптивність для мобільних пристроїв відсутня, на телефоні користуватись магазином не зручно. Крім цього, інтерфейс перевантажений великою кількістю інформації, це може дезорієнтувати користувача. Сайт погано оптимізований, нерідко сторінки завантажуються більше п'яти секунд.

Наступним сайтом є Velmet (рис. 1.3). Це брендовий сайт виробника військового екіпірування, його особливістю є окрема система замовлень та акцент на власну продукцію з детальними описами та оглядами.

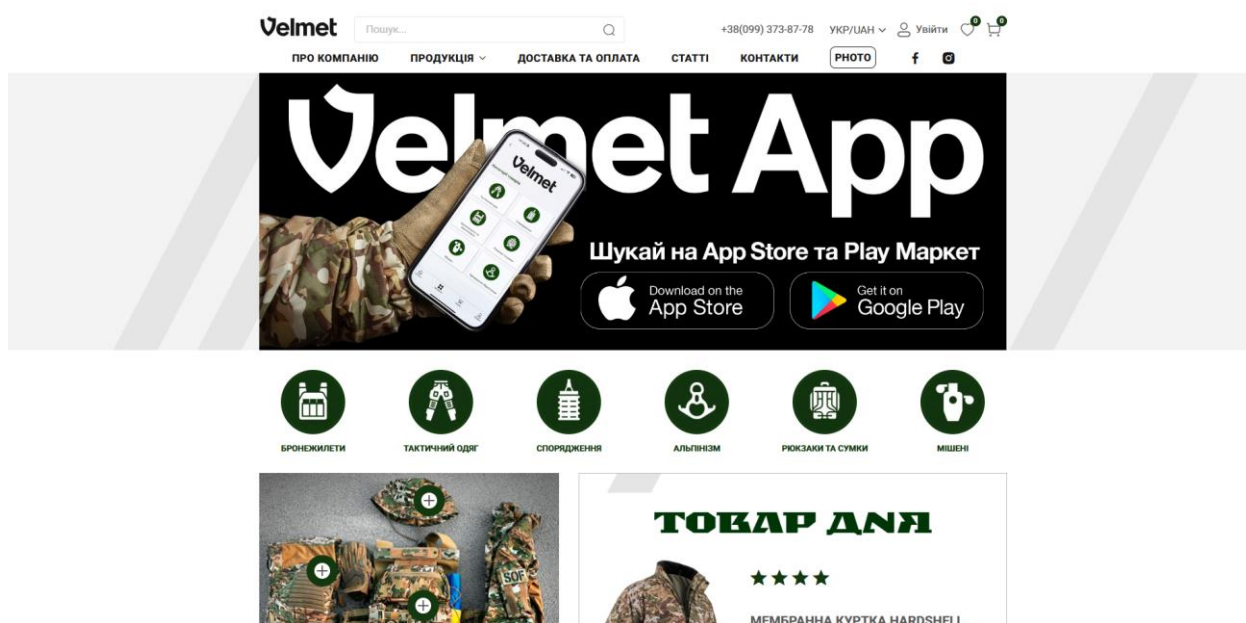


Рис. 1.3. Інтернет-магазин військових товарів Velmet[10]

Сайт має простий та зручний дизайн, ним приємно користуватись. Помітно що робиться акцент на українське виробництво. Реалізовано особистий кабінет, який дозволяє користувачам зберігати свої дані, переглядати історію замовлень і керувати адресами доставки. Інтерфейс підтримує як українську, так і англійську мову, також є можливість змінювати валюту між гривнею, долларом та євро, це зручно для клієнтів з-за кордону. Товари добре представлені в каталозі, є зображення, короткий опис, варіанти та ціна.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Однак пошукова система працює слабо, не завжди вдається знайти товар за ключовими словами. Відсутня детальна фільтрація за характеристиками, як-от розмір, колір чи наявність, що ускладнює вибір. Також сайт підтримує не всі сучасні способи оплати, тому можуть виникати незручності під час оформлення замовлення. Мобільна адаптація присутня, але деякі елементи потрібно доопрацювати для зручності.

Punisher (рис. 1.4) - український спеціалізований онлайн-магазин, орієнтований на продаж військових товарів. Магазин поєднує онлайн-платформу з фізичним магазином розташованим у Львові.

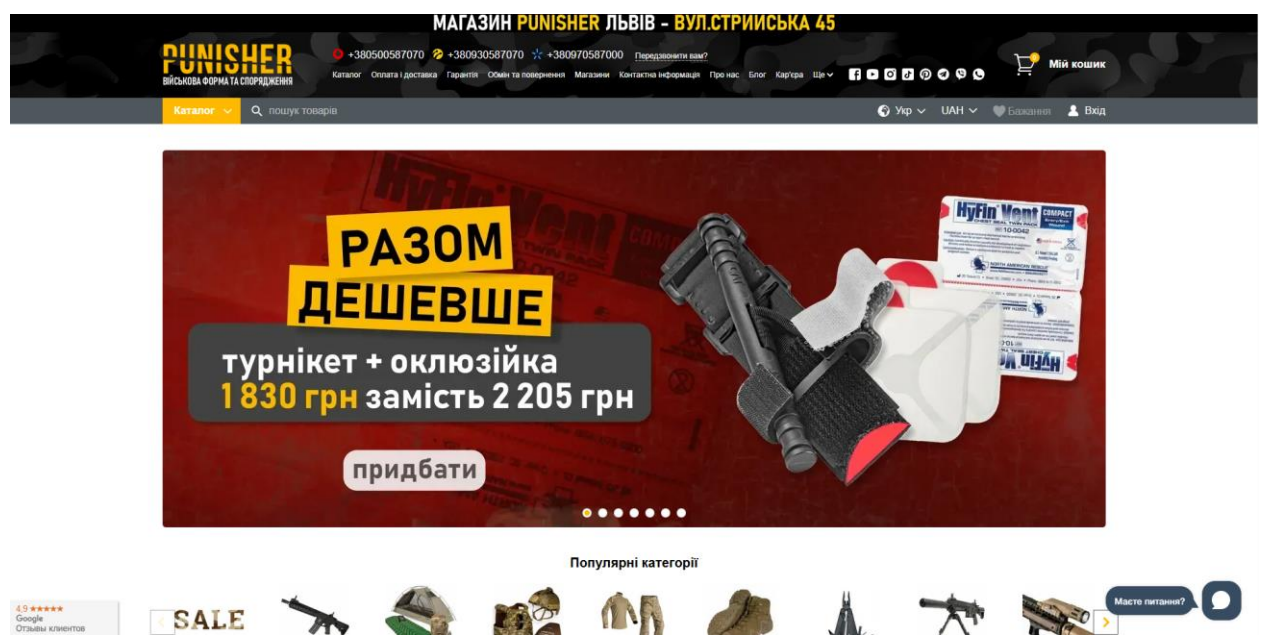


Рис. 1.4. Інтернет-магазин військових товарів Punisher[9]

На головній сторінці відображаються популярні товари, акційні пропозиції та найпопулярніші категорії, що дозволяє швидко ознайомитись з основним асортиментом. Реалізована детальна система фільтрації, яка дає змогу сортувати товари за кількома параметрами, за ціною, категоріями та іншими характеристиками. Для зворотного зв'язку є чат підтримки та кнопки для спілкування з підтримкою в месенджерах, що дуже полегшує комунікацію. Сайт підтримує українську та англійську мови, а також зміну валюти. Є можливість реєстрації та входу в особистий кабінет.

Недоліком є візуальне оформлення яке виглядає застарілим і не відповідає сучасним трендам дизайну. Сайт погано оптимізований, сторінки

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

завантажуються повільно. Також він не адаптований для мобільних пристроїв. Крім того, хоч особистий кабінет і присутній, але у ньому немає історії покупок, що знижує його користь для покупців. У цілому сайт виконує основні функції, але потребує оновлення дизайну, оптимізації та особистого кабінету.

Останнім сайтом є Militaria.pl (рис. 1.5). Це потужний європейський інтернет-магазин, орієнтований на продаж військової продукції, має понад 100 тисяч позицій, багатомовність.

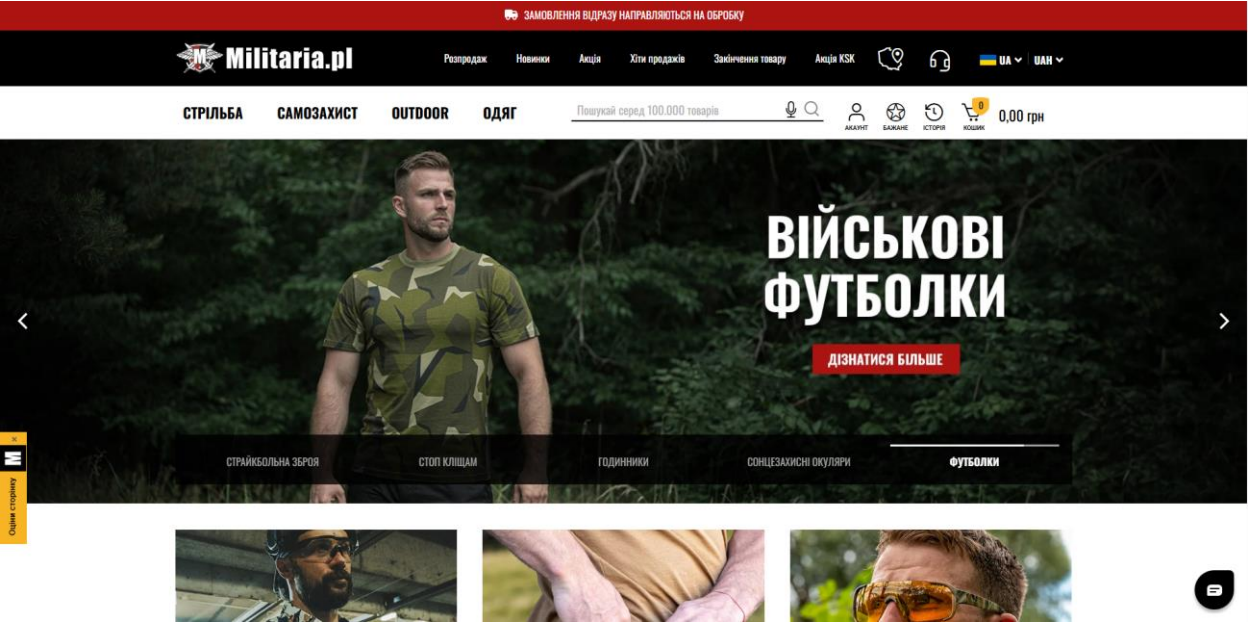


Рис. 1.5. Інтернет-магазин військових товарів Militaria.pl[7]

Сайт справляє позитивне враження завдяки сучасному дизайну та великій кількості функцій. Підтримується голосовий пошук, що є досить зручним для швидкого знаходження товару. Доступна історія переглядів, можливість порівнювати товари та додавати їх у список бажань. Каталог має деталізовану фільтрацію. Також сайт підтримує всі популярні платіжні системи, включаючи банківські картки, Apple Pay, Google Pay та інші. Окремий плюс - якісні фото, стильне оформлення та фокус на брендові позиції що створює враження професійного магазину. Перегляд товарів зручний і добре продуманий.

Попри це, сайт має й недоліки. Інтерфейс дещо перевантажений, на сторінках багато банерів, реклами, що може відволікати. Мобільна версія працює, але не така зручна як на комп'ютері, деякі елементи дрібні або слабо

адаптовані. Хоча доступна українська мова, перегляд іноді виглядає неповним або неточним. Також відчувається потреба в оптимізації, сторінки іноді завантажуються дуже довго.

Склавши переваги та недоліки у таблиці 1.1 можемо скласти порівняльну характеристику аналогів.

Таблиця 1.1

Таблиця порівняння аналогів інтернет-магазину військових товарів

Критерій	Abrams	Militarist	Velmet	Punisher	Militaria.pl	Розробляємий сайт
Дизайн	+	-	+	-	+	+
Навігація	+	+	+	+-	+	+
Особистий кабінет	-	+	+	+	+	+
Пошук	+-	+-	+-	+-	+	+
Рекомендації	+	-	+	-	+	-
Оплата	+	+-	+	+-	+	+
Адаптивність	+	+-	+	-	+	+
Наповненість	+	+	+-	+	+	+-
Зручність	+	+-	+	+-	+	+
Оптимізація	+	-	+	-	+-	+-

У ході аналізу існуючих інтернет-магазинів військових товарів було виявлено низку закономірностей, які можуть бути враховані під час розробки власного сайту. Як видно із порівняльного аналізу, за більшістю показників сайти Abrams та Militaria.pl вирізняються сучасним дизайном, інтуїтивною навігацією та широким функціоналом. Натомість сайти Punisher та Militarist виявилися менш зручними в користуванні, з обмеженою функціональністю та застарілим оформленням. Також було виявлено, що пошукова системи та рекомендаційна система це основні слабкі місця навіть у найвідоміших сайтів.

Для узагальнення у таблиці 1.2 виділено основні переваги та недоліки проаналізованих веб-ресурсів.

Таблиця 1.2

Переваги та недоліки розглянутих сайтів

Переваги	Недоліки
Гарна категоризація товарів	Застарілі елементи інтерфейсу на окремих сайтах
Особисті кабінети (у більшості випадків)	Недостатня інтеграція з сервісами онлайн оплати
Адаптація під мобільні пристрої	Великий функціонал з втратою зручності
Дизайн під тематику	Слабка пошукова система

Підсумовуючи проведений аналіз, було сформовано чітке уявлення про ключові функціональні вимоги, сильні та слабкі сторони конкурентів, це буде перевагою при розробці власного сайту. На основі цього було вирішено, який функціонал варто реалізувати, а від якого варто відмовитись.

Для зручного представлення функціоналу який планується реалізувати було створено схему з основними компонентами та їх описом.

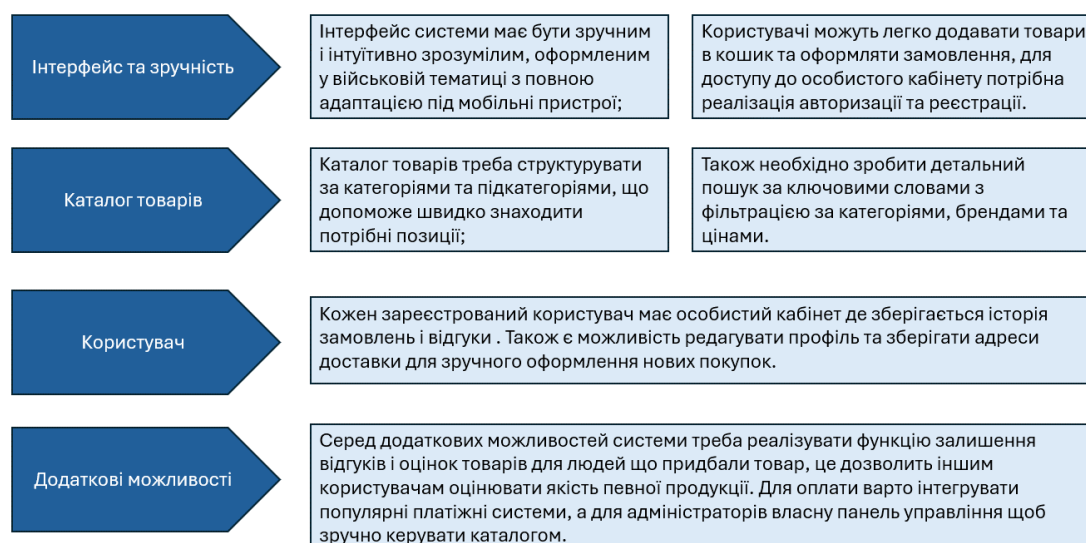


Рис. 1.6. Основні функції передбачені до реалізації

Деякі функції не будуть реалізовані. Зокрема, не передбачено голосовий пошук, оскільки він використовується рідко. Також система порівняння товарів, адже для більшості користувачів достатньо інформації в описі товару. Рекомендаційна система не буде реалізована через складність налаштування та відсутність великого обсягу даних на початку роботи сайту. А ще створення блогу, оскільки основна увагу проєкту зосереджена на продажі товарів.

Таким чином, при розробці сайту будуть враховані найкращі рішення конкурентів, уникнено їхніх слабких місць. Основна мета полягає у створенні функціонального інтерфейсу який буде зручним та зрозумілим для звичайного користувача.

1.3. Вибір архітектури веб-орієнтованої системи

Розробка інтернет-магазину військових товарів передбачає використання архітектурної моделі, яка забезпечить масштабованість, модульність, безпеку для проєкту. Серед веб-додатків найпоширенішими архітектурними підходами є монолітна архітектура, мікросервіси, архітектура MVC (Model-View-Controller), serverless-підхід та headless CMS. У цьому проєкті вирішено використати архітектуру MVC. Цей вибір пояснюється гнучкістю, можливістю додавання нових функцій та легкою структурою, яка розділяє логіку, інтерфейс та обробку даних.

Архітектура MVC дозволяє легко масштабувати систему, використовувати компоненти кілька разів та забезпечує зручність у тестуванні. Розділення логіки дозволяє одночасно працювати над інтерфейсом (View), логікою (Controller) та базою даних (Model).

Також розглянуто альтернативні рішення. Монолітна архітектура проста в реалізації, але важка для обслуговування та розширення. Мікросервіси дозволяють розширювати окремі частини системи, але потребують більше зусиль для взаємодії між сервісами. Serverless дозволяє легко масштабувати, але залежить від хмарних сервісів. Headless CMS ідеально підходить для вебсайтів з контентом, але потребує складної інтеграції. Для об'єктивного

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

вибору архітектури у таблиці 1.3 подано порівняльний аналіз переваг та недоліків основних архітектурних стилів.

Таблиця 1.3

Порівняльна характеристика архітектурних стилів [12]

Архітектура	Опис	Переваги	Недоліки
MVC	Розділення моделі, уявлення та контролера	Гнучкість, модульність, хороша підтримка	Потрібен досвід структуризації коду
Monolithic	Усі функції в одному додатку	Простота реалізації, швидкий старт	Складність масштабування та обслуговування
Microservices	Кожен компонент це окремий сервіс	Незалежність, масштабування, відмовостійкість	Ускладнення DevOps, синхронізація сервісів
Serverless	Подія запускає окрему функцію	Автоматичне масштабування, немає серверного коду	Висока залежність від хмарних платформ
Headless CMS	API і фронтенд розробляється окремо	Гнучкість для фронтенду, швидкість реакції	Потребує складної інтеграції та безпеки

Аналіз показав, що саме MVC забезпечує оптимальний баланс між простотою реалізації, підтримки сучасних технологій та можливістю подальшого розширення.

Для PHP існує багато популярних MVC фреймворків, такі як Laravel, Symfony, Codelgniter, Yii та Laminas. Щоб обґрунтувати вибір для цього проєкту, було проведено порівняльний аналіз основних фреймворків, представлений у таблиці 1.4.

Таблиця 1.4

Порівняння популярних MVC фреймворків для PHP [21]

Критерій	Laravel	Symfony	Codelgniter	Yii	Laminas
Архітектура	MVC, REST, HMVC підтримка	MVC, дуже гнучка	MVC	MVC + Gii генератор	MVC + компоненти
Простота освоєння	Висока	Низька	Дуже висока	Середня	Низька
Швидкість роботи	Висока	Середня	Висока	Висока	Середня
Вбудовані можливості	Багато (auth, queue, mail)	Базові можливості + додаткові	Мінімальні	Потужний кеш, RBAC, API	Широкий набір, але вручну
Підтримка ORM	Eloquent	Doctrine	Відсутня	ActiveRecord	Contrine

На основі обраної MVC-архітектури для розробки запропоновано використовувати фреймворк Laravel. Він підтримує повну реалізацію MVC, має вбудовані засоби для захисту (CSRF, валідація даних), забезпечує швидке створення маршрутів, міграцій баз даних та інтеграцію з платіжними сервісами. Це дозволяє зменшити час розробки та забезпечити високу якість програмного коду. Важливою перевагою є також підтримка шаблонізатора Blade, ORM Eloquent для роботи з базами даних, механізмів middleware та

модульності системи. Завдяки великій спільноті Laravel має багатий набір пакетів, що значно спрощує реалізацію складних функцій [24]. Усе це дозволяє забезпечити розширюваність, повторне використання компонентів і зручне адміністрування.

Таким чином, архітектура MVC з використанням Laravel забезпечує найкращий баланс між простотою, масштабованістю, безпекою та підтримкою.

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

У розробці інтернет-магазину військових товарів важливу роль відіграє правильний вибір інструментальних засобів, які забезпечать швидкість реалізації, підтримку сучасних стандартів, масштабованість та зручність в обслуговуванні системи [27]. Основним критерієм при виборі інструментальних засобів стала їхня підтримка архітектури MVC, інтеграція з платіжними сервісами, адаптивний інтерфейс та безпечна обробка даних.

У якості мови розробки обрано PHP [25], адже вона дуже популярна для веб-розробки, підтримується більшістю хостингів та має багато корисних бібліотек. Laravel як фреймворк для PHP дозволяє реалізувати структуру MVC, має зручний шаблонізатор Blade, систему для роботи з базами даних Eloquent та дозволяє легко здійснювати міграції баз даних. MySQL [13] виступає основною системою управління базами даних, оскільки вона стабільна, масштабована та добре працює з Laravel через Eloquent ORM.

Для розробки інтерфейсу використовуються HTML5, CSS3, JavaScript з Bootstrap та jQuery. Bootstrap [2] допомагає швидко створювати адаптивний дизайн, а jQuery спрощує роботу з елементами на сторінці. Для розробки використовується Visual Studio Code [6], який має багато плагінів, підтримує Git та має інтеграцію з терміналом. Composer [3] допомагає керувати залежностями проєкту, полегшуючи встановлення бібліотек.

Управління залежностями в проєкті ведеться через Composer, а контроль версій за допомогою Git та GitLab [4]. Це дозволяє відслідковувати зміни в

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

кодi та повертатися до попереднiх версiй. Apache використовується як веб-сервер, забезпечуючи стабiльну роботу PHP-додаткiв. Для створення та тестування додатку на локальному серверi застосовується XAMPP – зручне зередовище, що об'єднує Apache, MySQL, PHP та phpMyAdmin. Завдяки цьому локальне тестування здiйснюється без потреби налаштовувати сервер вручну.

Нижче наведено обґрунтування вибору кожного з iнструментальних засобiв у таблицi 1.5.

Таблиця 1.5

Обґрунтування вибору iнструментальних засобiв

Компонент	Обраний iнструмент	Обґрунтування
Мова розробки	PHP	Широко використовується у веб-розробцi, сумiсна з Apache, пiдтримується багатьма хостингами
Фреймворк	Laravel	Пiдтримка MVC, Blade-шаблони, зручна ORM (Eloquent), маршрутизацiя, авторизацiя, мiграцiї БД, масштабованiсть
База даних	MySQL	Реляцiйна СУБД, ефективна для зберiгання структурованих даних (наприклад, товарiв, замовлень, користувачiв), пiдтримується Laravel з ORM
Сервер	Apache	Надiйний i стабiльний HTTP-сервер, легко конфiгурується для PHP додаткiв
iнтерфейс	HTML5, CSS3, JavaScript	Стандартнi засоби фронтенду для реалiзацiї зручного та адаптивного iнтерфейсу

Бібліотеки JS	Bootstrap, jQuery	Bootstrap – для адаптивного дизайну, jQuery – для покращення взаємодії користувача
Бібліотеки JS	Bootstrap, jQuery	Bootstrap – для адаптивного дизайну, jQuery – для покращення взаємодії користувача
Розробка і тестування	Visual Studio Code	Зручне середовище розробки з підтримкою PHP, Blade, Git
Керування залежностями	Composer	Стандартний інструмент для керування бібліотеками PHP
Контроль версій	Git + GitLab	Забезпечує контроль змін

Вибір інструментів і середовища розробки забезпечує ефективну реалізацію системи відповідно до поставлених вимог. Використання Laravel у зв'язці з Apache, PHP і MySQL дозволяє побудувати гнучкий та надійний веб-додаток із сучасним інтерфейсом та зручною архітектурою [31].

Щодо вимог до апаратного забезпечення, для локальної розробки необхідний комп'ютер із процесором не нижче ніж 4-ядерний 2.5 ГГц, 8 ГБ оперативної пам'яті та не менше ніж 20 ГБ вільного місця на диску. Система повинна працювати на операційних системах Windows, macOS або Linux. Також рекомендовано використовувати SSD диск для підвищення швидкості обробки запитів.

Отже, обрані інструменти дозволяють розробити ефективний та безпечний веб-додаток, масштабувати його та адаптувати під різні платформи.

Висновки до першого розділу.

У першому розділі проаналізовано предметну область електронної комерції у сфері військових товарів. На основі визначених потреб

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

користувачів і специфіки продукції визначено основні завдання системи. Проведено огляд і порівняння існуючих рішень, що дозволило виявити переваги та недоліки сучасних інтернет-магазинів.

У межах аналізу визначено доцільність використання архітектури MVC, що забезпечить логічну структуру коду, модульність і зручність розробки. Для реалізації інтернет-магазину обрано фреймворк Laravel на базі технологій Apache, MySQL, PHP, що дозволить реалізувати необхідну функціональність із високим рівнем безпеки та продуктивності.

Також обґрунтовано вибір інструментів розробки та визначено вимоги до апаратного забезпечення. У результаті сформовано набір основних технологій і засобів реалізації що забезпечать високий рівень ефективності та безпеки.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ ВІЙСЬКОВИХ ТОВАРІВ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Створення інтернет-магазину військових товарів вимагає заздалегідь скласти перелік вимог до функціоналу, швидкодії, зручності користування та захисту, що стане основою для подальшої розробки.

Функціональні вимоги до системи формуються на основі очікувань майбутніх користувачів, а також логіки бізнес процесів [30]. Всі функції можна умовно поділити за двома основними ролями – покупець і адміністратор. Для покупця система має забезпечити можливість швидкої реєстрації та входу до особистого кабінету. Після цього користувач отримує доступ до повного каталогу товарів, який можна переглядати у зручному вигляді. Також передбачено функціонал пошуку і фільтрації товарів за категоріями, брендами та ціною, що дозволяє легко знайти потрібний товар. На сторінці кожного товару відображаються фото, опис, вартість та варіанти, після чого покупець може додати товар до кошика та оформити замовлення. Для зареєстрованих користувачів доступна історія покупок, що дозволяє переглядати інформацію і статуси замовлення. Для адміністраторів система має інший функціонал. Вони отримують доступ до панелі керування де можуть додавати, редагувати або видаляти товари. Також має бути можливість створювати категорії, підкатегорії та дочірні категорії, щоб зручно керувати каталогом. Адміністратор може обробляти замовлення, переглядати їх деталі, змінювати статус (наприклад, в обробці, доставлено). Крім того, передбачено базове керування акаунтами користувачів (у адміністратора є можливість видавати ролі).

Згідно з нефункціональними вимогами системи, вона повинна реагувати на дії користувача не більше ніж 1,5 секунди при стандартному навантаженні до 100 одночасних користувачів. Інтерфейс має бути гнучким, щоб коректно

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

відображався на всіх сучасних мобільних і десктопних пристроях. Особливу увагу має бути приділено безпеці, автентифікацію потрібно реалізувати з хешуванням паролів, захистом від CSRF, SQL-ін'єкцій, також слід зробити валідацію введених даних [20]. Система повинна бути надійною, у випадку помилок має бути передбачено збереження даних. Крім того, архітектура повинна дозволяти додавання нових модулів без переписування коду.

Для розгортання системи було визначено такі системні вимоги. Необхідний комп'ютер з чотириядерним процесором із тактовою частотою 2.5 ГГц або вище, оперативною пам'яттю обсягом не менше ніж 8 ГБ, а також вільним місцем на диску обсягом щонайменше 20 ГБ. Система повинна працювати під керуванням операційної системи Windows, Linux або macOS. Для розгортання використовуються XAMPP, PHP версії 8.0 та більше і база даних MySQL 8.0.

Часові характеристики системи визначають зручність та швидкість взаємодії користувача з інтерфейсом. Завантаження головної сторінки повинно відбуватись не довше ніж за 3 секунди, що забезпечить комфортне користування сайтом. Дії користувача, такі як фільтрація товарів чи додавання до кошика повинні виконуватись майже миттєво, з часом відгуку до 2 секунд. Обробка запиту під час оформлення замовлення також не повинна перевищувати 3 секунд.

Для ефективного функціонування веб-орієнтованої системи інтернет-магазину необхідно зробити сценарії взаємодії користувача з системою, а також визначити ключові сутності, їх атрибути та взаємозв'язки. Це дозволить закласти основу для побудови моделі системи.

Адміністратор має розширені права доступу та відповідає за управління каталогом товарів, обробку замовлень, керування обліковими записами користувачів та інші адміністративні функції. Покупцем може бути як гість (незареєстрований користувач), так і зареєстрований користувач. В обох випадках він має змогу переглядати товари, додавати їх до кошика, однак

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

оформлення замовлення та доступ до особистого кабінету мають тільки зареєстровані користувачі.

Для кращого розуміння логіки роботи системи та взаємодії користувачів з компонентами було створено діаграму варіантів використання (рис. 2.1), яка ілюструє типові сценарії взаємодії між користувачем та функціональними модулями додатку.

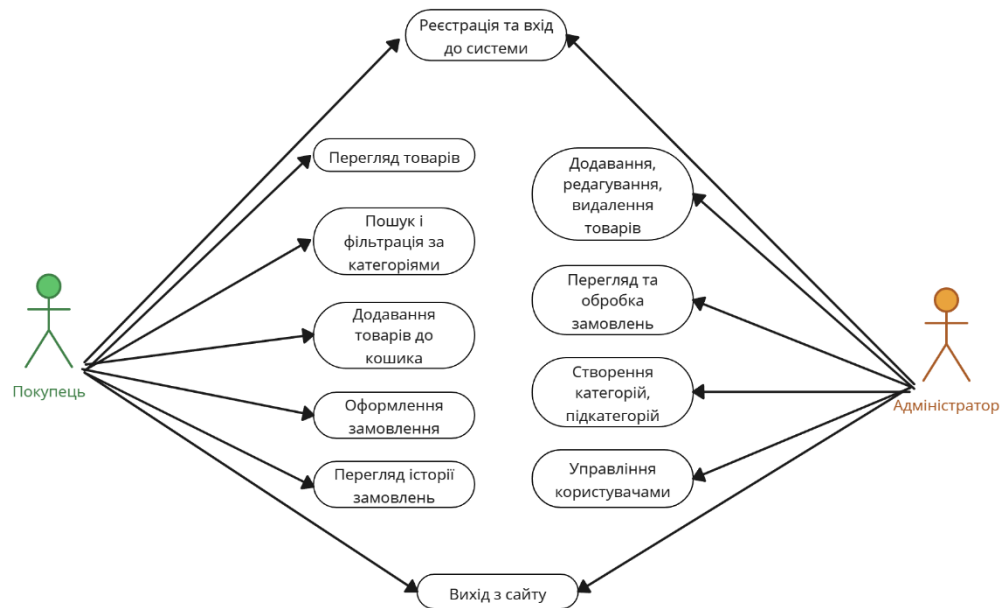


Рис. 2.1. Діаграма варіантів використання

На діаграмі варіантів використання відображено основні дії які можуть виконувати користувачі системи відповідно до своєї ролі (покупець або адміністратор). Діаграма охоплює повний життєвий цикл користувача в системі, від входу до завершення замовлення чи адміністрування. Вона є важливим для візуалізації функціональності додатку та подальшого проєктування архітектури.

Для покупця передбачені наступні варіанти використання. Це реєстрація або вхід до системи, що дозволяє створити обліковий запис або увійти до вже існуючого. Після входу користувач може переглядати каталог товарів, шукати продукції, фільтрувати її за категоріями, брендами чи ціною. Обрані товари додаються до кошика, після чого виконується оформлення замовлення,

вводиться контактна інформація, обирається спосіб доставки та підтвердження покупки. Для зареєстрованих користувачів доступна функція перегляду історії замовлень. Завершити сеанс можна через функцію виходу з системи.

Щодо адміністратора, то він після авторизації отримує доступ до управління контентом. Зокрема, він може додавати, редагувати або видаляти товари, керувати категоріями та підкатегоріями каталогу. Також адмін переглядає й обробляє замовлення, змінює їх статус, переглядає деталі, підтверджує або скасовує покупки. Окремою функцією є управління користувачами: перегляд профілів, модерація доступу. Сесію адміністратора також можна завершити через функцію виходу з панелі.

Ця діаграма є основою для визначення логіки роботи системи, побудови структури бази даних і формування контролерів. Вона дозволяє чітко відобразити, які дії має підтримувати додаток і забезпечує зручну візуалізацію функціональних вимог [16].

На основі аналізу функціональних вимог було виділено наступні сутності, що реалізують бізнес-логіку системи. Ці сутності відображено у вигляді UML-діаграми класів, наведеної на рисунку 2.2. Вона демонструє основні об'єкти, їхні поля, методи та зв'язки. У фреймворку Laravel для побудови взаємозв'язків між моделями використовуються стандартні методи. Наприклад, `belongsTo` означає що модель належить до іншої, а `hasMany` – відношення “один до багатьох”. Прикладом такої взаємодії є `product()->belongsTo(Category::class)`, означає що товар належить до певної категорії.

Діаграма демонструє ключові зв'язки між сутностями системи [17]. Один користувач може мати багато замовлень, адрес доставки, відгуків та товарів в списку бажань. Кожен товар має зв'язок з категорією, підкатегорією, дочірньою категорією та брендом. Також товар може мати кілька зображень (`ProductImageGallery`), кілька варіантів (`ProductVariant`), а кожен варіант має кілька опцій (`ProductVariantItem`), наприклад, розмір L.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

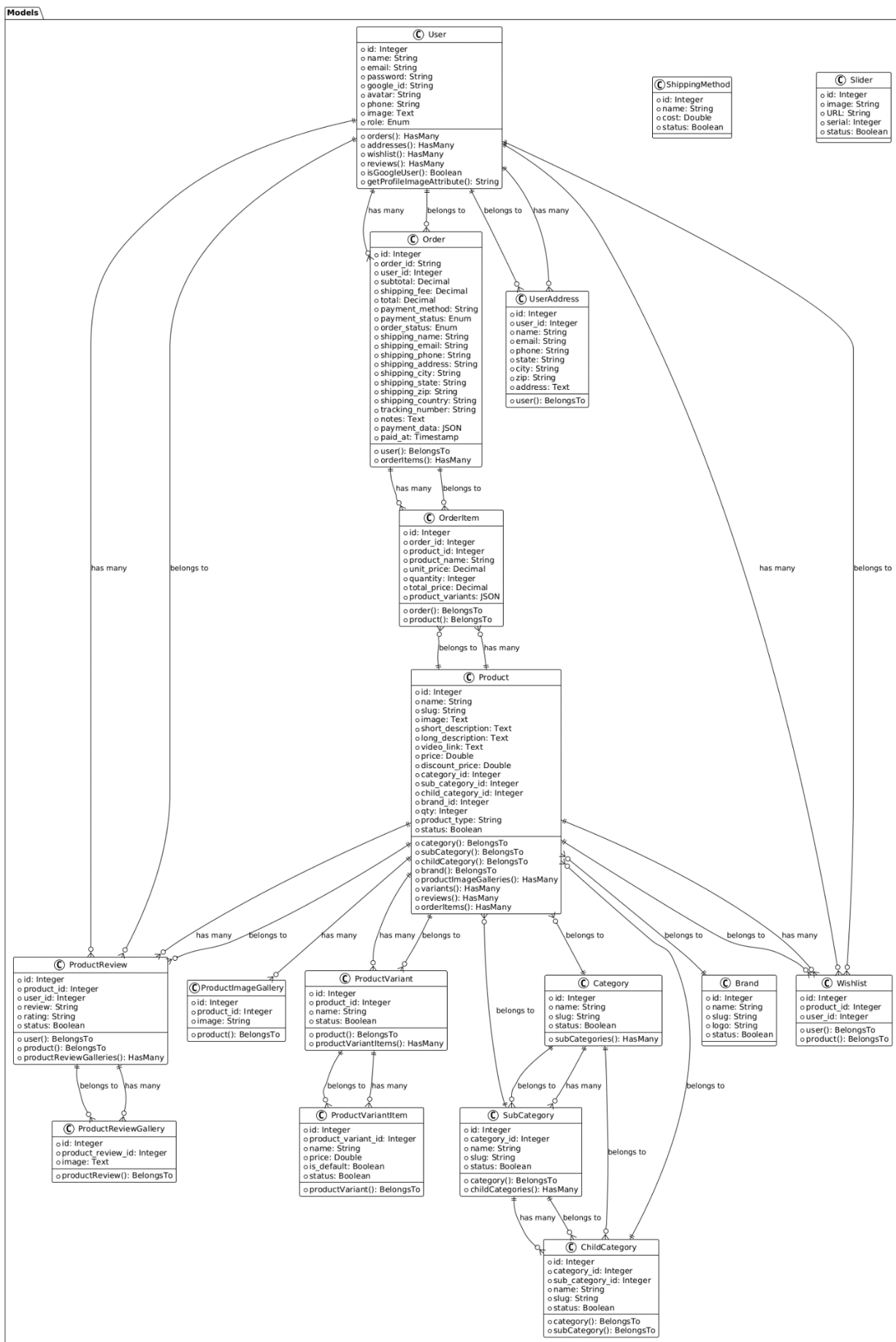


Рис. 2.2. Діаграма класів

Важливим є також зв'язок категорій між собою. Основна категорія (Category) може містити підкатегорії (SubCategory), які мають дочірні категорії (ChildCategory). Така структура дозволяє створити гарну категоризацію каталогу. Також є об'єкти які не мають зв'язків, але забезпечують певну функціональність, наприклад, Slider використовується для динамічної зміни слайдера банерів на головній сторінці, а ShippingMethod для методів доставки (наприклад, Нова Пошта або Укрпошта).

Діаграма класів дозволяє візуалізувати логіку побудови моделей у системі, відобразити їх зв'язки, та слугує основою для формування бази даних.

2.2. Розробка бази даних системи

Для збереження, обробки та організації інформації для інтернет-магазину військових товарів використовується реляційна база даних MySQL. Дана система управління базами даних обрана завдяки своїй надійності, широкій підтримці, високій продуктивності та повній інтеграції з фреймворком Laravel, що дозволяє ефективно працювати з базою даних за допомогою вбудованої ORM системи Eloquent [23].

На рисунку 2.3 представлено схему бази даних.

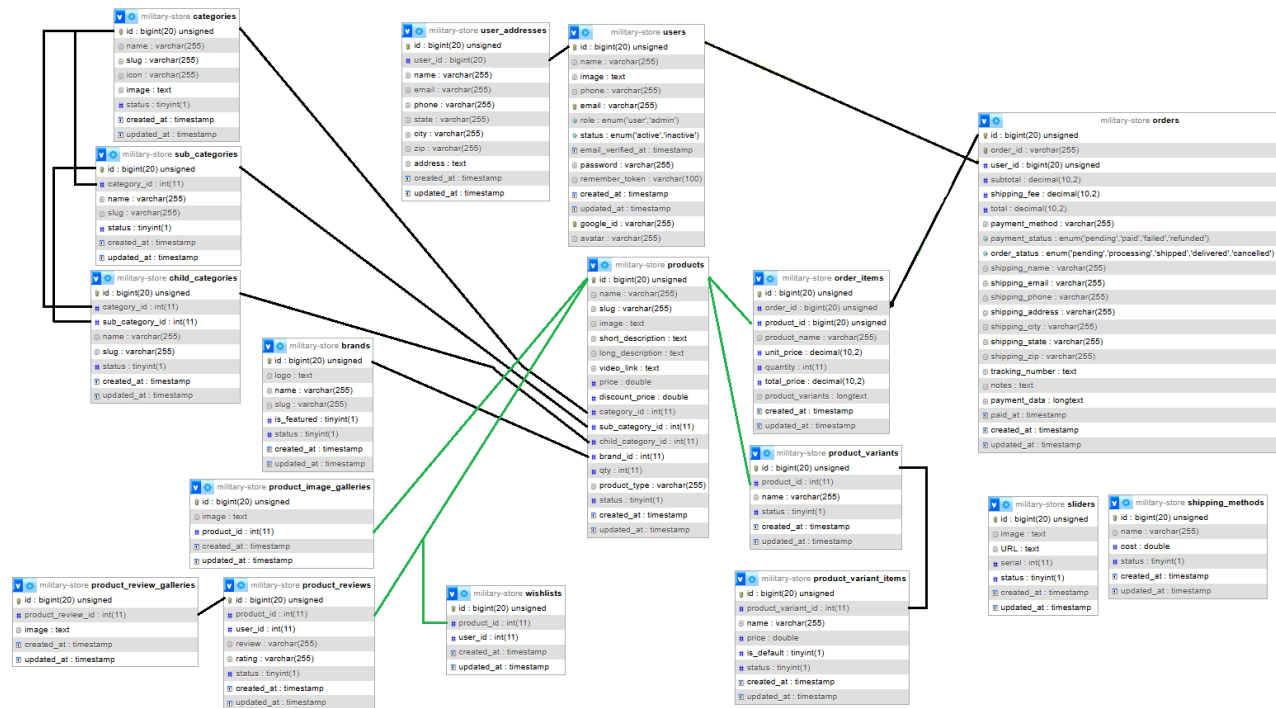


Рис. 2.3. Схема бази даних

Нижче наведено опис таблиць бази даних у таблицях 2.1 – 2.17.

Таблиця 2.1

Опис таблиці Users

Поле	Тип	Призначення
id	bigint	Ідентифікатор
name	string	Ім'я користувача
image	text	Шлях до збереження фото профілю
phone	string	Номер телефону користувача
email	string	Електронна пошта для авторизації
role	enum('user', 'admin')	Роль користувача у системі
status	enum('active', 'inactive')	Статус облікового запису
password	string	Хешований пароль для аутентифікації
remember_token	string	Токен для функції “запам’ятати мене”
google_id	varchar	Ідентифікатор для авторизації через Google

Таблиця Users містить інформацію про зареєстрованих користувачів. Поля включають ім'я, електронну пошту, номер телефону, роль (користувач або адміністратор), статус (активний, неактивний), зображення профілю, пароль тощо. Зв'язки: має багато замовлень, адрес, відгуків, обраних товарів.

Таблиця 2.2

Опис таблиці Categories

Поле	Тип	Призначення
id	bigint	Ідентифікатор

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 2.2

name	string	Ім'я категорії
slug	string	URL посилання
icon	string	Іконка для відображення категорії
status	enum('active', 'inactive')	Статус облікового запису

Таблиця 2.3

Опис таблиці Sub Categories

Поле	Тип	Призначення
id	bigint	Ідентифікатор
category_id	integer	Зовнішній ключ до батьківської категорії
name	string	Ім'я підкатегорії
slug	string	URL посилання
status	enum('active', 'inactive')	Статус облікового запису

Таблиця 2.4

Опис таблиці Child Categories

Поле	Тип	Призначення
id	bigint	Ідентифікатор
category_id	integer	Зовнішній ключ до основної категорії
sub_category_id	integer	Зовнішній ключ до підкатегорії
name	string	Ім'я дочірньої категорії
slug	string	URL посилання

Продовження таблиці 2.4

status	enum('active', 'inactive')	Статус облікового запису
--------	-------------------------------	--------------------------

В таблицях 2.2 – 2.4 описано категорії які реалізують трирівневу структуру каталогу товарів, categories – головні категорії (одяг, спорядження), sub categories – підкатегорії (куртки, штани), child categories – дочірні категорії (зимові куртки, вітровки). Зв'язки: категорія має багато підкатегорій, підкатегорія має багато дочірніх категорій.

Таблиця 2.5

Опис таблиці User Adresses

Поле	Тип	Призначення
id	bigint	Ідентифікатор
user_id	bigint	Зовнішній ключ який зв'язує адресу з користувачем
name	varchar	Ім'я отримувача
phone	varchar	Номер телефону
email	varchar	Електронна пошта
city	varchar	Місто доставки
state	varchar	Область
zip_code	string	Поштовий індекс
address	text	Адреса доставки

Ця таблиця зберігає адреси доставки які користувач може використовувати при оформленні замовлення. Це дозволяє покупцям мати кілька адрес, обирати між ними та швидко обробляти замовлення без повторного введення даних. Зв'язок: один користувач може мати декілька адрес.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.6

Опис таблиці Brands

Поле	Тип	Призначення
id	bigint	Ідентифікатор
logo	text	Шлях до зображення логотипу бренду
name	string	Назва бренду
slug	string	URL посилання
is_featured	boolean	Позначка, чи відображатиметься бренд на головній сторінці
status	enum('active', 'inactive')	Статус облікового запису

Містить інформацію про бренди товарів, їх назву, логотип, статус тощо.
Зв'язок: один бренд має багато товарів.

Таблиця 2.7

Опис таблиці Products

Поле	Тип	Призначення
id	bigint	Ідентифікатор
name	string	Назва товару
slug	string	URL посилання
image	text	Шлях до основного зображення товару
phone	string	Номер телефону користувача
short_description	text	Короткий опис товару
long_description	text	Повний опис товару
video_link	text	Посилання на відео огляд товару
price	double	Основна ціна товару
discount_price	double	Ціна товару зі знижкою

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 2.7

category_id	integer	Зовнішній ключ до категорії
sub_category_id	integer	Зовнішній ключ до підкатегорії
child_category_id	integer	Зовнішній ключ до дочірньої категорії
brand_id	integer	Зовнішній ключ до бренду
qty	integer	Кількість товару на складі
product_type	string	Тип товару (наприклад, топ, рекомендований, новий)
status	boolean	Статус активності товару

Основна таблиця яка зберігає дані про назву, опис, зображення, ціну, кількість, категорії, бренд, статус товару. Зв'язки: належить до категорій, має багато зображень, варіантів, замовлень, відгуків.

Таблиця 2.8

Опис таблиці Product Image Galleries

Поле	Тип	Призначення
id	bigint	Ідентифікатор
image	text	Шлях до додаткового зображення товару
product_id	integer	Зовнішній ключ до товару

Зберігає додаткові зображення товару. Кожен запис містить шлях до зображення та id товару. Зв'язок: належить до товару.

Таблиця 2.9

Опис таблиці Product Variants

Поле	Тип	Призначення
id	bigint	Ідентифікатор
product_id	integer	Зовнішній ключ до товару

Продовження таблиці 2.9

name	string	Назва типу варіанту (наприклад, розмір, колір)
status	enum('active', 'inactive')	Статус облікового запису

Таблиця 2.10

Опис таблиці Product Variant Items

Поле	Тип	Призначення
id	bigint	Ідентифікатор
product_variant_id	integer	Зовнішній ключ до варіанту товару
name	string	Назва варіанту (наприклад, L, Білий)
price	double	Ціна для цього конкретного варіанту товару
is_default	boolean	Позначка, чи є варіант вибраним за замовчуванням
status	boolean	Статус активності елементу варіанту

Таблиці 2.9 – 2.10 використовуються для реалізації опцій товару. Product Variant – тип варіанту (розмір, колір тощо), Product Variant Item – конкретні значення (L, XL, зелений). Зв'язки: один товар має багато варіантів, кожен варіант має багато опцій.

Таблиця 2.11

Опис таблиці Product Reviews

Поле	Тип	Призначення
id	bigint	Ідентифікатор

Продовження таблиці 2.11

product_id	int	Зовнішній ключ до товару
user_id	int	Зовнішній ключ до користувача що залишив відгук
rating	varchar	Оцінка товару
comment	text	Відгук
status	boolean	Статус (адміністратор має модерувати відгуки, по замовчуванню їх не видно)

Містить відгуки користувачів про товари. Зберігає текст коментаря, рейтинг, дату, id користувача та товару. Зв'язки: належить товару і користувачу.

Таблиця 2.12

Опис таблиці Product Review Galleries

Поле	Тип	Призначення
id	bigint	Ідентифікатор
product_review_id	int	Зовнішній ключ до відгука
image	text	Шлях до зображення

Зберігає фото які прикріплені до відгуків. Зв'язок: належить одному відгуку.

Таблиця 2.13

Опис таблиці Orders

Поле	Тип	Призначення
id	bigint	Ідентифікатор
user_id	bigint	Зовнішній ключ до користувача який оформив замовлення

Продовження таблиці 2.13

subtotal	decimal	Сума замовлення без урахування доставки
total	decimal	Загальна сума з урахуванням доставки
shipping_fee	decimal	Вартість доставки
payment_method	varchar	Вибраний спосіб оплати
payment_status	enum('pending', 'paid', 'failed')	Статус оплати
order_status	enum('pending', 'processing', 'shipped', 'delivered', 'canceled')	Статус замовлення
shipping_name	varchar	Ім'я отримувача
shipping_email	varchar	Електронна пошта
shipping_phone	varchar	Номер телефону
shipping_address	varchar	Адреса доставки
shipping_city	varchar	Місто доставки
shipping_state	varchar	Область
shipping_zip	varchar	Поштовий індекс
tracking_number	text	ТТН для відстеження посилки
notes	text	Коментарі до замовлення
payment_data	longtext	Відповідь від платіжної системи
paid_at	timestamp	Час здійснення оплати

Таблиця зберігає інформацію про оформленні замовлення. Вона включає дані про користувача, суму покупки, вибраний спосіб оплати, статус доставки та оплати, контактні та адресні дані для відправлення. Це одна з ключових таблиць, яка пов'язує в собі покупця, товари, метод доставки та результати оплати.

Таблиця 2.14

Опис таблиці Order Items

Поле	Тип	Призначення
id	bigint	Ідентифікатор
order_id	bigint	Зовнішній ключ до замовлення
product_id	bigint	Зовнішній ключ до товарів
product_name	varchar	Назва товару
unit_price	decimal	Ціна за одиницю товару
quantity	int	Кількість
total_price	decimal	Загальна ціна
product_variants	longtext	Варіанти товарів

Зберігає деталі товарів в замовленні, який саме товар був замовлений, яка кількість, ціна, варіанти.

Таблиця 2.15

Опис таблиці Wishlist

Поле	Тип	Призначення
id	bigint	Ідентифікатор
product_id	int	Зовнішній ключ до товарів
user_id	int	Зовнішній ключ до користувача

Таблиця реалізує функціонал списку бажань, зареєстровані користувачі можуть зберігати товари які їм сподобались.

Таблиця 2.16

Опис таблиці Sliders

Поле	Тип	Призначення
id	bigint	Ідентифікатор
image	text	Шлях до зображення
URL	text	Посилання при натисканні

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 2.16

serial	int	Порядок відображення банерів
status	tinyint	Статус відображення

Таблиця відповідає за виведення слайдеру банерів головній сторінці. Містить зображення, посилання, порядковий номер та статус відображення.

Таблиця 2.17

Опис таблиці Shipping Methods

Поле	Тип	Призначення
id	bigint	Ідентифікатор
name	varchar	Назва способу
cost	double	Вартість доставки
status	tinyint	Статус відображення методу

Містить дані про доступні методи доставки які користувач може вибрати при оформленні замовлення.

Кожна операція над сутностями (наприклад, додавання нового товару чи редагування) викликає відповідні методи контролера, які формують запити до бази даних через модель. Завдяки підтримці зовнішніх ключів, дані автоматично пов'язуються між таблицями, забезпечуючи логіку.

Після внесення даних у базу, вони стають доступними для подальшої обробки та відображення на стороні клієнта. Для взаємодії з даними використовуються запити до моделей, що реалізовані у вигляді методів класів. Ці методи дозволяють динамічно отримувати колекції товарів за заданими фільтрами, сортувати, групувати. Таким чином, база даних не лише зберігає інформацію, а й активно бере участь у реалізації логіки роботи інтернет-магазину [26].

2.3. Проектування та реалізація алгоритмів роботи системи

Процес розробки інтернет-магазину військових товарів передбачає побудову логіки взаємодії між користувачем, інтерфейсом і внутрішніми компонентами системи. На рисунку 2.3 представлено UML-діаграму активності, яка демонструє основні етапи взаємодії користувача (покупця) і адміністратора із системою [15].

Спочатку користувач потрапляє на головну сторінку. Далі система перевіряє чи хоче користувач авторизуватися. Якщо так – відкривається форма авторизації. У разі успішного входу або реєстрації користувач отримує доступ до особистого кабінету.

Після авторизації система перевіряє чи є користувач адміністратором. Якщо так – перенаправляє його до адмін панелі де він може керувати товарами, категоріями, замовленнями, а також вийти з панелі. Усі ці дії реалізовано через окремі контролери та доступні тільки при наявності ролі адміністратора.

Якщо користувач звичайний покупець, він має змогу переглядати каталог товарів, шукати продукцію або переходити конкретних категорій. Далі відбувається вибір товару, його перегляд, а також можливість додати товар до кошика. Після додавання користувач може або продовжити покупки, або перейти до кошика. Якщо вибрано оформлення замовлення, система перевіряє чи авторизований користувач. Якщо ні, потрібно зареєструватись або увійти. Далі користувач заповнює контактні дані та адресу для доставки, обирає спосіб оплати і підтверджує замовлення. Система обробляє його, зберігає дані в базі даних та змінює статус замовлення. Наприкінці процесу користувач отримує повідомлення про успішне оформлення, після чого може залишити відгук про товар.

Ця діаграма відображає повний цикл взаємодії користувачів з системою, від першого входу на сайт до написання відгуку про товар. Система перевіряє права доступу, реагує на дії користувача. Це дозволяє забезпечити логіку поведінки інтерфейсу.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

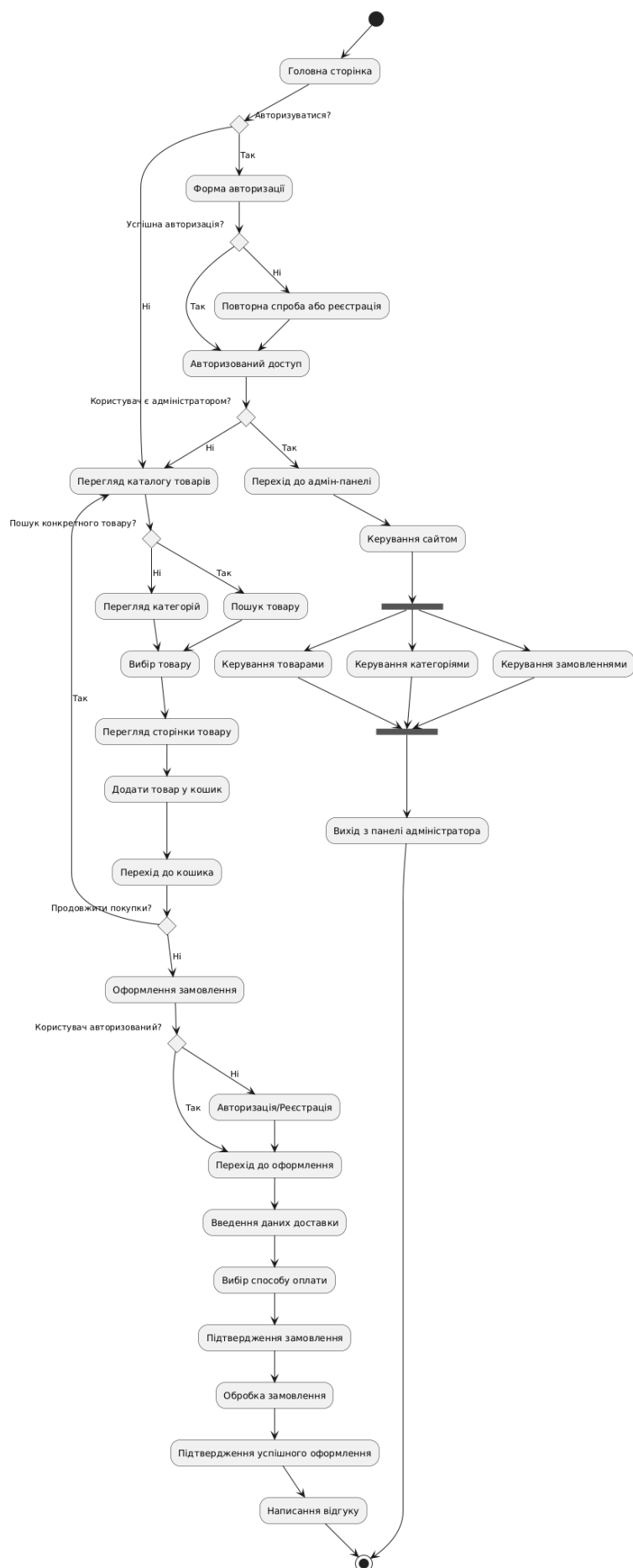


Рис. 2.4. Діаграма активності

Змн.	Арк.	№ докум.	Підпис	Дата

ІПЗ.КР.Б-121-25-ПЗ

Арк.

41

Для представлення процесу оформлення замовлення було створено діаграму послідовностей (рис 2.5) яка демонструє логіку цього процесу. Взаємодія починається з того, що користувач додає товар до кошика. Система перевіряє наявність цього товару, оновлює вміст кошика та відображає актуальну інформацію. Після цього користувач переходить до оформлення, заповнює форму з контактними та адресними даними. Ці дані тимчасово зберігаються в сесії до моменту остаточного підтвердження.

Наступним етапом є обробка платежу. Якщо обрано онлайн оплату, то система створює попереднє замовлення (pending order), перенаправляє користувача на сторінку платіжного сервісу (LiqPay чи WayForPay) і чекає на результат операції. Після завершення платежу платіжна система надсилає callback запит, який обробляється сервером через спеціальний клас. Підпис відповіді перевіряється для забезпечення безпеки. Якщо верифікація пройшла успішно, система створює замовлення в базі даних, зберігає інформацію та очищує сесію й кошик. У випадку накладеного платежу замовлення створюється одразу після підтвердження форми.

Після оформлення замовлення користувач отримує можливість переглядати список своїх замовлень у кабінеті, а також скасовувати їх якщо статус дозволяє це зробити (якщо статус замовлення “відправлено”, скасувати не можна).

Ці алгоритми використовуються для забезпечення безпеки, використання транзакцій у базі даних для їх збереження, а також можливість вибору різних методів оплати. Кожна дія користувача фіксується, що допомагає виявляти помилки. Система створена так, щоб вона була зручною для клієнтом і надійною для адміністраторів. Завдяки розробці на фреймворку Laravel, його модульність дозволяє легко додавати нові функції.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

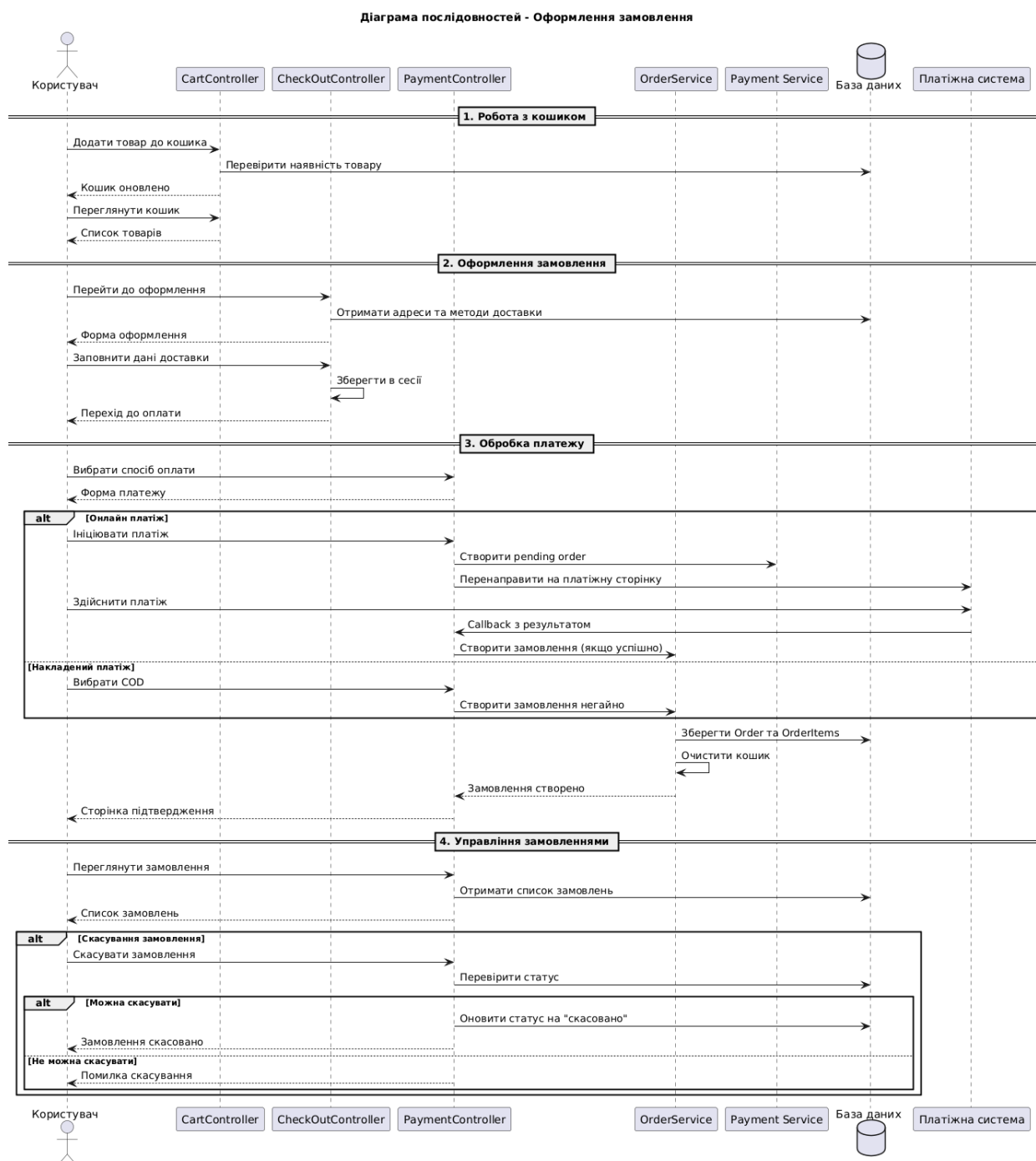


Рис. 2.5. Діаграма послідовності оформлення замовлення

Отже, алгоритми системи чітко поділяють обробку запитів через контролери для забезпечення гнучкості та надійності в обробці та керуванні замовленнями. Створені діаграми чітко визначили основні бізнес-процеси, показали взаємодію між системою та користувачем.

2.4. Реалізація додатку

На основі попередніх рішень було реалізовано інтернет-магазин військових товарів за допомогою фреймворку Laravel. Під час розробки

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

створено логіку взаємодії між інтерфейсом, базою даних і серверною частиною згідно з принципами архітектури MVC (Model-View-Controller).

Для реалізації було використано сучасний стек технологій. Основною мовою програмування є PHP 8.4 з використанням фреймворку Laravel 12. У якості бази даних використано MySQL, а для фронтенду Blade-шаблони у поєднанні з Bootstrap 5, що дозволяє створити адаптивний дизайн та зручний інтерфейс. Керування залежностями здійснюється через Composer, а контроль версій за допомогою Git. Основним середовищем розробки є Visual Studio Code.

Структура проєкту побудована відповідно до принципів MVC. У директорії App зосереджена основна логіка додатку. Зокрема, в `http/controllers` розміщено контролери які поділені на частини для клієнтів і адміністраторів, у `models` - Eloquent моделі що описують сутності бази даних (товари, категорії, користувачі тощо), в `datatables` - спеціальні класи для інтеграції з jQuery Datatables, що забезпечують зручне управління таблицями. У `providers` - сервіс провайдери для реєстрації подій, маршрутів та залежностей, в `traits` - багаторазово використовувані функції (наприклад, завантаження зображень). У `view/` розташовані класи для керування логікою відображення. Конфігурація проєкту зосереджена в директорії `config` яка містить стандартні як стандартні файли (`app.php`, `auth.php`, `database.php`), так і додаткові, наприклад `cart.php` що відповідає за параметри кошика. Файли з маршрутами розміщені в директорії `routes`, де `web.php` містить маршрути для загальнодоступної частини сайту, `admin.php` – для панелі адміністратора, а `auth.php` для аутентифікації та реєстрації. Фронтенд частина знаходиться у `resources/views` та поділяється на дві основні групи: `admin` шаблони для керування сайтом (панель адміністратора) та `frontend` шаблони для публічної частини (головна сторінка, каталог, реєстрація тощо). Директорія `public` містить усі доступні ресурси, у папці `frontend` – стилі, скрипти та зображення клієнтської частини, у `backend` – елементи оформлення адмін-панелі, а в `uploads` користувацькі файли, зображення товарів чи логотипи брендів.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Реалізовано дві основні частини функціоналу. Користувацька частина включає головну сторінку, каталог товарів із фільтрами за категорією, брендом і ціною. Детальну сторінку товару, варіантами та описом. Також кошик з можливістю редагування кількості та сторінку оформлення замовлення й модулі реєстрації та входу. Адміністративна частина передбачає керування товарами, можливість редагування категорій, управління брендами та перегляд і обробку замовлень.

Тепер розглянемо основні етапи реалізації інтернет-магазину. Найбільш складним аспектом реалізації є система реалізації інтеграції платіжних сервісів з підтримкою декількох способів оплати: LiqPay, WayForPay та накладеного платежу [5;11]. Для забезпечення безпеки та гнучкості реалізовано патерн Service-Repository з використанням dependency injection, що дозволило знизити зв'язність компонентів.

Клас PaymentResponseHandler (Додаток А) відповідає за обробку відповідей від платіжних систем. Метод handleCallback реалізує логіку асинхронної обробки callback запитів з верифікацією криптографічного підпису, що захищає від атак типу man-in-middle. Після успішного платежу замовлення тимчасово кешується в Redis [19]. А потім створюється остаточний запис через клас OrderService (Додаток Б) який реалізовує створення замовлення із використанням механізму транзакцій (DB::beginTransaction). Це гарантує що у разі помилки всі зміни скасовуються.

Керування кошиком реалізовано у класі CartController з використанням пакету Gloudemans Shoppingcart (Додаток В). Метод addToCart обробляє варіанти товарів, динамічно розраховує ціну та зберігає опції у сесії користувача. Додатково функціонал AJAX забезпечує оновлення кошика в режимі реального часу без перезавантаження сторінки.

Для ефективної роботи з великим обсягом товарів реалізовано серверний рендер таблиць через ProductDataTable (Додаток Г), що формує запити на основі Eloquent ORM з категоріями, брендами та рейтингами. Інтерфейс

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

підтримує динамічне створення кнопок дій (dropdown) для швидкого управління кожним товаром.

Завантаження зображень реалізовано через трейт ImageUploadTrait (Додаток Г), який містить методи uploadImage, uploadMultiImage, updateImage. Усі зображення отримують унікальні імена, а старі автоматично видаляються.

Фільтрація товарів реалізована в FrontendProductController через методи applyPriceFilter, applyRatingFilter, applySorting (Додаток Д). Ці методи враховують знижку, рейтинги та інші параметри фільтрації. Пошук реалізовано через індексацію по назві, опису та категоріях із використанням LIKE і whereHas.

Система управління відгуками (ReviewController) підтримує галереї зображень, їх модерацію та зміну статусу (Додаток Ж). Зображення зберігаються через модель ProductReviewGallery.

Усі платіжні сервіси реєструються у AppServiceProvider (Додаток И) через методи register і boot, що забезпечує загальну конфігурацію та інтеграцію з іншими компонентами.

Модель Product (Додаток К) містить зв'язки hasMany та belongsTo з іншими сутностями, а також реалізує scope-методи для фільтрації товарів за статусом, типом чи знижкою. Ці методи забезпечують ефективну побудову запитів та підтримку бізнес-логіки.

Технічна реалізація показала ефективність обраного стеку технологій. Здійснено чітке розділення між клієнтською, серверною логікою та базою даних, що значно спрощує тестування.

Висновки до другого розділу

У другому розділі було здійснено проєктування та реалізацію інтернет-магазину військових товарів. На основі функціональних вимог було визначено основні сценарії використання системи для різних типів користувачів (покупця та адміністратора), розроблено UML діаграму активності, яка показує логіку взаємодії з додатком, а також побудовано структуру бази даних.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

У процесі реалізації було застосовано архітектуру MVC, притаманну фреймворку Laravel, що забезпечило чіткий поділ відповідальностей між компонентами системи: Controllers, Models, Views. Для управління даними використовувалась ORM-система Eloquent, що значно спростило роботу з базою даних MySQL і дозволила ефективно реалізувати зв'язки між таблицями та фільтрацію.

Реалізовано ключові функціональні модулі, такі як особистий кабінет, роботу з товарами, кошиком, оформленням замовлення та адміністративну панель. Окрему увагу приділено фільтрації, пошуку, варіантам товарів і керванню контентом. Результатом виконання розділу стала реалізація програмної логіки веб додатку, яка готова до подальшого тестування.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБ-ДОДАТКОМ ІНТЕРНЕТ-МАГАЗИНУ ВІЙСЬКОВИХ ТОВАРІВ

3.1. Порядок встановлення та налаштування параметрів системи

Для запуску веб-додатку інтернет-магазину військових товарів, потрібно налаштувати локальне серверне середовище. Система базується на фреймворку Laravel, знадобляться такі інструменти як PHP версії 8.0 або вище, Composer для управління пакетами, MySQL для бази даних, а також Node.js з npm.

Першим кроком є встановлення середовища розробки. Рекомендовано використовувати XAMPP або аналоги які мають сервер Apache, інтерпретатор PHP та систему управління базами даних MySQL. Для розробки на фреймворку Laravel також необхідно встановити Composer – менеджер залежностей PHP, Node.js з менеджером пакетів npm, а також середовище розробки, наприклад Visual Studio Code.

Після встановлення середовища необхідно завантажити проєкт або клонувати репозиторій у папку веб-сервера. Далі слід виконати інсталяцію залежностей за допомогою команди `composer install`. Файл `.env` є головним для зберігання змінних середовища. У ньому вказується конфігурація підключення до бази даних, платіжних системи (ключі LiqPay, WayForPay), параметри Google OAuth. Після створення файлу треба виконати команду `php artisan key:generate` для генерації унікального ключа шифрування сесій.

Структура бази даних створюється виконанням міграцій (`php artisan migrate`), а початкові дані (наприклад, категорії, товари, бренди) додаються адміністратором вручну після запуску сайту. Для роботи інтерфейсу також необхідна компіляція ресурсів за допомогою `npm install` та `npm run dev`.

Після завершення усіх підготовчих дій додаток запускається командою `php artisan serve`, після цього на веб-сайт можна перейти за адресою <http://127.0.0.1:8000>. Перший вхід в панель адміністратора може буде

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

здійснений за допомогою облікового запису адміністратора, який створюється окремим сидером, що запускається командою `php artisan db:seed`.

На рисунку 3.1 зображено основні логічні компоненти системи та зв'язки між ними. Архітектура побудована на фреймворку Laravel, який виступає базою для Backend і Frontend компонентів. Компонент Backend керує моделями, що реалізовані через Eloquent ORM, яка взаємодіє з базою даних.

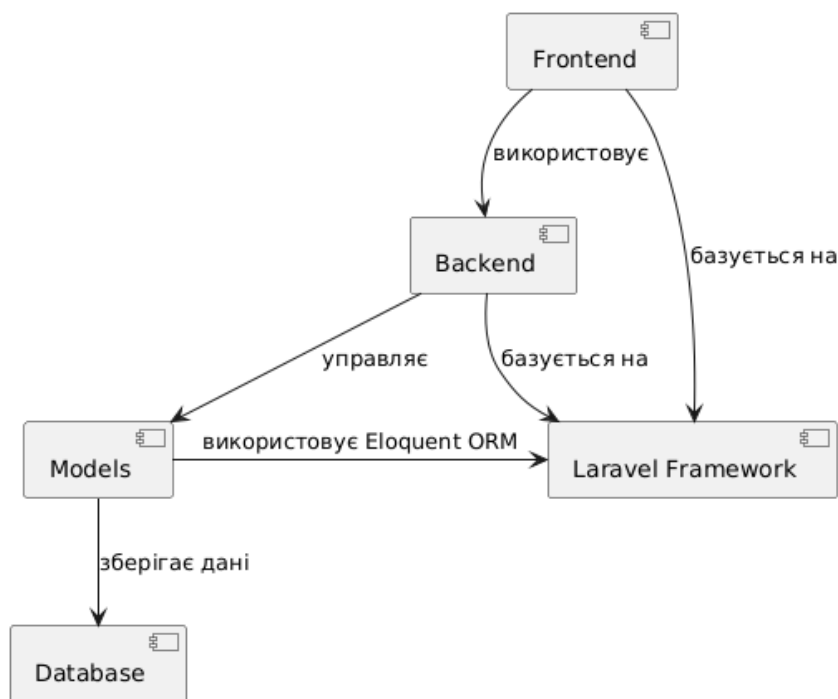


Рис. 3.1. Діаграма компонентів

Опис компонентів:

Frontend - клієнтська частина, що використовує API бекенду та візуалізує дані для користувача.

Backend – логіка обробки запитів, управління даними та маршрутизація.

Laravel Framework – основа для побудови структури контролерів, моделей та маршрутів.

Models – об'єкти які описують таблиці бази даних.

Database – реляційна система управління базами даних (MySQL), яка зберігає всі дані про товари, користувачів, замовлення тощо.

Таким чином, після правильного встановлення та налаштування система

готова до тестування або розгортання на хостингу.

3.2. Структура інтерфейсу веб-додатку

Інтерфейс інтернет-магазину військових товарів розроблений з урахуванням зручної взаємодії користувача з основними елементами системи. Він розділений на дві основні частини: публічну (доступну всім відвідувачам та користувачам сайту) та адміністративну (доступну лише авторизованим адміністраторам).

Публічна частина починається з головної сторінки (рис 3.2, рис 3.3), де користувач бачить слайдер з банерами, категоріями, новинки товарів, а також брендів магазину.

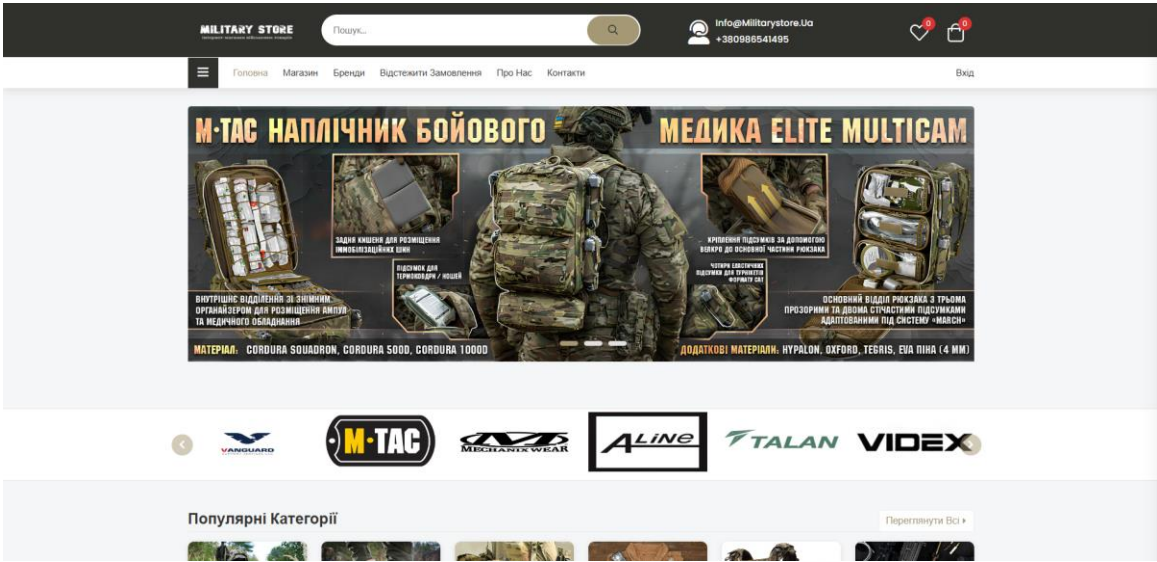


Рис. 3.2. Головна сторінка

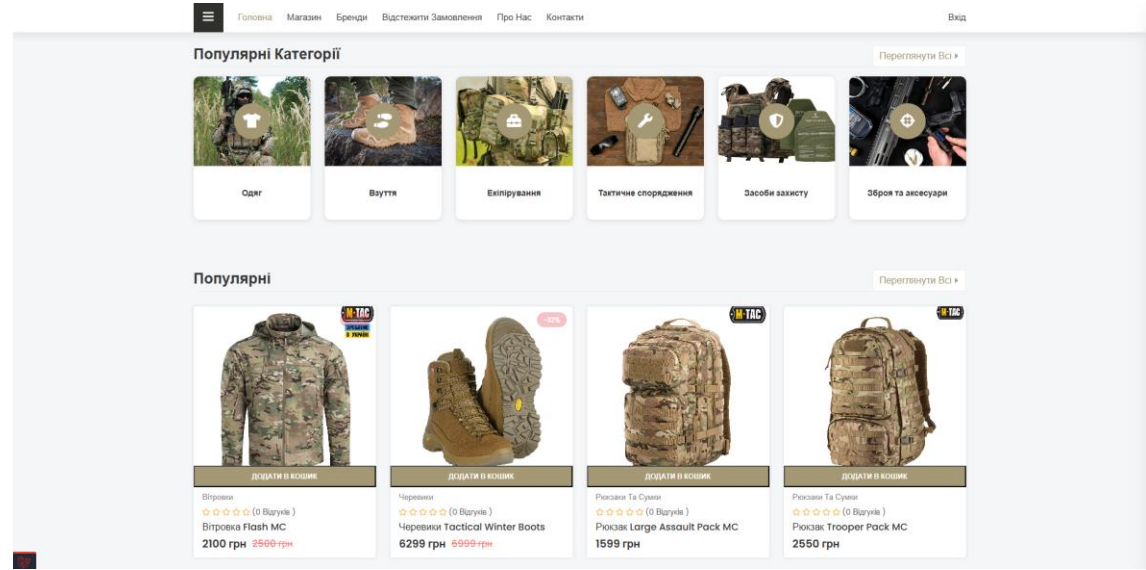


Рис. 3.3. Головна сторінка

Головна сторінка виконує функцію навігаційного хаба, з якого можна перейти до окремої категорії товарів. Каталог реалізований у вигляді плитки з товарами, кожен з яких відображає зображення, назву, ціну та рейтинг з кількістю відгуків. На рисунку 3.4 зображено випадаюче меню категорій товарів. Меню підтримує вкладену структуру що дозволяє користувачу швидко знайти необхідний товар.

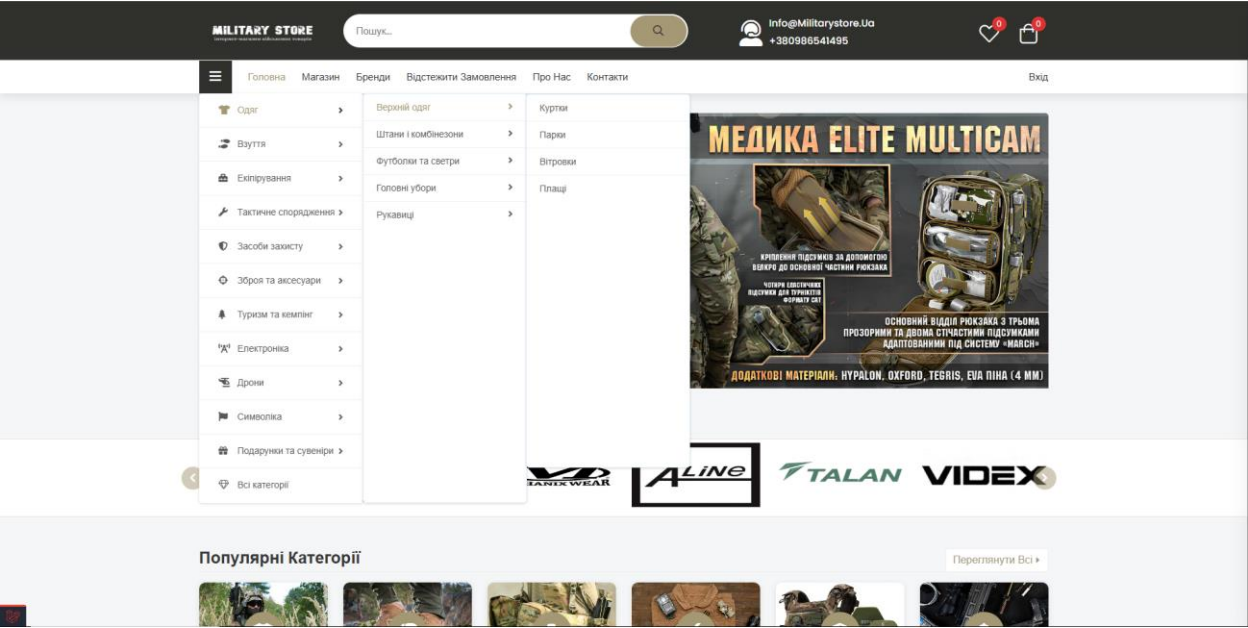


Рис. 3.4. Меню категорій товарів

У верхній частині сторінки доступні елементи навігації – пошук, вхід, кошик, улюблені товари, контактна інформація та інші. При переході до картки товару (рис. 3.5) відкривається окрема сторінка, де виведено повну інформацію про товар, включаючи галерею зображень, опис, варіанти (наприклад, розмір), а також кнопку додавання товару у кошик. В нижній частині сторінки є кнопки з можливістю перегляду детального опису товару, таблиці розмірів (для одягу) та відгуки.

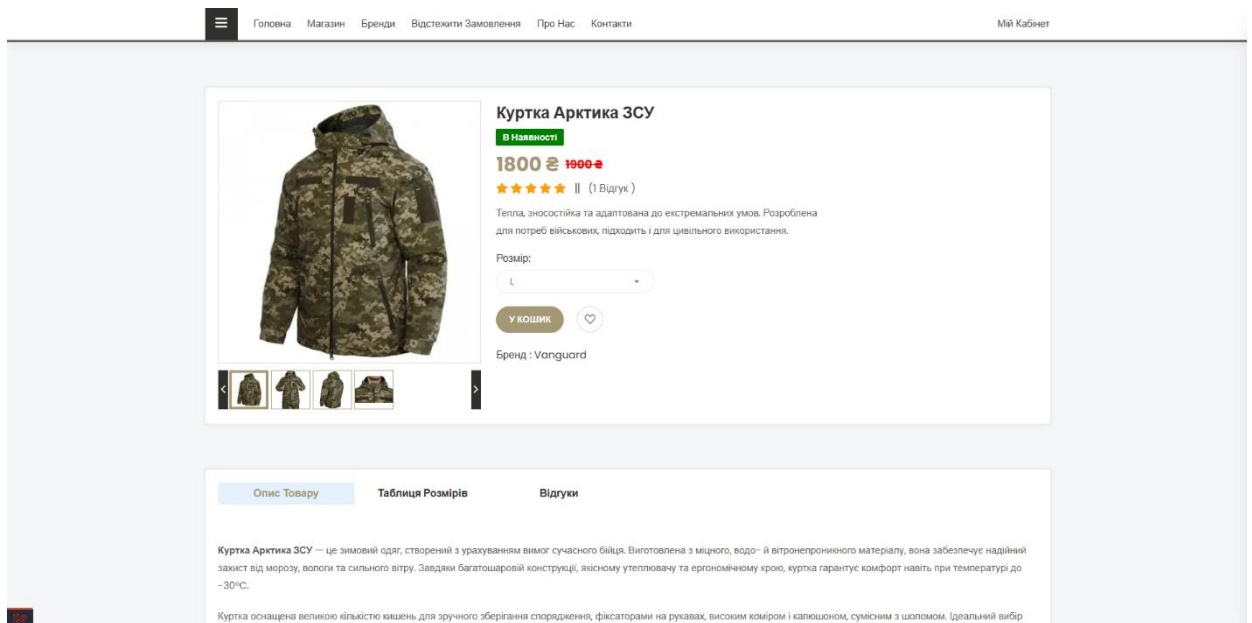


Рис. 3.5. Сторінка товару

При натисканні на іконку кошика з правого боку відкривається меню де показано додані товари та кнопки детального перегляду та оформлення замовлення. При переході на сторінку перегляду кошика (рис. 3.6) показується перелік доданих товарів, ціна, наявність. На цій сторінці є можливість змінювати кількість одиниць товару, видаляти позиції або переходити до оформлення замовлення.

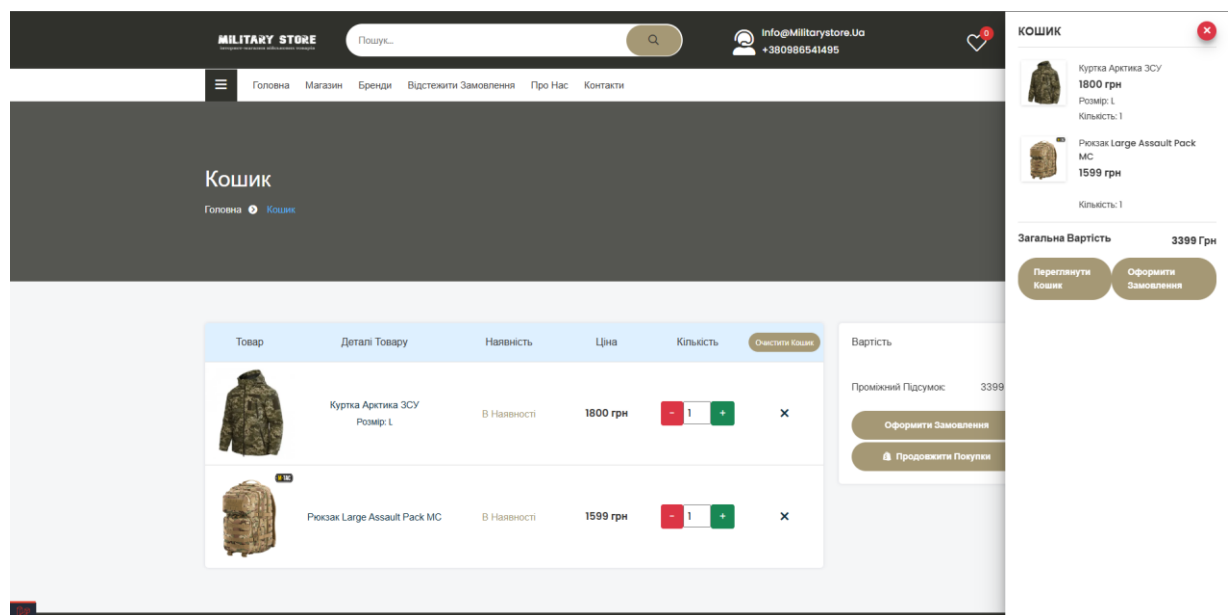


Рис. 3.6. Сторінка та бічне меню кошика

Після натискання кнопки “Оформити замовлення” не авторизований користувач потрапляє на сторінку входу в акаунт (рис 3.7), де вводяться дані які були використані при реєстрації акаунту, також є можливість увійти за допомогою Google. Зареєструвати акаунт (рис 3.8) можна на тій самій сторінці.

Рис. 3.7. Сторінка авторизації

Рис. 3.8. Сторінка реєстрації

Після входу в акаунт користувач потрапляє на сторінку оформлення замовлення, де потрібно обрати або створити контактні дані для доставки, після чого обрати спосіб доставки.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Головна Магазин Бренди Відстежити Замовлення Про Нас Контакти Мій Кабінет

Оформлення Замовлення

Головна • Оформлення Замовлення

Доставка

Додати Нову Адресу

Вибрати Адресу

Ім'я: Максим Бардан
Телефон: +380988421280
Email: Maxbar@gmail.com
Місто: Київ
Поштовий Індекс: 03131
Адреса: Столице 30

Доставка

☒ Самовівез з Нової Пошти
Вартість: 139 грн

☐ Самовівез з УкрПошти
Вартість: 89 грн

☐ Самовівез з Meest ПОШТА
Вартість: 99 грн

Вартість Товарів: 3399 грн
Вартість Доставка: 139 грн
Загалом: 3538 грн

☒ Я Прочитав і погоджуюсь з Умовами Та Правилами Веб-Сайту*

Оформити Замовлення

Рис. 3.9. Сторінка оформлення замовлення

При натисканні кнопки “Оформити замовлення” користувач потрапляє на сторінку оплати (рис. 3.10). Доступно три способи оплати: LiqPay, WayForPay, накладений платіж.

MILITARY STORE Пошук... Info@Militarystore.Ua +380988541495

Головна Магазин Бренди Відстежити Замовлення Про Нас Контакти Мій Кабінет

Оплата

Головна • Оплата

LiqPay

WayForPay

Накладений Платіж

Оплата через LiqPay

Деталі Оплати: Сума До Оплати: 3538 грн

Спосіб Оплати: LiqPay

Після натискання кнопки "Оплатити" ви будете перенаправлені на безпечну сторінку LiqPay для завершення платежу.

Оплатити Через LiqPay

Вартість замовлення

Вартість Товарів: 3399 грн
Вартість Доставка: 139 грн
Загально: 3538 грн

Рис. 3.10. Сторінка оплати

На рисунку 3.11 показано сторінку успішної LiqPay оплати замовлення.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

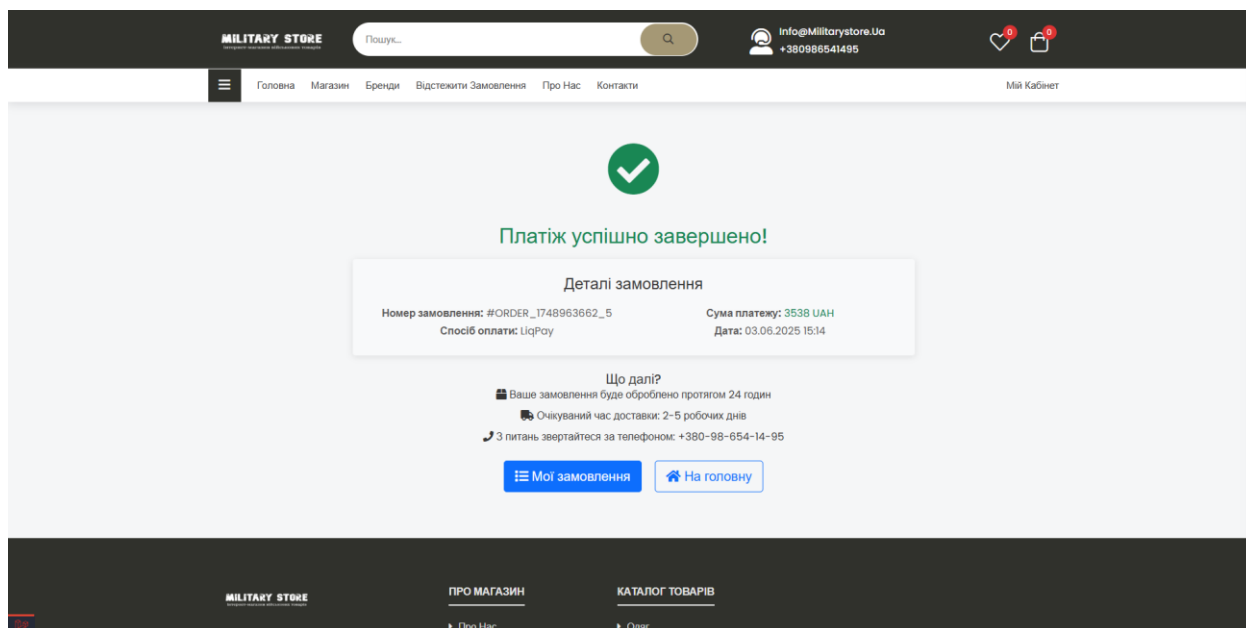


Рис. 3.11. Сторінка успішної оплати

Для зареєстрованих користувачів доступний особистий кабінет (рис. 3.12), в якому можна переглядати історію замовлень, відгуки, та редагувати персональні дані. У центральній частині сторінки знаходиться форма для редагування особистих даних: ім'я, телефон, email, а також можливість завантаження аватарки. Нижче реалізовано функціонал зміни пароля, користувач повинен ввести поточний пароль, новий та підтвердити його.

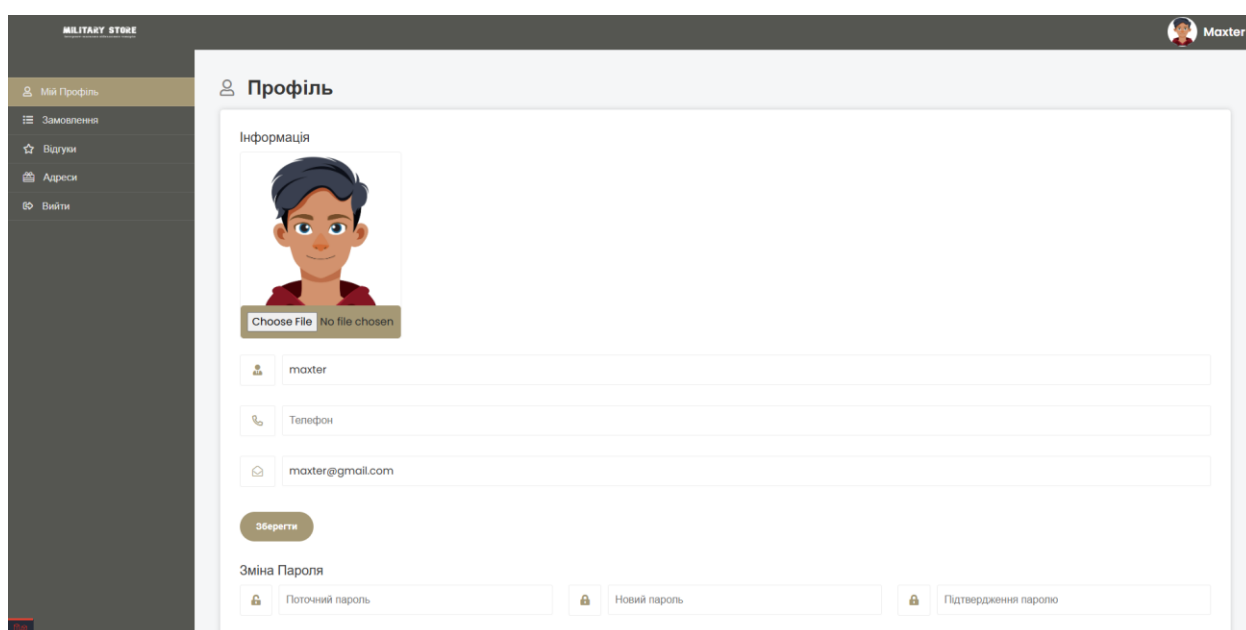


Рис. 3.12. Інтерфейс особистого кабінету користувача

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

На рисунку 3.13 зображено сторінку перегляду замовлень користувача. У цьому розділі відображаються всі підтвердженні замовлення з інформацією про них: номер замовлення, дата оформлення, загальна сума, спосіб оплати, статус замовлення і оплати. Кожне замовлення можна переглянути детальніше або скасувати, якщо воно ще не відправлене.



Рис. 3.13. Інтерфейс розділу “Мої замовлення” в особистому кабінеті

Рисунок 3.14 демонструє інтерфейс з детальною інформацією про замовлення. Тут виведено адресу доставки, спосіб оплати, список товарів із ціною, кількістю, варіантами, зображенням, а також сумарну вартість. В нижній частині сторінки користувач має змогу скасувати замовлення (за умови, що воно ще не відправлене).

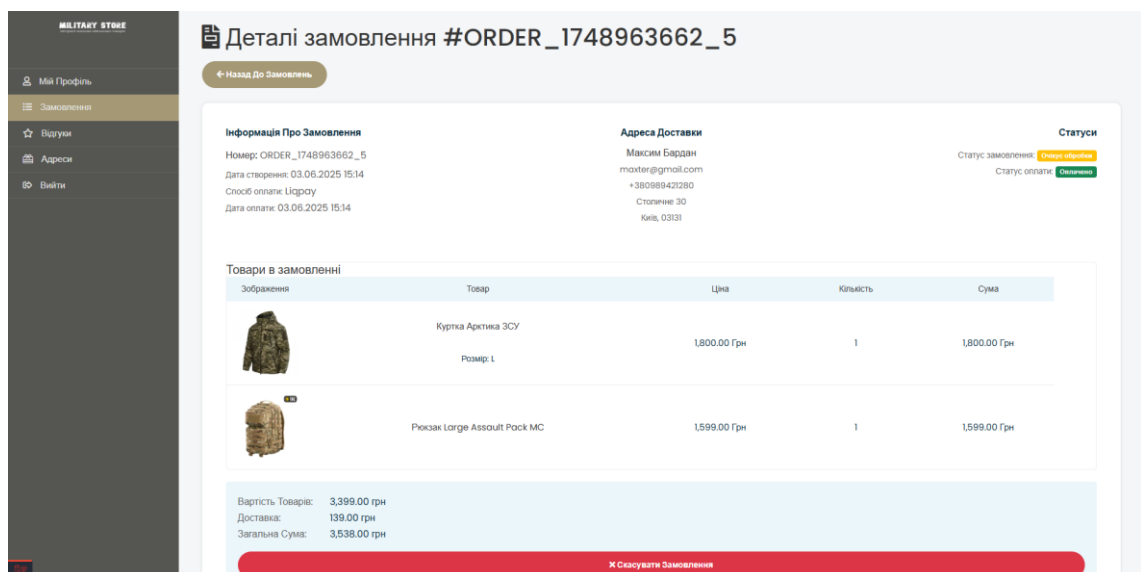


Рис. 3.14. Деталі замовлення

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

На рисунку 3.15 зображено інтерфейс адміністративної панелі після авторизації адміністратора. В загальній панелі виводиться загальна статистика: кількість замовлень, успішно виконані замовлення, відгуки, категорії, бренди та зареєстровані користувачі.

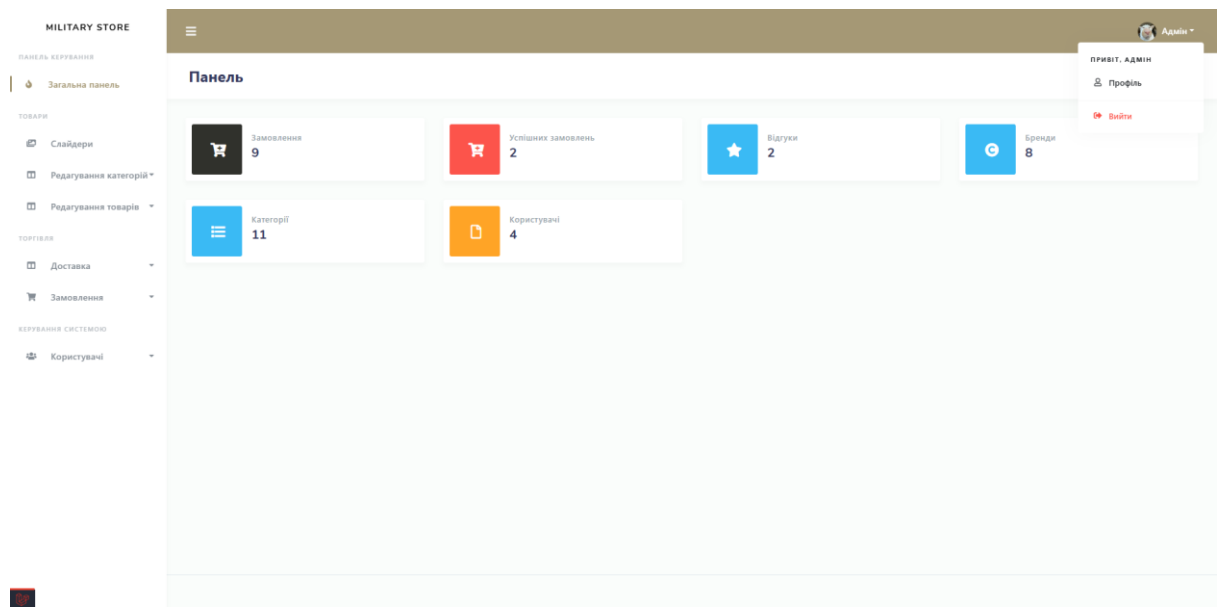


Рис. 3.15. Головна сторінка панелі адміністратора

На рисунку 3.16 представлено приклад сторінки для керування категоріями. Таблиця відображає інформацію про категорії, статус активності, а також кнопку редагування чи видалення кожної категорії. У правому верхньому куті розміщена кнопка створення нової категорії, що відкриває форму для додавання даних.

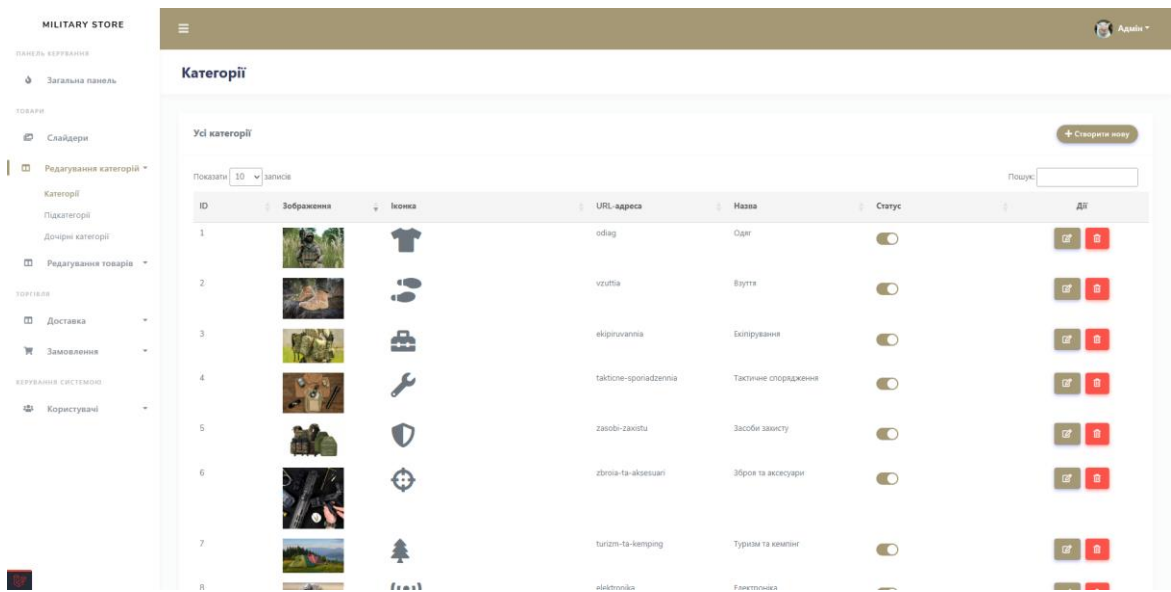


Рис. 3.16. Інтерфейс керування категоріями товарів

3.3. Тестування функціоналу програмного комплексу

Після завершення розробки програмного забезпечення було проведено тестування основного функціоналу системи. Метою тестування є перевірка системи на відповідність технічним вимогам, виявлення помилок та підтвердження коректності роботи основних модулів [28]. Тестування здійснювалося вручну, з використанням типових сценаріїв роботи користувача (покупця) та адміністратора. У процесі перевірки оцінювалися стабільність системи, правильність виконання бізнес-логіки, відображення інтерфейсу, обробка форм і взаємодія з базою даних.

З боку покупця були протестовані такі дії: реєстрація нового користувача, авторизація, перегляд каталогу товарів, фільтрація та пошук, перегляд картки товару, вибір варіантів (розмір, колір), додавання до кошика, редагування кількості товарів у кошику, оформлення замовлення з введенням контактних даних, а також доступ до особистого кабінету для перегляду історії замовлень і редагування профілю.

З боку адміністратора перевірялося: вхід до адміністративної панелі, додавання, редагування та видалення товарів, створення категорій та брендів, завантаження зображень, керування замовленнями та зміна їх статусів. Особлива увага приділялася перевірці обмеження доступу до адмін-панелі, неавторизовані користувачі були успішно перенаправлені на сторінку входу.

Було використано 10 тестових сценаріїв. Результати занесено у таблицю 3.1.

Таблиця 3.1

Результати тестування функціональних модулів

№	Сценарій	Очікуваний результат	Фактичний результат	Статус
1	Реєстрація користувача	Створено нового користувача	Відповідає	Пройдено
2	Авторизація	Перехід до особистого кабінету	Відповідає	Пройдено

Продовження таблиці 3.1

3	Перегляд товарів	Відображено список товарів	Відповідає	Пройдено
4	Фільтрація за категорією	Показано товари лише вибраної категорії	Відповідає	Пройдено
5	Додавання в кошик	Товар з обраними параметрами збережено у кошику	Відповідає	Пройдено
6	Оформлення замовлення	Запис замовлення збережено в базі даних	Відповідає	Пройдено
7	Вхід адміністратора	Доступ до адмін-панелі відкрито	Відповідає	Пройдено
8	Додавання нового товару	Товар збережено у каталозі	Відповідає	Пройдено
9	Видалення товару	Товар видалено з бази даних та списку	Відповідає	Пройдено
10	Зміна статусу замовлення	Статус замовлення збережено та виведено	Відповідає	Пройдено

Для виведення помилок було використано бібліотеку Toastr. На рисунку 3.17 можна побачити валідацію введених даних при створенні нового товару.

The screenshot shows a web application interface for creating a new product. The form is titled "Товар" (Product) and "Створити товар" (Create product). It includes several input fields: "Фотографія" (Image) with a "Choose File" button, "Назва" (Name), "Категорія" (Category), "Підкатегорія" (Subcategory), "Дочірна категорія" (Subcategory), "Бренд" (Brand), and "Ціна" (Price). On the right side of the form, there are five red error messages indicating validation failures: "The long description field is required.", "The short description field is required.", "The price field is required.", "The name field is required.", and "The image field is required."

Рис. 3.17. Валідація введених даних

Загалом у ході тестування також було проведено оцінку поведінки системи при одночасному виконанні кількох користувацьких дій, що

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

дозволило оцінити її стабільність під навантаженням. Жодних критичних помилок або збоїв не було виявлено.

Проведене тестування підтвердило працездатність і надійність розробленої системи. Усі модулі функціонують згідно з поставленими вимогами, інтерфейси адаптовані для різних ролей користувачів, а обробка запитів до бази даних відбувається без помилок.

Висновки до третього розділу

У третьому розділі було проведено аналіз інтерфейсу та функціональних особливостей розробленого веб-додатку інтернет-магазину військових товарів. Розглянуто порядок встановлення та налаштування параметрів системи, зокрема залежності середовища, конфігураційні файли, базу даних і запуск локального сервера.

Описано структуру інтерфейсу з точки зору двох основних ролей: користувача (покупця) та адміністратора. Було проаналізовано роботу таких елементів, як головна сторінка, каталог товарів, картка товару, кошик, особистий кабінет користувача, а також адміністративна панель для керування контентом і замовленнями.

В межах розділу проведено ручне тестування програмного комплексу за ключовими сценаріями взаємодії з системою. Результати тестування показали, що основні функції веб-додатку працюють коректно та відповідають технічним вимогам. Усі протестовані модулі виконують поставлені завдання без критичних помилок, а інтерфейс залишається стабільним і зручним для взаємодії.

Таким чином, на основі аналізу встановлення, структури та тестування, можна зробити висновок про успішну реалізацію розробленої системи. Система придатна до використання, має інтуїтивно зрозумілий інтерфейс, належний функціонал та забезпечує надійну взаємодію з користувачами.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У межах бакалаврської кваліфікаційної роботи було розроблено веб-додаток інтернет-магазину військових товарів. Результатом виконання роботи стала побудова сучасного веб-ресурсу, який дозволяє здійснювати повний цикл взаємодії між продавцем і покупцем, від ознайомлення з продукцією до оформлення замовлення та обробки його в адміністративній панелі.

На етапі теоретичного аналізу було досліджено особливості функціонування інтернет-магазинів, здійснено порівняльний огляд існуючих рішень, виділено їх переваги та недоліки. Визначено архітектуру розроблюваної системи, розроблено діаграми варіантів використання, активності та компонентів, що показують основну логіку роботи додатку.

У рамках реалізації програмного забезпечення створено багатофункціональний веб-додаток з двома рівнями доступу: для користувача (покупця) та адміністратора. Система побудована на основі фреймворку Laravel, з використанням технологій PHP, MySQL, HTML, CSS, JavaScript, що забезпечило високу масштабованість, надійність та зручність. Реалізовано каталог товарів з багаторівневою структурою категорій, систему варіантів (розмір, колір), кошик, реєстрацію/авторизацію, оформлення замовлення та інтерфейс адміністративного керування контентом.

Після завершення розробки було здійснено тестування ключових функціональних модулів, результати якого підтвердили стабільну та коректну роботу системи відповідно до технічного завдання. Виявлені під час розробки неточності були усунені, а всі основні сценарії взаємодії із системою виконано без збоїв.

Розроблений веб-додаток відповідає поставленим цілям і завданням, а його функціонал охоплює всі необхідні аспекти взаємодії між користувачем і системою. Завдяки модульній структурі та використанню сучасного технологічного стеку проєкт легко масштабувати та оновлювати.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abrams [Електронний ресурс] – Режим доступу до ресурсу: <https://www.abrams.com.ua/>
2. Bootstrap 5 Документація [Електронний ресурс] – Режим доступу до ресурсу: <https://bootstrap21.org/uk/docs/5.0/>
3. Composer Документація [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ukraine.com.ua/wiki/hosting/php/composer/how-to/>
4. Git, GitLab та Github: особливості систем контролю версій та їх відмінності [Електронний ресурс] – Режим доступу до ресурсу: <https://pnn.com.ua/ua/blog/detail/git-gitlab-and-github-difference-and-peculiarities-of-version-control-systems>
5. LiqPay API Документація [Електронний ресурс] – Режим доступу до ресурсу: <https://www.liqpay.ua/doc/api>
6. Microsoft Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio
7. Militaria.pl [Електронний ресурс] – Режим доступу до ресурсу: <https://militaria.pl/ua>
8. Militarist [Електронний ресурс] – Режим доступу до ресурсу: <https://militarist.ua/ua/>
9. Punisher [Електронний ресурс] – Режим доступу до ресурсу: <https://punisher.com.ua/>
10. Velmet [Електронний ресурс] – Режим доступу до ресурсу: <https://velmet.ua/>
11. WayForPay API Документація [Електронний ресурс] – Режим доступу до ресурсу: <https://wiki.wayforpay.com/uk>
12. Архітектура програмного забезпечення: як правильне проєктування забезпечує ефективність та безпеку [Електронний ресурс] – Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/arhitektura-programnogo-zabezpechennya>

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

13. База даних MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://promoter.net.ua/articles/baza-danix-mysql.html>

14. Гонтар Л. Вплив електронної комерції на сучасні тенденції розвитку ринку збуту в умовах воєнного стану // Науковий вісник Ужгородського національного університету. Серія: Економіка. – 2024 – Вип. 5(23) – С. 59–70 – DOI: [https://doi.org/10.52058/2786-6300-2024-5\(23\)-59-70](https://doi.org/10.52058/2786-6300-2024-5(23)-59-70)

15. Діаграма активності UML – Unified Modeling Language (UML) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mindonmap.com/uk/blog/uml-activity-diagram/>

16. Діаграма варіантів використання UML [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mindonmap.com/uk/blog/what-is-a-uml-use-case-diagram/>

17. Діаграми класів [Електронний ресурс] – Режим доступу до ресурсу: <https://ua5.org/oop/392-diagrami-klasiv.html>

18. Дубель М., Барвінченко О. Особливості розвитку електронної комерції в Україні в умовах війни // Підприємництво та інновації. – 2023. – №27. – С. 16-22. DOI: 10.32782/2415-3583/27.2

19. Загальна інформація про Redis [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ukraine.com.ua/wiki/redis/overview/>

20. Кращі практики безпеки PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ranktracker.com/uk/blog/best-php-security-practices-top-tips-to-safeguard-your-web-applications/>

21. Найкращі PHP фреймворки [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hostzealot.com.ua/blog/about-vps/najkrashi-php-frejmworki-dlya-veb-rozrobnikiv>

22. Переваги використання фреймворку Laravel [Електронний ресурс] – Режим доступу до ресурсу: <https://xn--80ahcnklsjod4a9r.xn--j1amh/articles/yaki-perevagi-vikoristannya-frejmworku-laravel>

23. Переваги фреймворку Laravel [Електронний ресурс] – Режим доступу до ресурсу: <https://brander.ua/blog/perevahy-freymworku-laravel>

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

24. Побудова динамічних вебзастосунків за допомогою MVC [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.ithillel.ua/articles/building-dynamic-web-applications-using-mvc>

25. Посібник з PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/uk/index.php>

26. Проектування реляційних баз даних [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.ithillel.ua/articles/relational-database-design-guide>

27. Розробка інтернет магазину – основні характеристики та функціональність [Електронний ресурс] – Режим доступу до ресурсу: <https://web24.pro/rozrobka-sajtiv-blog/rozrobka-internet-magazyn-osnovni-harakterystyky-ta-funkczionalnist/>

28. Тестування інтерфейсу користувача [Електронний ресурс] – Режим доступу до ресурсу: <https://terentevdesignstudio.com/blog/testuvannya-ui/>

29. Тренди та виклики українського ринку eCommerce у 2024 році [Електронний ресурс] – Режим доступу до ресурсу: <https://uaateam.agency/blog/trendy-ta-vyklyky-ukrainskogo-rynku-ecommerce/>

30. Функціональні та нефункціональні вимоги [Електронний ресурс] – Режим доступу до ресурсу: <https://www.guru99.com/uk/functional-vs-non-functional-requirements.html>

31. Що таке Laravel та які його можливості [Електронний ресурс] – Режим доступу до ресурсу: <https://cityhost.ua/uk/blog/scho-take-laravel-ta-yaki-yogo-mozhливosti.html>

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

class PaymentResponseHandler
{
    private LiqPayService $liqPayService;
    private WayForPayService $wayForPayService;
    private OrderService $orderService;
    public function __construct(LiqPayService $liqPayService,
WayForPayService $wayForPayService, OrderService $orderService)
    {
        $this->liqPayService = $liqPayService;
        $this->wayForPayService = $wayForPayService;
        $this->orderService = $orderService;
    }
    /*** Обробка успішної відповіді*/
    public function handleSuccessResponse(?string $data, ?string $signature):
View\RedirectResponse
    {
        // Якщо дані не передані, перевіряємо pending_order в сесії
        if (!$data || !$signature) {
            return $this->handleSuccessWithoutData();
        }
        Log::info('Processing LiqPay success with data', [
            'data_length' => strlen($data),
            'signature_length' => strlen($signature)
        ]);
        // Перевіряємо підпис
        if (!$this->liqPayService->verifySignature($data, $signature)) {
            Log::warning('LiqPay signature verification failed');
            return redirect()->route('user.orders')->with('error', 'Помилка
верифікації платежу');

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    // Декодуємо дані
    $paymentData = $this->liqPayService->decodePaymentData($data);
    if (!$paymentData) {
        return redirect()->route('user.orders')->with('error', 'Помилка обробки
даних платежу');
    }
    Log::info('LiqPay payment data decoded successfully', $paymentData);
    if ($paymentData['status'] === 'success') {
        return $this->processSuccessfulPayment($paymentData);
    } else {
        Log::warning('LiqPay payment not successful', [
            'status' => $paymentData['status'] ?? 'unknown',
            'payment_data' => $paymentData
        ]);
        return redirect()->route('user.payment')->with('error', 'Платіж не був
завершений. Статус: ' . ($paymentData['status'] ?? 'невідомий'));
    }
}

/** Обробка успішної відповіді без даних */
private function handleSuccessWithoutData(): View|RedirectResponse
{
    Log::warning('LiqPay success called without data or signature');
    $pendingOrder = $this->liqPayService->getPendingOrder();
    if ($pendingOrder) {
        Log::info('Found pending order in session', ['order_id' =>
$pendingOrder['order_id']]);
        // Створюємо замовлення з даних сесії
        $paymentDataArr = [
            'order_id' => $pendingOrder['order_id'],

```

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'status' => 'success',
        'amount' => $pendingOrder['amount'],
        'currency' => 'UAH',
        'payment_id' => 'manual_' . time()
    ];

    $order = $this->orderService-
>createOrderFromPayment($paymentDataArr, $pendingOrder);

    return view('frontend.pages.payment-success', [
        'order_id' => $pendingOrder['order_id'],
        'amount' => $pendingOrder['amount'],
        'payment_method' => 'LiqPay',
        'note' => $order ? null : 'Ваш платіж обробляється. Деталі будуть
оновлені найближчим часом.'
    ]);
}

return redirect()->route('user.orders')->with('success', 'Платіж успішно
оброблено!');
}

/** Обробка успішного платежу */

private function processSuccessfulPayment(array $paymentData):
View|RedirectResponse
{
    $orderId = $paymentData['order_id'] ?? null;
    if (!$orderId) {
        Log::warning('Order ID not found in payment data', $paymentData);
        return redirect()->route('user.orders')->with('error', 'Помилка
ідентифікації замовлення');
    }

    // Перевіряємо чи замовлення вже існує в базі даних

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$existingOrder = Order::where('order_id', $orderId)->first();
if ($existingOrder) {
    Log::info('Order already exists, redirecting to success page', ['order_id'
=> $orderId]);
    return view('frontend.pages.payment-success', [
        'order_id' => $orderId,
        'amount' => $existingOrder->total,
        'payment_method' => 'LiqPay'
    ]);
}
// Спробуємо створити замовлення
$pendingOrder = $this->liqPayService->getPendingOrder($orderId);
if (!$pendingOrder) {
    Log::error('Pending order not found', ['order_id' => $orderId]);
    return redirect()->route('user.orders')->with('error', 'Помилка при
створенні замовлення. Будь ласка, зверніться до служби підтримки.');
}
$order = $this->orderService->createOrderFromPayment($paymentData,
$pendingOrder);
if ($order) {
    return view('frontend.pages.payment-success', [
        'order_id' => $orderId,
        'amount' => $paymentData['amount'] ?? 0,
        'payment_method' => 'LiqPay'
    ]);
} else {
    Log::error('Failed to process successful payment', ['order_id' =>
$orderId]);
    return redirect()->route('user.orders')->with('error', 'Помилка при
створенні замовлення. Будь ласка, зверніться до служби підтримки.');
```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}
/** Обробка callback */
public function handleCallback(string $data, string $signature): array
{
    // Перевіряємо підпис
    if (!$this->liqPayService->verifySignature($data, $signature)) {
        Log::warning('LiqPay callback signature verification failed');
        return ['status' => 'error', 'message' => 'Signature verification failed',
'code' => 400];
    }
    // Декодуємо дані
    $paymentData = $this->liqPayService->decodePaymentData($data);
    if (!$paymentData) {
        return ['status' => 'error', 'message' => 'Failed to decode payment data',
'code' => 400];
    }
    Log::info('LiqPay callback payment data', $paymentData);
    if ($paymentData['status'] === 'success') {
        $orderId = $paymentData['order_id'] ?? null;
        $pendingOrder = $this->liqPayService->getPendingOrder($orderId);
        if ($pendingOrder) {
            $order = $this->orderService-
>createOrderFromPayment($paymentData, $pendingOrder);
            Log::info('LiqPay callback payment processing result', [
                'order_id' => $orderId,
                'processed' => (bool) $order
            ]);
        } else {

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Log::warning('Pending order not found for callback', ['order_id' =>
$orderId]);
    }
    } else {
        Log::warning('LiqPay payment not successful', [
            'status' => $paymentData['status'] ?? 'unknown',
            'order_id' => $paymentData['order_id'] ?? 'unknown'
        ]);
    }
    return ['status' => 'ok', 'message' => 'OK', 'code' => 200];
}

```

Додаток Б

```

class OrderService
{
    private PaymentValidator $validator;

    public function __construct(PaymentValidator $validator)
    {
        $this->validator = $validator;
    }

    /** Створення замовлення з даних платежу */

    public function createOrderFromPayment(array $paymentData, array
$pendingOrder): ?Order
    {
        try {
            // Валідуємо дані платежу

            $paymentErrors = $this->validator-
>validatePaymentData($paymentData);

            if (!empty($paymentErrors)) {
                Log::error('Invalid payment data', ['errors' => $paymentErrors]);
            }

            return null;
        }
    }
}

```

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    $orderId = $paymentData['order_id'];
    // Перевіряємо, чи замовлення вже існує
    $existingOrder = Order::where('order_id', $orderId)->first();
    if ($existingOrder) {
        Log::info('Order already exists in database: ' . $orderId);
        return $existingOrder;
    }
    DB::beginTransaction();
    try {
        $address = $pendingOrder['address'];
        // Валідуюємо адресу
        $addressErrors = $this->validator->validateAddressData($address);
        if (!empty($addressErrors)) {
            Log::error('Invalid address data', ['errors' => $addressErrors]);
            throw new \Exception('Неповні або невірні дані адреси');
        }
        // Валідуюємо дані кошика
        $cartErrors = $this->validator-
>validateCartData($pendingOrder['cart_items']);
        if (!empty($cartErrors)) {
            Log::error('Invalid cart data', ['errors' => $cartErrors]);
            throw new \Exception('Невірні дані кошика');
        }
        $amounts = $this->calculateAmounts($pendingOrder);
        // Визначаємо статус оплати та дату оплати на основі методу
оплати
        $paymentMethod = $paymentData['payment_method'] ?? 'liqpay';
        $paymentStatus = $paymentMethod === 'Накладений платіж' ?
'pending' : 'paid';
        $paidAt = $paymentMethod === 'Накладений платіж' ? null :
now();
        // Створюємо замовлення

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$order = Order::create([
    'order_id' => $orderId,
    'user_id' => $pendingOrder['user_id'],
    'subtotal' => $amounts['cart_total'],
    'shipping_fee' => $amounts['shipping_fee'],
    'total' => $amounts['final_amount'],
    'payment_method' => $paymentMethod,
    'payment_status' => $paymentStatus,
    'order_status' => 'pending',
    'shipping_name' => $address['name'],
    'shipping_email' => $address['email'],
    'shipping_phone' => $address['phone'],
    'shipping_address' => $address['address'],
    'shipping_city' => $address['city'],
    'shipping_state' => $address['state'] ?? '',
    'shipping_zip' => $address['zip'],
    'notes' => $address['notes'] ?? '',
    'payment_data' => $paymentData,
    'paid_at' => $paidAt
]);

Log::info('Order created successfully', ['order_db_id' => $order->id]);
// Створюємо елементи замовлення
$this->createOrderItems($order, $pendingOrder['cart_items']);
DB::commit();
// Очищуємо дані
$this->cleanupAfterOrder();
Log::info('Order processed successfully: ' . $orderId, ['order_id' =>
$order->id]);

return $order;

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    } catch (\Exception $e) {
        DB::rollback();
        Log::error('Error creating order in database: ' . $e->getMessage(), [
            'order_id' => $orderId,
            'error' => $e->getMessage(),
            'file' => $e->getFile(),
            'line' => $e->getLine(),
        ]);
        throw $e;
    }
} catch (\Exception $e) {
    Log::error('Error in createOrderFromPayment: ' . $e->getMessage(), [
        'payment_data' => $paymentData,
        'error' => $e->getMessage(),
    ]);
    return null;
}
}

/** Розрахування суми */
private function calculateAmounts(array $pendingOrder): array
{
    $shippingFee = Session::has('shipping_method') ?
Session::get('shipping_method')['cost'] : 0;
    $cartTotal = $pendingOrder['amount'] - $shippingFee;
    $finalAmount = $pendingOrder['amount'];
    Log::info('Using amounts from pending order', [
        'cart_total' => $cartTotal,
        'shipping_fee' => $shippingFee,
        'final_amount' => $finalAmount,
        'pending_amount' => $pendingOrder['amount']
    ]);
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    });
    return [
        'cart_total' => $cartTotal,
        'shipping_fee' => $shippingFee,
        'final_amount' => $finalAmount
    ];
}

/** Створення елементів замовлення */

private function createOrderItems(Order $order, array $cartItems): void
{
    Log::info('Processing cart items', ['cart_items_count' =>
count($cartItems)]);

    foreach ($cartItems as $item) {
        $productId = is_array($item) ? $item['id'] : $item->id;
        $name = is_array($item) ? $item['name'] : $item->name;
        $qty = is_array($item) ? $item['qty'] : $item->qty;
        $price = is_array($item) ? $item['price'] : $item->price;
        $options = is_array($item) ? ($item['options'] ?? []) : ($item->options ?
$item->options->toArray() : []);

        Log::info('Processing cart item', [
            'item_id' => $productId,
            'item_name' => $name,
            'item_qty' => $qty,
            'item_price' => $price
        ]);

        $product = Product::find($productId);
        if (!$product) {
            Log::warning('Product not found for item', ['item_id' => $productId]);
            continue;
        }
    }
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$orderItem = OrderItem::create([
    'order_id' => $order->id,
    'product_id' => $productId,
    'product_name' => $name,
    'unit_price' => $price,
    'quantity' => $qty,
    'total_price' => $price * $qty,
    'product_variants' => $options
]);

Log::info('Order item created', [
    'order_item_id' => $orderItem->id,
    'product_id' => $productId,
    'quantity' => $qty
]);

// Оновлюємо запаси продукту
$this->updateProductStock($product, $qty);
}
}

/** Оновлення запасів продукту */
private function updateProductStock(Product $product, int $qty): void
{
    if ($product->qty > 0) {
        $soldQty = $product->qty;
        $newQty = max(0, $soldQty - $qty);
        $product->qty = $newQty;
        $product->save();
        Log::info('Product quantity updated', [
            'product_id' => $product->id,
            'old_qty' => $soldQty,
            'new_qty' => $newQty

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    });
}
}
/** Очистка даних сесії після створення замовлення */
private function cleanupAfterOrder(): void
{
    Cart::destroy();
    Session::forget(['address', 'pending_order']);
}
}

```

Додаток Б

```

class OrderService
{
    private PaymentValidator $validator;
    public function __construct(PaymentValidator $validator)
    {
        $this->validator = $validator;
    }
    /** Створення замовлення з даних платежу */
    public function createOrderFromPayment(array $paymentData, array
$pendingOrder): ?Order
    {
        try {
            // Валідуємо дані платежу
            $paymentErrors = $this->validator-
>validatePaymentData($paymentData);
            if (!empty($paymentErrors)) {
                Log::error('Invalid payment data', ['errors' => $paymentErrors]);
                return null;
            }
        }
    }
}

```

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$orderId = $paymentData['order_id'];
// Перевіряємо, чи замовлення вже існує
$existingOrder = Order::where('order_id', $orderId)->first();
if ($existingOrder) {
    Log::info('Order already exists in database: ' . $orderId);
    return $existingOrder;
}
DB::beginTransaction();
try {
    $address = $pendingOrder['address'];
    // Валідуємо адресу
    $addressErrors = $this->validator->validateAddressData($address);
    if (!empty($addressErrors)) {
        Log::error('Invalid address data', ['errors' => $addressErrors]);
        throw new \Exception('Неповні або невірні дані адреси');
    }
    // Валідуємо дані кошика
    $cartErrors = $this->validator-
>validateCartData($pendingOrder['cart_items']);
    if (!empty($cartErrors)) {
        Log::error('Invalid cart data', ['errors' => $cartErrors]);
        throw new \Exception('Невірні дані кошика');
    }
    $amounts = $this->calculateAmounts($pendingOrder);
    // Визначаємо статус оплати та дату оплати на основі методу
оплати

    $paymentMethod = $paymentData['payment_method'] ?? 'liqpay';
    $paymentStatus = $paymentMethod === 'Накладений платіж' ?
'pending' : 'paid';
    $paidAt = $paymentMethod === 'Накладений платіж' ? null :
now();
    // Створюємо замовлення

```

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$order = Order::create([
    'order_id' => $orderId,
    'user_id' => $pendingOrder['user_id'],
    'subtotal' => $amounts['cart_total'],
    'shipping_fee' => $amounts['shipping_fee'],
    'total' => $amounts['final_amount'],
    'payment_method' => $paymentMethod,
    'payment_status' => $paymentStatus,
    'order_status' => 'pending',
    'shipping_name' => $address['name'],
    'shipping_email' => $address['email'],
    'shipping_phone' => $address['phone'],
    'shipping_address' => $address['address'],
    'shipping_city' => $address['city'],
    'shipping_state' => $address['state'] ?? '',
    'shipping_zip' => $address['zip'],
    'notes' => $address['notes'] ?? '',
    'payment_data' => $paymentData,
    'paid_at' => $paidAt
]);

Log::info('Order created successfully', ['order_db_id' => $order->id]);
// Створюємо елементи замовлення
$this->createOrderItems($order, $pendingOrder['cart_items']);
DB::commit();
// Очищуємо дані
$this->cleanupAfterOrder();
Log::info('Order processed successfully: ' . $orderId, ['order_id' =>
$order->id]);

return $order;

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    } catch (\Exception $e) {
        DB::rollback();
        Log::error('Error creating order in database: ' . $e->getMessage(), [
            'order_id' => $orderId,
            'error' => $e->getMessage(),
            'file' => $e->getFile(),
            'line' => $e->getLine(),
        ]);
        throw $e;
    }
} catch (\Exception $e) {
    Log::error('Error in createOrderFromPayment: ' . $e->getMessage(), [
        'payment_data' => $paymentData,
        'error' => $e->getMessage(),
    ]);
    return null;
}
}

/** Розрахування суми */
private function calculateAmounts(array $pendingOrder): array
{
    $shippingFee = Session::has('shipping_method') ?
Session::get('shipping_method')['cost'] : 0;
    $cartTotal = $pendingOrder['amount'] - $shippingFee;
    $finalAmount = $pendingOrder['amount'];

    Log::info('Using amounts from pending order', [
        'cart_total' => $cartTotal,
        'shipping_fee' => $shippingFee,
        'final_amount' => $finalAmount,
    ]);
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'pending_amount' => $pendingOrder['amount']
    );
    return [
        'cart_total' => $cartTotal,
        'shipping_fee' => $shippingFee,
        'final_amount' => $finalAmount
    ];
}

/** Створення елементів замовлення */

private function createOrderItems(Order $order, array $cartItems): void
{
    Log::info('Processing cart items', ['cart_items_count' =>
count($cartItems)]);

    foreach ($cartItems as $item) {
        $productId = is_array($item) ? $item['id'] : $item->id;
        $name = is_array($item) ? $item['name'] : $item->name;
        $qty = is_array($item) ? $item['qty'] : $item->qty;
        $price = is_array($item) ? $item['price'] : $item->price;
        $options = is_array($item) ? ($item['options'] ?? []) : ($item->options ?
$item->options->toArray() : []);

        Log::info('Processing cart item', [
            'item_id' => $productId,
            'item_name' => $name,
            'item_qty' => $qty,
            'item_price' => $price
        ]);

        $product = Product::find($productId);
        if (!$product) {
            Log::warning('Product not found for item', ['item_id' => $productId]);
            continue;
        }
    }
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    $orderItem = OrderItem::create([
        'order_id' => $order->id,
        'product_id' => $productId,
        'product_name' => $name,
        'unit_price' => $price,
        'quantity' => $qty,
        'total_price' => $price * $qty,
        'product_variants' => $options
    ]);
    Log::info('Order item created', [
        'order_item_id' => $orderItem->id,
        'product_id' => $productId,
        'quantity' => $qty
    ]);
    // Оновлюємо запаси продукту
    $this->updateProductStock($product, $qty);
}
}
/** Оновлення запасів продукту */
private function updateProductStock(Product $product, int $qty): void
{
    if ($product->qty > 0) {
        $soldQty = $product->qty;
        $newQty = max(0, $soldQty - $qty);
        $product->qty = $newQty;
        $product->save();
        Log::info('Product quantity updated', [
            'product_id' => $product->id,
            'old_qty' => $soldQty,

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'new_qty' => $newQty
    });
}
}

/** Очищення даних сесії після створення замовлення */
private function cleanupAfterOrder(): void
{
    Cart::destroy();
    Session::forget(['address', 'pending_order']);
}
}

```

Додаток В

```

class CartController extends Controller
{
    public function cartDetails()
    {
        $cartItems = Cart::content();
        return view('frontend.pages.cart-detail', compact('cartItems'));
    }

    public function addToCart(Request $request)
    {
        $product = Product::findOrFail($request->product_id);
        if($product->qty === 0){
            return response(['status' => 'error', 'message' => 'Товару немає в наявності']);
        }
        $variants = [];
        $variantTotalAmount = 0;
        if($request->has('variants_items')){
            foreach($request->variants_items as $item_id){

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $variantItem = ProductVariantItem::find($item_id);
        $variants[$variantItem->productVariant->name]['name'] =
$variantItem->name;
        $variants[$variantItem->productVariant->name]['price'] =
$variantItem->price;
        $variantTotalAmount += $variantItem->price;
    }
}
$productPrice = 0;
if ($product->discount_price > 0){
    $productPrice = $product->discount_price;
} else {
    $productPrice = $product->price;
}
$cartData = [];
$cartData['id'] = $product->id;
$cartData['name'] = $product->name;
$cartData['qty'] = $request->qty;
$cartData['price'] = $productPrice;
$cartData['weight'] = 10;
$cartData['options']['variants'] = $variants;
$cartData['options']['variants_total'] = $variantTotalAmount;
$cartData['options']['image'] = $product->image;
$cartData['options']['slug'] = $product->slug;
Cart::add($cartData);
return response(['status' => 'success', 'message' => 'Додано до кошику!']);
}

public function updateProductQty(Request $request)
{
    $productId = Cart::get($request->rowId)->id;

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$product = Product::findOrFail($productId);
// check product quantity
if($product->qty === 0){
    return response(['status' => 'error', 'message' => 'Товар закінчився на
складі']);
}elseif($product->qty < $request->qty){
    return response(['status' => 'error', 'message' => 'Недостатня кількість
товару на складі']);
}
Cart::update($request->rowId, $request->quantity);
$productTotal = $this->getProductTotal($request->rowId);
return response(['status' => 'success', 'message' => 'Кількість товару
оновлено!', 'product_total' => $productTotal]);
}
/** get product total */
public function getProductTotal($rowId)
{
    $product = Cart::get($rowId);
    $total = ($product->price + $product->options->variants_total) * $product-
>qty;
    return $total;
}
/** get cart total amount */
public function cartTotal()
{
    $total = 0;
    foreach(Cart::content() as $product){
        $total += $this->getProductTotal($product->rowId);
    }
}

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $total;
    }

    /** clear all cart products */
    public function clearCart()
    {
        Cart::destroy();
        return response(['status' => 'success', 'message' => 'Кошик успішно
очищено']);
    }

    /** Remove product form cart */
    public function removeProduct($rowId)
    {
        Cart::remove($rowId);
        toastr('Товар успішно видалено!', 'success');
        return redirect()->back();
    }

    /** Get cart count */
    public function getCartCount()
    {
        return Cart::content()->count();
    }

    /** Get all cart products */
    public function getCartProducts()
    {
        return Cart::content();
    }

    /** Romve product form sidebar cart */
    public function removeSidebarProduct(Request $request)
    {
        Cart::remove($request->rowId);
    }

```

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						86
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return response(['status' => 'success', 'message' => 'Товар успішно
видалено!']);

```

```

    }
}

```

Додаток Г

```

class ProductDataTable extends DataTable
{
    /** Build the DataTable class. ** @param QueryBuilder $query Results
from query() method. */
    public function dataTable(QueryBuilder $query): EloquentDataTable
    {
        return (new EloquentDataTable($query))
            ->addColumn('action', function($query){
                $editBtn = "<a href='".route('admin.products.edit', $query->id).'"
class='btn btn-primary'><i class='far fa-edit'></i></a>";
                $deleteBtn = "<a href='".route('admin.products.destroy', $query-
>id).'" class='btn btn-danger ml-2 delete-item'><i class='far fa-trash-
alt'></i></a>";
                $moreBtn = '<div class="dropdown dropleft d-inline">
<button class="btn btn-primary dropdown-toggle ml-1" type="button"
id="dropdownMenuButton2" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
<i class="fas fa-cog"></i>
</button>
<div class="dropdown-menu" x-placement="bottom-start"
style="position: absolute; transform: translate3d(0px, 28px, 0px); top: 0px; left:
0px; will-change: transform;">

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

```
<a class="dropdown-item has-icon" href="'.route('admin.products-
image-gallery.index', ['product' => $query->id]).'"><i class="far fa-heart"></i>
Галерея зображень</a>
```

```
<a class="dropdown-item has-icon" href="'.route('admin.products-
variant.index', ['product' => $query->id]).'"><i class="far fa-file"></i>
Варіанти</a>
```

```
</div>
</div>;

return $editBtn.$deleteBtn.$moreBtn;
})

->addColumn('image', function($query){
    return "<img width='70px' src='".asset($query->image).'" ></img>";
})

->addColumn('type', function($query){
    switch ($query->product_type) {
        case 'new_arrival':
            return '<i class="badge badge-success">Новинка</i>';
            break;
        case 'featured_product':
            return '<i class="badge badge-warning">Рекомендований</i>';
            break;
        case 'top_product':
            return '<i class="badge badge-info">Топ</i>';
            break;
        default:
            return '<i class="badge badge-dark">Немає</i>';
            break;
    }
})

->addColumn('status', function($query){
```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						88
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if($query->status == 1){
            $button = '<label class="custom-switch mt-2">
                <input type="checkbox" checked name="custom-switch-
checkbox" data-id="'. $query->id.'" class="custom-switch-input change-status" >
                <span class="custom-switch-indicator"></span>
            </label>';
        }else {
            $button = '<label class="custom-switch mt-2">
                <input type="checkbox" name="custom-switch-checkbox" data-
id="'. $query->id.'" class="custom-switch-input change-status">
                <span class="custom-switch-indicator"></span>
            </label>';
        }
        return $button;
    })
    ->rawColumns(['image', 'type', 'status', 'action'])
    ->setRowId('id');
}

/** Get the query source of dataTable.*/
public function query(Product $model): QueryBuilder
{
    return $model->newQuery();
}

/**
* Optional method if you want to use the html builder.
*/
public function html(): HtmlBuilder
{
    return $this->builder()
        ->setTableId('product-table')

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						89
Змн.	Арк.	№ докум.	Підпис	Дата		

```

->columns($this->getColumns())
->minifiedAjax()
//->dom('Bfrtip')
->orderBy(0)
->selectStyleSingle()
->language([
    'url' => '//cdn.datatables.net/plug-
ins/1.10.25/i18n/Ukrainian.json'
])
->buttons([
    Button::make('excel'),
    Button::make('csv'),
    Button::make('pdf'),
    Button::make('print'),
    Button::make('reset'),
    Button::make('reload')
]);
}

/** Get the dataTable columns definition.*/
public function getColumns(): array
{
    return [
        Column::make('id'),
        Column::make('image')->title('Зображення'),
        Column::make('slug')->title('URL'),
        Column::make('name')->title('Назва'),
        Column::make('price')->title('Ціна'),
        Column::make('type')->title('Тип'),
        Column::make('qty')->title('Кількість'),
        Column::make('status')->title('Статус'),

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Column::computed('action')
        ->exportable(false)
        ->printable(false)
        ->width(200)
        ->title('Дії')
        ->addClass('text-center'),
    ];
}

/** Get the filename for export. */
protected function filename(): string
{
    return 'Product_' . date('YmdHis');
}
}

```

Додаток Г

```

trait ImageUploadTrait {
    public function uploadImage(Request $request, $inputName, $path)
    {
        if($request->hasFile($inputName)){
            $image = $request->{$inputName};
            $ext = $image->getClientOriginalExtension();
            $imageName = 'media_'.uniqid().'.'.$ext;
            $image->move(public_path($path), $imageName);
            return $path.'/'.$imageName;
        }
    }

    public function uploadMultiImage(Request $request, $inputName, $path)
    {
        $imagePaths = [];

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if($request->hasFile($inputName)){
    $images = $request->{$inputName};
    foreach($images as $image){
        $ext = $image->getClientOriginalExtension();
        $imageName = 'media_'.uniqid().'.'.$ext;
        $image->move(public_path($path), $imageName);
        $imagePaths[] = $path.'/'.$imageName;
    }
    return $imagePaths;
}
}

public function updateImage(Request $request, $inputName, $path,
SoldPath=null)
{
    if($request->hasFile($inputName)){
        if(File::exists(public_path($soldPath))){
            File::delete(public_path($soldPath));
        }
        $image = $request->{$inputName};
        $ext = $image->getClientOriginalExtension();
        $imageName = 'media_'.uniqid().'.'.$ext;
        $image->move(public_path($path), $imageName);
        return $path.'/'.$imageName;
    }
}

/** Delete File */
public function deleteImage(string $path)
{
    if(File::exists(public_path($path))){
        File::delete(public_path($path));
    }
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
  }
}

```

Додаток Д

```

class FrontendProductController extends Controller
{
    public function productsIndex(Request $request)
    {
        $query = $this->getBaseProductQuery();
        if ($request->has('category')) {
            $category = Category::where('slug', $request->category)->firstOrFail();
            $query->where('category_id', $category->id);
        } elseif ($request->has('subcategory')) {
            $category = SubCategory::where('slug', $request->subcategory)-
            >firstOrFail();
            $query->where('sub_category_id', $category->id);
        } elseif ($request->has('childcategory')) {
            $category = ChildCategory::where('slug', $request->childcategory)-
            >firstOrFail();
            $query->where('child_category_id', $category->id);
        } elseif ($request->has('brand')) {
            $brand = Brand::where('slug', $request->brand)->firstOrFail();
            $query->where('brand_id', $brand->id);
        } elseif ($request->has('search')) {
            $query->where(function ($q) use ($request) {
                $q->where('name', 'like', '%'.$request->search.'%')
                ->orWhere('long_description', 'like', '%'.$request->search.'%')
                ->orWhereHas('category', function($subQ) use ($request) {
                    $subQ->where('name', 'like', '%'.$request->search.'%')
                    ->orWhere('long_description', 'like', '%'.$request->search.'%');
                });
            });
        }
    }
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						93
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        });
    });
}
$query = $this->applyPriceFilter($query, $request);
$query = $this->applyRatingFilter($query, $request);
$query = $this->applySorting($query, $request);
$products = $query->paginate(12);
$categories = Category::where('status', 1)->get();
$brands = Brand::where('status', 1)->get();
$maxPrice = Product::where('status', 1)
    ->selectRaw('MAX(COALESCE(NULLIF(discount_price, 0), price)) as
max_price')
    ->value('max_price');
$maxPrice = $maxPrice ?? 1000;
return view('frontend.pages.product', compact('products', 'categories',
'brands', 'maxPrice'));
}
private function applyPriceFilter($query, Request $request)
{
    if (!$request->has('range')) {
        return $query;
    }
    $price = explode(':', $request->range);
    $from = $price[0];
    $to = $price[1];
    return $query->where(function($q) use ($from, $to) {
        $q->where(function($subQ) use ($from, $to) {
            $subQ->where('discount_price', '>', 0)
                ->whereNotNull('discount_price')
                ->where('discount_price', '>=', $from)

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        ->where('discount_price', '<=', $to);
    })->orWhere(function($subQ) use ($from, $to) {
        $subQ->where(function($innerQ) use ($from, $to) {
            $innerQ->where('discount_price', '=', 0)
                ->orWhereNull('discount_price');
        })
        ->where('price', '>=', $from)
        ->where('price', '<=', $to);
    });
});
}

private function applyRatingFilter($query, Request $request)
{
    if (!$request->has('rating')) {
        return $query;
    }

    return $query->havingRaw('reviews_avg_rating >= ?', [$request->rating]);
}

private function applySorting($query, Request $request)
{
    if (!$request->has('sort')) {
        return $query->orderBy('id', 'DESC');
    }

    switch ($request->sort) {
        case 'price_low_high':
            return $query->orderByRaw('COALESCE(NULLIF(discount_price,
0), price) ASC');
        case 'price_high_low':
            return $query->orderByRaw('COALESCE(NULLIF(discount_price,
0), price) DESC');
    }
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						95
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        case 'rating':
            return $query->orderBy('reviews_avg_rating', 'DESC');
        case 'newest':
            return $query->orderBy('created_at', 'DESC');
        default:
            return $query->orderBy('id', 'DESC');
    }
}

private function getBaseProductQuery()
{
    return Product::withAvg(['reviews' => function ($query) {
        $query->where('status', 1);
    }], 'rating')->withCount(['reviews' => function ($query) {
        $query->where('status', 1);
    }])
    ->with(['variants', 'category', 'productImageGalleries'])
    ->where('status', 1);
}

public function showProduct(string $slug)
{
    $product = Product::with(['category', 'subCategory', 'childCategory',
'productImageGalleries', 'variants', 'brand'])->where('slug', $slug)->where('status',
1)->first();

    $reviews = ProductReview::where('product_id', $product->id)-
>where('status', 1)->paginate(10);

    return view('frontend.pages.product-detail', compact('product', 'reviews'));
}

public function categoriesIndex()
{
    $categories = Category::where('status', 1)

```

					ПІЗ.КР.Б-121-25-ПІЗ	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дата		

```

->with(['subCategories' => function($query){
    $query->where('status', 1)
    ->with(['childCategories' => function($query){
        $query->where('status', 1);
    }]);
}])
->get();
return view('frontend.pages.categories', compact('categories'));
}
}

```

Додаток Ж

```

class ReviewController extends Controller
{
    use ImageUploadTrait;
    public function index(UserProductReviewsDataTable $dataTable)
    {
        return $dataTable->render('frontend.dashboard.review.index');
    }
    public function create(Request $request)
    {
        $request->validate([
            'rating' => ['required'],
            'review' => ['required', 'max:200'],
            'images.*' => ['image']
        ]);
        $checkReviewExist = ProductReview::where(['product_id' => $request-
        >product_id, 'user_id' => Auth::user()->id])->first();
        if($checkReviewExist){
            toastr('Ви вже додали відгук на цей товар!', 'error');
            return redirect()->back();
        }
    }
}

```

					ІПЗ.КР.Б-121-25-ІПЗ	Арк.
						97
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    $imagePaths = $this->uploadMultiImage($request, 'images', 'uploads');
    $productReview = new ProductReview();
    $productReview->product_id = $request->product_id;
    $productReview->user_id = Auth::user()->id;
    $productReview->rating = $request->rating;
    $productReview->review = $request->review;
    $productReview->status = 0;
    $productReview->save();
    if(!empty($imagePaths)){
        foreach($imagePaths as $path){
            $reviewGallery = new ProductReviewGallery();
            $reviewGallery->product_review_id = $productReview->id;
            $reviewGallery->image = $path;
            $reviewGallery->save();
        }
    }
    toastr('Відгук успішно додано!', 'success');
    return redirect()->back();
}
}

```

Додаток И

```

class AppServiceProvider extends ServiceProvider
{
    /**
     * Register any application services.
     */
    public function register(): void
    {
        $this->app->bind(LiqPayService::class, function ($app) {
            return new LiqPayService();
        });
    }
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$this->app->bind(PaymentValidator::class, function ($app) {
    return new PaymentValidator();
});
$this->app->bind(CodService::class, function ($app) {
    return new CodService();
});
$this->app->bind(WayForPayService::class, function ($app) {
    return new WayForPayService();
});
$this->app->bind(OrderService::class, function ($app) {
    return new OrderService($app->make(PaymentValidator::class));
});
$this->app->bind(PaymentResponseHandler::class, function ($app) {
    return new PaymentResponseHandler(
        $app->make(LiqPayService::class),
        $app->make(WayForPayService::class),
        $app->make(OrderService::class)
    );
});
}

/** Bootstrap any application services.*/
public function boot(): void
{
    Route::middleware(['web', 'auth', 'role:admin'])
        ->prefix('admin')
        ->as('admin.')
        ->group(base_path('routes/admin.php'));
}
}

```

					ІІЗ.КР.Б-121-25-ІІЗ	Арк.
						99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

class Product extends Model
{
    use HasFactory;
    public function category()
    {
        return $this->belongsTo(Category::class);
    }
    public function subCategory()
    {
        return $this->belongsTo(SubCategory::class);
    }
    public function childCategory()
    {
        return $this->belongsTo(ChildCategory::class);
    }
    public function productImageGalleries()
    {
        return $this->hasMany(ProductImageGallery::class);
    }
    public function variants()
    {
        return $this->hasMany(ProductVariant::class);
    }
    public function brand()
    {
        return $this->belongsTo(Brand::class);
    }
    public function reviews()
    {

```

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						100
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $this->hasMany(ProductReview::class);
    }

    public function wishlist()
    {
        return $this->hasMany(Wishlist::class);
    }

    public function orderItems()
    {
        return $this->hasMany(OrderItem::class);
    }
}

```

					ІПЗ.КР.Б-121-25-ІЗ	Арк.
						101
Змн.	Арк.	№ докум.	Підпис	Дата		