

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:

«Вебплатформа продажу мото-товарів»

Виконав студент групи ПЗ-22-2
МОРОЗЮК Юрій Анатолійович

Керівник роботи:
ЄФРЕМОВ Юрій Миколайович

Рецензент:
ТОЛСТОЙ Ігор Анатолійович

Житомир – 2026

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
«28» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **МОРОЗІЮК Юрій Анатолійович**

Керівник роботи: **ЄФРЕМОВ Юрій Миколайович**

Тема роботи: **«Вебплатформа продажу мото-товарів»,**

затверджена Наказом закладу вищої освіти від **«27» лютого 2026 р., № 92**

Вихідні дані для роботи: **вимоги до вебзастосунку електронної торгівлі,
каталог демонстраційних товарів, ролі покупця й адміністратора.**

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Єфремов Ю.М.	01.03.2026р.	01.03.2026р
2	Єфремов Ю.М.	01.03.2026р	01.03.2026р
3	Єфремов Ю.М.	01.03.2026р	01.03.2026р

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра містить 78 сторінок, 20 рисунків, 18 таблиць, 30 використаних джерел та 3 додатки.

Метою роботи є створення вебзастосунку інтернет-магазину MotoMarket PRO, призначеного для продажу мотоекіпірування, запасних частин, витратних матеріалів, електроніки та аксесуарів. Система забезпечує повний базовий цикл електронної торгівлі: пошук товару, перегляд характеристик, роботу з обраним і кошиком, оформлення та перегляд замовлень.

Об'єктом дослідження є процес розроблення інформаційної системи для автоматизації продажу товарів спеціалізованого мотوماгазину. Предметом дослідження є методи серверної розробки на PHP, організації реляційної бази даних MySQL, керування сесіями, побудови адаптивного інтерфейсу та розмежування прав покупця й адміністратора.

У роботі виконано аналіз предметної області, сформовано функціональні та нефункціональні вимоги, спроектовано архітектуру і структуру бази даних, описано алгоритми автентифікації та оформлення замовлення. Реалізовано каталог із пошуком, фільтрами й сортуванням, картки товарів, відгуки, профіль, список бажань, кошик, історію замовлень, контактну форму та адміністративну панель.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ-МАГАЗИН, МОТОТОВАРИ, PHP, MYSQL, PDO, ЕЛЕКТРОННА КОМЕРЦІЯ, КАТАЛОГ, КОШИК, ЗАМОВЛЕННЯ, АДМІНІСТРАТИВНА ПАНЕЛЬ.

					ІПЗ.КР.Б – 121 – 26 - ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Морозюк Ю.А.			Вебплатформа продажу мото-товарів	Літ.	Арк.	Аркушів
Керівник		Єфремов Ю.М					3	78
						Житомирська політехніка, група ІПЗ-22-2		
Рецензент		Толстой І.А.						
Зав.каф		Вакалюк Т.А.						

ABSTRACT

The bachelor's qualification thesis contains 78 pages, 20 figures, 18 tables, 30 references, and 3 appendices.

The purpose of the work is to develop the MotoMarket PRO online store for motorcycle equipment, spare parts, consumables, electronics, and accessories. The system covers the essential e-commerce workflow: product discovery, product details, wishlist, shopping cart, checkout, and order history.

The object of research is the development process of an information system that automates sales operations for a specialized motorcycle store. The subject includes PHP server-side development, MySQL relational data modeling, session-based authentication, responsive interface design, and role-based access.

The thesis analyzes the domain and comparable solutions, defines functional and quality requirements, presents the architecture and database schema, and describes authentication and checkout algorithms. The implemented application contains a searchable catalog, product filters, product reviews, customer profiles, wishlist, persistent cart, checkout, order tracking, contact requests, and an administrator dashboard.

KEYWORDS: ONLINE STORE, MOTORCYCLE GOODS, PHP, MYSQL, PDO, E-COMMERCE, PRODUCT CATALOG, SHOPPING CART, ORDER, ADMINISTRATION.

					ІІЗ.КР.Б – 121 – 26 - ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CRUD – Create, Read, Update, Delete; базові операції керування даними
CSS – Cascading Style Sheets; каскадні таблиці стилів
DBMS / СУБД – система управління базами даних
HTML – HyperText Markup Language; мова розмітки гіпертексту
HTTP – HyperText Transfer Protocol; протокол обміну вебданими
PDO – PHP Data Objects; універсальний інтерфейс доступу до баз даних
PHP – серверна мова програмування загального призначення
SQL – Structured Query Language; мова структурованих запитів
UI – User Interface; інтерфейс користувача
UX – User Experience; досвід взаємодії користувача
XSS – Cross-Site Scripting; міжсайтове виконання сценаріїв

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	9
1.1. Постановка задачі	9
1.2. Аналіз аналогів програмного продукту	12
1.3. Вибір архітектури платформи	15
1.4. Обґрунтування вибору інструментальних засобів	19
Висновки до першого розділу	23
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА МОТОМАРКЕТ PRO....	24
2.1. Визначення варіантів використання та структури системи	24
2.2. Проектування бази даних.....	29
2.3. Проектування та реалізація алгоритмів роботи.....	34
2.4. Особливості програмної реалізації вебзастосунку.....	40
Висновки до другого розділу.....	44
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ	45
3.1. Порядок встановлення та налаштування системи.....	45
3.2. Структура інтерфейсу та порядок взаємодії з платформою.....	49
3.3. Тестування роботи програмного продукту	58
Висновки до третього розділу	62
ВИСНОВКИ	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
ДОДАТКИ.....	67

ВСТУП

Актуальність теми – Сучасний розвиток інформаційних технологій та стрімке поширення електронної комерції суттєво змінюють підходи до організації торговельної діяльності. Все більше підприємств переходять від традиційних способів продажу до використання веборієнтованих інформаційних систем, які забезпечують цілодобовий доступ до товарів, спрощують процес взаємодії з клієнтами та дозволяють автоматизувати значну частину бізнес-процесів. Особливо актуальним це є для невеликих спеціалізованих магазинів, які прагнуть розширити аудиторію покупців та підвищити ефективність своєї роботи без значних фінансових витрат.

Ринок мототехніки та супутніх товарів в Україні продовжує активно розвиватися. Разом зі зростанням кількості власників мотоциклів збільшується попит на запчастини, екіпірування, аксесуари та сервісне обслуговування. Водночас значна частина спеціалізованих мотомагазинів використовує застарілі способи організації продажів або обмежується розміщенням товарів на універсальних маркетплейсах і дошках оголошень. Такі рішення не завжди забезпечують достатній рівень контролю над асортиментом, замовленнями та взаємодією з клієнтами.

Існуючі маркетплейси орієнтовані на широкий спектр товарів і не враховують специфіку діяльності мотомагазинів. Крім того, продавці змушені дотримуватися правил сторонніх платформ, сплачувати комісії та мають обмежені можливості щодо персоналізації інтерфейсу й організації власних бізнес-процесів. У зв'язку з цим актуальним завданням є створення власного вебзастосунку, який забезпечить повний цикл електронної торгівлі та надасть можливість самостійного керування всіма аспектами роботи магазину.

Важливим напрямом сучасної веброзробки є використання перевірених технологій, які забезпечують надійність, безпеку та простоту впровадження програмного забезпечення. Одним із таких рішень є поєднання PHP, MySQL, HTML5, CSS3 та JavaScript, що дозволяє створювати повноцінні вебзастосунки без використання складної серверної інфраструктури. Тому

					ПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

розробка вебзастосунку для автоматизації роботи мотомагазину є актуальним завданням як з практичної, так і з навчально-дослідницької точки зору.

Мета роботи – випускної кваліфікаційної роботи є проектування та розробка вебзастосунку «MotoMarket PRO» для автоматизації процесів продажу мототоварів, керування каталогом продукції, оформлення замовлень та адміністрування діяльності спеціалізованого мотомагазину. Досягнення поставленої мети передбачає створення інформаційної системи, яка забезпечує централізоване зберігання даних, підтримку основних бізнес-процесів електронної торгівлі, розмежування прав доступу користувачів та зручний інтерфейс для покупців і адміністратора.

Завдання роботи – для досягнення поставленої мети необхідно дослідити особливості організації продажу мототоварів у мережі Інтернет, проаналізувати існуючі програмні рішення, визначити вимоги до розроблюваної системи, спроектувати її архітектуру та структуру бази даних, реалізувати основні функціональні модулі вебзастосунку MotoMarket PRO, забезпечити механізми автентифікації користувачів і розмежування прав доступу, створити інструменти керування товарами та замовленнями, а також виконати тестування системи для перевірки її працездатності та відповідності поставленим вимогам.

Об'єктом дослідження – є процес автоматизації продажу товарів спеціалізованого мотомагазину із використанням сучасних вебтехнологій, що охоплює представлення товарного асортименту, взаємодію користувачів із каталогом продукції, оформлення та обробку замовлень, а також управління даними магазину в умовах електронної комерції.

Предмет дослідження – є методи та засоби розробки веборієнтованих інформаційних систем електронної комерції, зокрема використання мови програмування PHP, системи керування базами даних MySQL, інтерфейсу PDO, механізмів автентифікації користувачів, транзакційної обробки даних та технологій створення адаптивних вебінтерфейсів.

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1. АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1. Постановка задачі

Спеціалізований мотомагазин працює з широким та різномірним асортиментом товарів, до якого входять захисні шоломи, мотоекіпірування, запасні частини, витратні матеріали, електроніка та аксесуари для мотоциклів. У традиційній моделі організації продажів покупець отримує інформацію про наявність продукції шляхом телефонного звернення до магазину або безпосередньо під час відвідування торгової точки. Такий підхід має низку суттєвих недоліків, зокрема ускладнює оперативне порівняння товарів за характеристиками, не дозволяє самостійно перевіряти актуальні складські залишки та створює додаткові труднощі при оформленні повторних замовлень без участі працівників магазину.

Використання інтернет-магазину як сучасного формату електронної комерції дозволяє ефективно вирішити зазначені проблеми. Проте значна частина невеликих спеціалізованих продавців мотоциклетної продукції або розміщує свої товари на великих маркетплейсах, втрачаючи власну тематичну ідентичність та можливість індивідуального представлення бренду, або взагалі не має власного цифрового представництва в мережі Інтернет. На універсальних торговельних платформах мотоциклетна продукція часто розміщується серед тисяч інших категорій товарів, що значно ускладнює навігацію для цільової аудиторії. Крім того, можливості фільтрації за технічними характеристиками нерідко є обмеженими, а продавці не мають повного контролю над структурою контенту, оформленням каталогу та процесами обробки замовлень.

Саме ці фактори стали основною передумовою для проєктування та розробки власного вебзастосунку MotoMarket PRO. Розроблена система орієнтована на створення зручного цифрового середовища як для покупців, так і для адміністрації магазину. Платформа надає можливість здійснювати

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

пошук і фільтрацію товарів за визначеними параметрами, включаючи категорію, бренд, ціну та інші характеристики. Також реалізовано механізм відображення актуальних залишків товарів у режимі реального часу з автоматичним оновленням інформації після кожного оформленого замовлення, що забезпечує високу точність даних та зменшує ризик продажу відсутніх позицій.

Окрім функціоналу для покупців, система передбачає повноцінний адміністративний модуль, призначений для ефективного управління діяльністю магазину. Для адміністраторів створено окремий захищений інтерфейс, доступ до якого надається лише після проходження процедури автентифікації та перевірки ролей користувача. За допомогою цього інтерфейсу адміністратор може здійснювати повний облік товарів, додавати нові позиції до каталогу, редагувати інформацію про наявні товари та контролювати складські залишки. Також передбачено можливість перегляду замовлень, зміни їх статусів на різних етапах виконання, опрацювання контактних звернень від клієнтів та управління обліковими записами зареєстрованих покупців. Крім того, адміністративна панель забезпечує централізований доступ до всієї інформації, необхідної для оперативного прийняття управлінських рішень. Система дозволяє відстежувати історію замовлень, аналізувати активність користувачів та контролювати актуальність даних у каталозі товарів. Це сприяє підвищенню ефективності роботи персоналу та зменшенню кількості помилок під час виконання повсякденних операцій.

Таким чином, впровадження вебзастосунку MotoMarket PRO спрямоване на підвищення ефективності роботи спеціалізованого мотомагазину, автоматизацію основних бізнес-процесів та покращення якості обслуговування клієнтів. Для наочного представлення функціональних вимог до системи основні задачі платформи MotoMarket PRO, їх зміст та відповідні інструменти реалізації наведено у таблиці (табл. 1.1).

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Таблиця 1.1

Основні задачі системи та їх реалізація

Задача	Зміст	Реалізація у MotoMarket PRO
Структурований каталог	Стандартизовані поля замість довільного опису	Таблиця products, сторінки index.php і shop.php
Пошук і фільтрація	Вибір товарів за текстом та атрибутами	LIKE-запити, фільтри категорії, бренду й ціни
Облік залишків	Актуальний stock із оновленням після замовлення	UPDATE products у транзакції checkout.php
Кошик і замовлення	Збереження вибраних позицій та фіксація покупки	Таблиці cart, orders, order_items через PDO
Персоналізація	Профіль, список бажань, відгуки, історія	Таблиці users, wishlist, reviews
Адміністрування	Керування магазином з окремим контуром доступу	Сторінки каталогу admin/, перевірка is_admin()

Дані, наведені в таблиці 1.1, демонструють основні функціональні задачі вебзастосунку MotoMarket PRO та способи їх реалізації в системі. Реалізація структурованого каталогу, механізмів пошуку й фільтрації, обліку складських залишків, оформлення замовлень, персоналізації користувацького досвіду та адміністративного управління забезпечує комплексну автоматизацію діяльності мотوماгазину. Запропоновані рішення спрямовані на підвищення зручності використання системи для покупців і спрощення виконання операцій для адміністрації магазину. Для обґрунтування доцільності розробки власного програмного продукту доцільно провести аналіз існуючих аналогів та визначити їхні переваги й недоліки.

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1.2. Аналіз аналогів програмного продукту

Перед розробкою було розглянуто три типи майданчиків, якими фактично користуються покупці мотоциклетних товарів в Україні: великий універсальний маркетплейс ROZETKA, спеціалізований мотомагазин MotoZona, а також дошки оголошень і групи у соціальних мережах.

ROZETKA — найбільший універсальний маркетплейс України з широким асортиментом, у тому числі мотокатегорії. Сильні сторони: розвинений пошук і фільтри, система рейтингів і відгуків, впізнаваний процес оформлення замовлення та відстеження доставки. Проблема для нішевого продавця полягає в тому, що асортимент мотоциклетних товарів розпорошений між сотнями продавців, бренди конкурують на одній сторінці, тематична подача відсутня, а продавець не контролює власний адміністративний простір незалежно від платформи. Інтерфейс аналога наведено на рисунку (рис. 1.1).

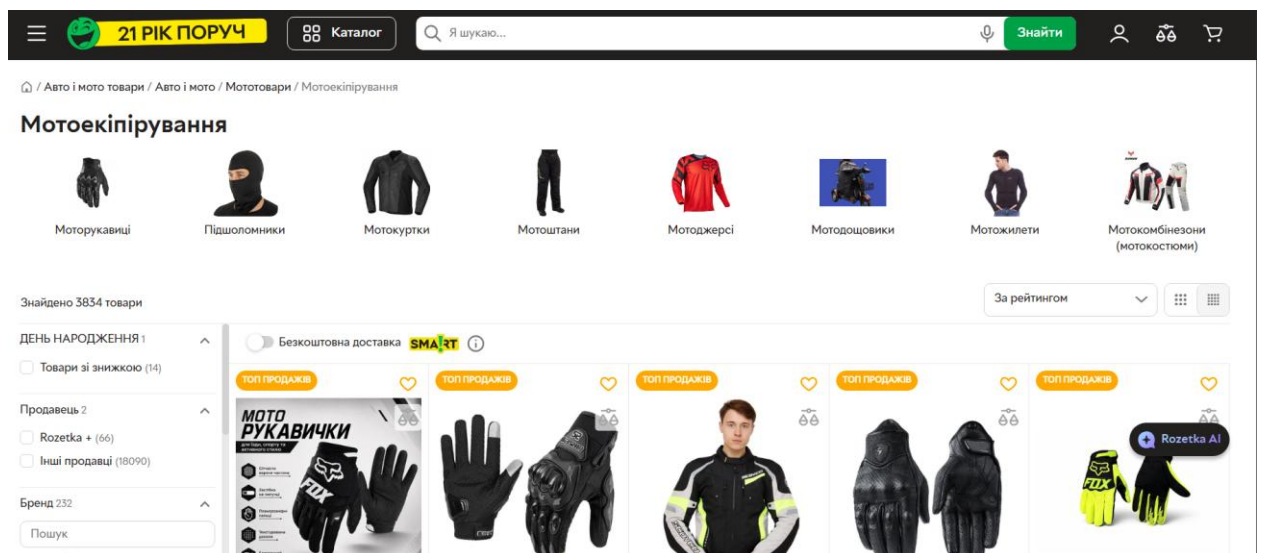


Рис. 1.1 – Платформа ROZETKA (розділ мотоекіпірування)

MotoZona — це спеціалізований інтернет-магазин мотоциклетних товарів, який орієнтований на вузьку, але дуже активну аудиторію. Його сильні сторони полягають у чіткій тематичній структурі категорій (шоломи, екіпірування, запчастини, аксесуари), дизайні, що відповідає стилю

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

байкерської культури, та релевантності цільовій групі — власникам і фанатам мотоциклів. Завдяки цьому користувачі швидко знаходять потрібні товари, а магазин краще відповідає потребам своєї аудиторії.

Таким чином, MotoZona демонструє приклад нішевого магазину з сильним позиціонуванням, але водночас ілюструє типові виклики галузі, які можна вирішити за допомогою вдосконалення UX та автоматизації бізнес-процесів. Аналіз функціональних можливостей цього ресурсу показує, що навіть спеціалізовані інтернет-магазини потребують постійного розвитку для задоволення зростаючих вимог користувачів. Зокрема, важливими напрямками вдосконалення є покращення навігації, оптимізація процесу оформлення замовлень, розширення можливостей персоналізації та впровадження сучасних засобів управління каталогом і замовленнями (рис. 1.2).

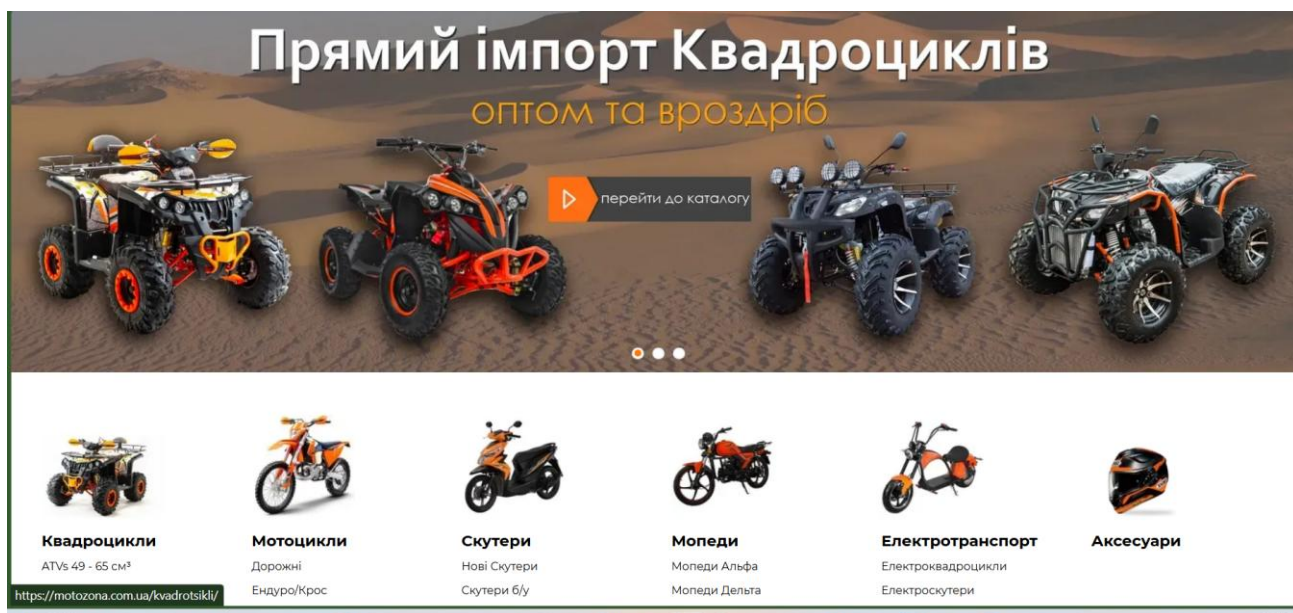


Рис. 1.2 – Спеціалізований мотوماгазин MotoZona

Групи у Facebook — найпоширеніший стихійний варіант продажу мотоциклетних товарів серед приватних продавців. Сильні сторони: низький поріг входження, активна аудиторія та миттєвий зворотний зв'язок у коментарях. Проблеми: повна відсутність єдиного формату картки товару, актуальний залишок невідомий, постійний кошик та історія замовлень не

підтримуються, а координація повторних покупок ускладнена. Інтерфейс груп у соціальних мережах наведено на рисинку (рис. 1.3).

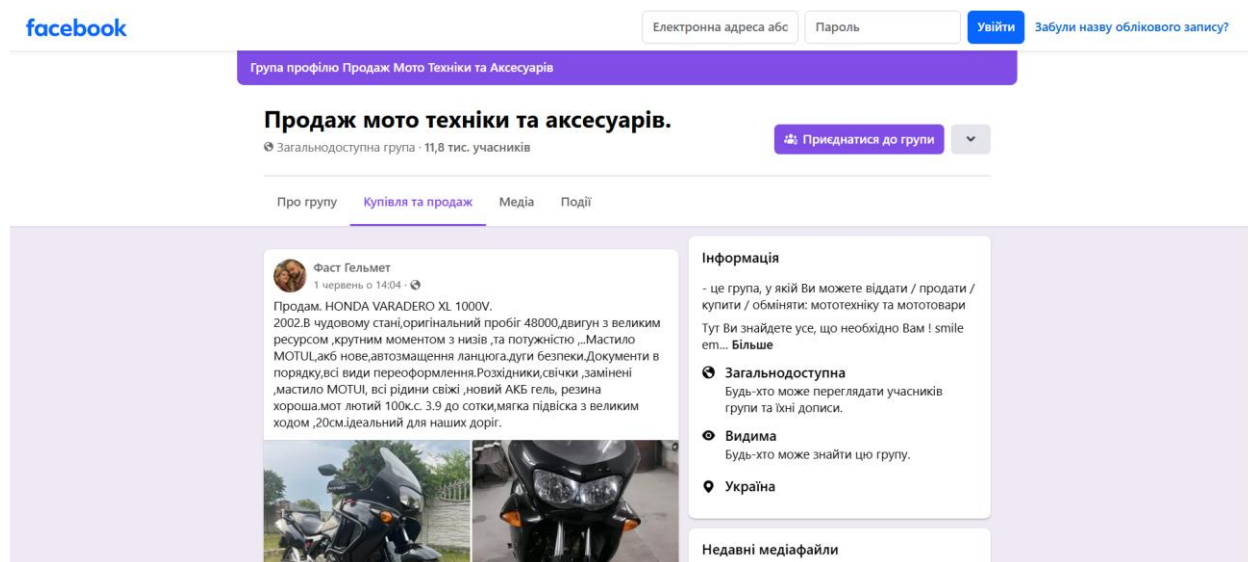


Рис. 1.3 – Групи продажу мототоварів у Facebook

У результаті аналізу від спеціалізованих магазинів запозичено підхід до тематичної структури каталогу та орієнтованого дизайну, від маркетплейсів логіку відгуків, рейтингів та прозорого процесу замовлення, а слабкі сторони всіх аналогів закриті власною структурованою базою даних, контролем залишків і повноцінною адміністративною панеллю це наведено у таблиці (табл. 1.2).

Таблиця 1.2

Аналіз конкурентних рішень у сфері продажу мотоциклетних товарів

Критерій	ROZETKA	MotoZona	Facebook	MotoMarket PRO
Тематичний каталог	Частково	Так	Ні	Так
Пошук і фільтри	Розширені	Залежить від сайту	Обмежені	Так
Постійний кошик	Так	Так	Ні	Так
Список бажань	Так	Часто	Ні	Так
Облік залишків	Так	Так	Ні	Так

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк. 14
Змн.	Арк.	№ докум.	Підпис	Дата		

Аналіз, наведений у (табл. 1.2), дозволяє порівняти функціональні можливості існуючих рішень у сфері продажу мотоциклетних товарів та визначити їхні переваги й недоліки. Кожному з порівняних критеріїв відповідає окремий програмний модуль, реалізований у розділі 2: тематичний каталог відповідає сценарію UC-Catalog, пошук і фільтри — UC-Search, постійний кошик — UC-Cart, список бажань — UC-Wishlist, а облік залишків — сутності бази даних ERD-Products.

Великі маркетплейси забезпечують широкий набір функцій для покупців, проте не орієнтовані виключно на мототематику. Спеціалізовані магазини, такі як MotoZona, мають більш релевантну структуру каталогу та тематичне оформлення, однак їхня функціональність часто обмежується базовими можливостями електронної комерції. Продажі через Facebook та інші соціальні платформи характеризуються простотою розміщення оголошень, але не забезпечують належного рівня автоматизації процесів, контролю залишків товарів та управління замовленнями.

1.3. Вибір архітектури платформи

Вебзастосунок MotoMarket PRO побудований на основі класичної багаторівневої клієнт-серверної архітектури, яка забезпечує чітке розмежування між рівнем представлення даних, рівнем серверної бізнес-логіки та рівнем зберігання інформації. Використання такого підходу дозволяє зробити структуру програмного забезпечення більш зрозумілою, спростує процес розробки та подальшого супроводу системи. Крім того, дана архітектура дає можливість простежити повний цикл обробки запиту в межах одного PHP-файлу: від отримання параметрів користувача та перевірки його прав доступу до виконання SQL-запитів і формування готової HTML-сторінки для відображення у браузері [1].

Клієнтський рівень відповідає за взаємодію користувача із системою та забезпечує відображення інформації у зручному вигляді. Для цього використовуються HTML-сторінки, стилізовані за допомогою CSS, а також

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

мінімальний набір JavaScript-скриптів, необхідних для реалізації мобільного меню, підтвердження небезпечних дій, адаптивної сітки, асинхронного оновлення окремих елементів інтерфейсу та базової доступності [12, 13, 14, 15, 16, 23, 29, 30]. Серверний рівень реалізовано мовою програмування PHP, яка виконує обробку запитів користувачів, маршрутизацію між сторінками, перевірку сесій, валідацію введених даних та формування SQL-запитів через інтерфейс PDO [2]. Рівень даних представлений реляційною системою управління базами даних MySQL, що забезпечує надійне зберігання інформації, підтримку зовнішніх ключів для збереження цілісності зв'язків між таблицями та можливість використання транзакцій під час виконання критично важливих операцій [8, 9].

Використання серверного рендерингу HTML є доцільним і повністю виправданим рішенням для даного проєкту. Такий підхід дозволяє значно спростити архітектуру системи, оскільки відсутня необхідність у створенні окремого REST API, використанні сучасних frontend-фреймворків або налаштуванні додаткових інструментів збірки проєкту. Альтернативні підходи, зокрема розробка односторінкових застосунків (SPA) на основі React чи Vue з окремим серверним бекендом, потребують складнішої інфраструктури, більшої кількості ресурсів та додаткового часу на підтримку. Для невеликої системи, орієнтованої на роботу одного продавця та адміністратора, подібні рішення є надмірними. Натомість використання PHP із серверним формуванням сторінок забезпечує простоту розгортання, сумісність із більшістю хостинг-платформ та можливість швидкого запуску застосунку навіть у локальному середовищі Open Server без необхідності виконання додаткових налаштувань [25, 27, 28].

Публічна частина системи та адміністративна панель функціонують у межах єдиної програмної структури та використовують спільне підключення до бази даних. Водночас між ними реалізовано механізми розмежування доступу, які гарантують безпеку роботи з даними та захист адміністративних функцій від несанкціонованого використання. Для перевірки факту

					ПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

авторизації користувача використовується функція `is_logged_in()`, яка визначає наявність активної сесії. Додатково функція `is_admin()` перевіряє роль користувача та підтверджує наявність адміністративних прав доступу відповідно до запису в таблиці `users`, а робота із сесією виконується в єдиному місці, що узгоджується з базовими рекомендаціями щодо `session management` [17, 18]. Такий підхід забезпечує централізоване управління доступом та спрощує контроль прав користувачів у системі.

Крім того, обрана архітектура забезпечує достатній рівень продуктивності для вебзастосунку даного типу. Основна обробка даних виконується на сервері, що дозволяє зменшити навантаження на клієнтські пристрої та забезпечити коректну роботу системи навіть на малопотужних комп'ютерах або мобільних пристроях. Завдяки цьому вебзастосунок MotoMarket PRO є зручним, надійним та придатним для подальшого розвитку й впровадження в реальних умовах використання.

З метою підвищення зручності супроводу програмного коду всі спільні компоненти вебзастосунку винесені до окремих файлів. До них належать модулі підключення до бази даних, конфігураційні параметри системи, елементи шапки та підвалу вебсторінок, а також інші багаторазово використовувані компоненти інтерфейсу. Основні налаштування зосереджені у файлі `config.php`, а допоміжні модулі розміщені в каталозі `includes`. Подібна організація структури проєкту дозволяє зменшити дублювання коду, підвищити його читабельність та значно спростити внесення змін у майбутньому [6]. Обробка запитів виконується окремими РНР-сценаріями, а спільні шаблони лише формують повторно використовувані частини інтерфейсу. Загальна схема взаємодії всіх модулів, рівнів та компонентів вебзастосунку MotoMarket PRO наведена на рисунку (рис. 1.4), а відповідна діаграма розгортання компонентів наведена на рисунку (рис. 3.1). Основні архітектурні компоненти системи, їх роль та обґрунтування вибору наведено у таблиці (табл. 1.3).

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

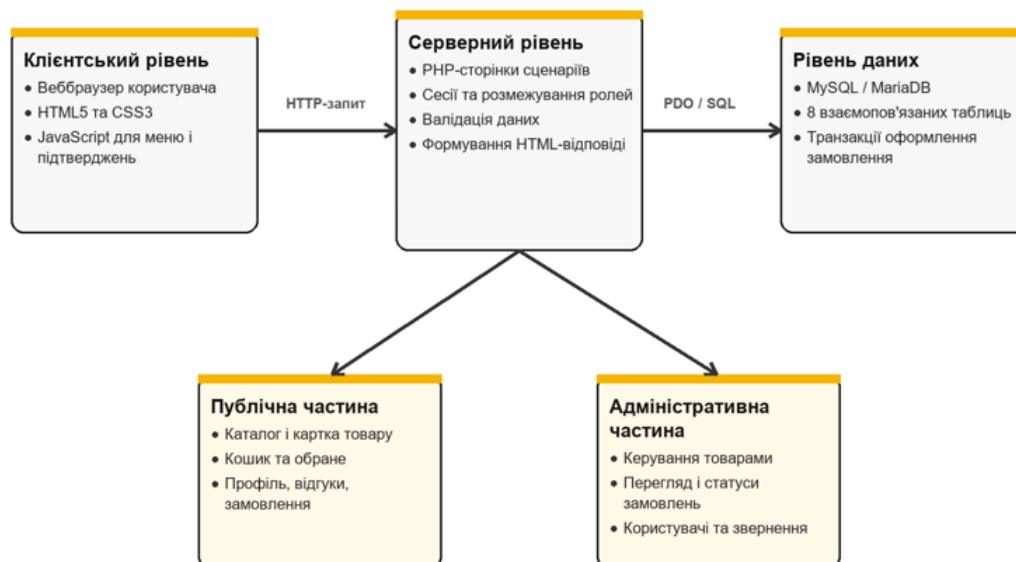


Рис. 1.4 – Загальна архітектура MotoMarket PRO

Таблиця 1.3

Архітектурні компоненти та їх обґрунтування

Компонент	Роль у системі	Причина вибору
HTML5	Структура сторінок і форм	Стандартна підтримка всіма браузерами
CSS3	Адаптивне оформлення інтерфейсу	Не потребує зовнішнього UI-фреймворку, повний контроль над стилем
JavaScript	Мобільне меню та підтвердження дій	Достатньо для невеликих клієнтських інтеракцій
PHP	Серверна бізнес-логіка, сесії, маршрутизація	Природна інтеграція з Apache, доступне середовище розгортання
PDO	Уніфікований доступ до MySQL	Підготовлені запити, єдиний інтерфейс, захист від SQL-ін'єкцій
MySQL	Реляційне зберігання всіх даних	Зовнішні ключі, транзакції, індекси, широка підтримка хостингів

Наведені в таблиці компоненти формують цілісну архітектуру вебзастосунку та забезпечують узгоджену взаємодію між усіма рівнями системи. Кожна технологія виконує чітко визначену роль, що дозволяє уникнути надлишкової складності та зберегти зрозумілу структуру програмного продукту.

1.4. Обґрунтування вибору інструментальних засобів

Головним критерієм під час вибору кожного програмного інструменту для розробки вебзастосунку MotoMarket PRO була його практична доцільність та здатність ефективно вирішувати конкретні завдання проєкту. При цьому перевага надавалася технологіям, які дозволяють реалізувати необхідний функціонал найпростішим та найбільш зрозумілим способом без використання зайвих компонентів або складних програмних рішень. Такий підхід дозволив зберегти простоту архітектури системи, забезпечити її стабільну роботу та спростити подальший супровід програмного продукту.

Мову програмування PHP було обрано як основу серверної частини вебзастосунку завдяки її широкому поширенню, простоті використання та природній інтеграції з вебсервером Apache. Важливою перевагою також є наявність великої кількості вбудованих засобів для роботи з HTTP-запитами, формами користувачів, файлами cookie та механізмами сесій. Крім того, PHP входить до складу популярних локальних середовищ розробки, зокрема Open Server, що значно спрощує процес створення та тестування вебзастосунку. Використання цієї мови дозволяє реалізувати повний цикл обробки запиту в межах одного програмного файлу, починаючи від отримання даних користувача і завершуючи формуванням готової HTML-сторінки. Завдяки цьому логіка роботи системи залишається зрозумілою та зручною для аналізу, що особливо важливо в межах навчального проєкту.

Для взаємодії із системою управління базами даних використовується розширення PDO, яке виступає єдиним інтерфейсом доступу до MySQL. Однією з головних переваг PDO є підтримка підготовлених запитів через механізм `prepare()` та `execute()`, що дозволяє автоматично відокремлювати

					ПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

SQL-команди від користувацьких даних і значно підвищує рівень безпеки застосунку. Такий підхід практично виключає можливість успішного проведення SQL-ін'єкцій. Додатково використовується режим PDO::ERRMODE_EXCEPTION, який забезпечує генерацію винятків при виникненні помилок під час виконання запитів. Це дозволяє централізовано обробляти помилки за допомогою конструкції try/catch та своєчасно реагувати на нестандартні ситуації [2, 3]. Альтернативним рішенням могло бути використання розширення mysqli, однак PDO має більш універсальний характер і забезпечує єдиний програмний інтерфейс для роботи з різними СУБД, що потенційно спрощує подальшу модернізацію системи.

Як систему керування базами даних обрано MySQL, оскільки структура предметної області природно відповідає реляційній моделі даних. Основні сутності системи, такі як користувачі, товари, категорії, кошики, замовлення та відгуки, мають чітко визначені взаємозв'язки, які ефективно реалізуються за допомогою зовнішніх ключів. Такий підхід дозволяє підтримувати цілісність даних та уникати виникнення логічних помилок у базі даних. Особливе значення має підтримка механізму транзакцій у рушії InnoDB. Завдяки транзакціям процес оформлення замовлення може виконуватися як єдина атомарна операція, що включає створення запису про замовлення, оновлення залишків товару на складі та очищення кошика покупця. Якщо на будь-якому етапі виникає помилка, система виконує операцію rollback(), повертаючи базу даних до попереднього узгодженого стану [8, 9, 10]. Це забезпечує надійність роботи застосунку та захист від втрати даних.

Для безпечного зберігання облікових даних користувачів використовується функція password_hash() із застосуванням алгоритму bcrypt. У результаті цього в базі даних зберігаються не самі паролі, а лише їхні криптографічні хеші. Перевірка правильності введеного пароля здійснюється за допомогою функції password_verify(), яка порівнює введене значення з хешем, що міститься в базі даних. Такий підхід відповідає сучасним рекомендаціям щодо інформаційної безпеки та дозволяє значно знизити

					ПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

ризика компрометації облікових записів користувачів [5]. Для підтримки стану авторизації використовується стандартний механізм сесій PHP, у якому зберігаються ідентифікатор користувача, його ім'я та роль доступу. Це дозволяє ефективно контролювати права користувачів під час роботи із системою [6].

Користувацький інтерфейс вебзастосунку було реалізовано без використання сторонніх CSS-фреймворків. Таке рішення дозволило створити унікальний дизайн із чорно-жовтою кольоровою схемою, яка відповідає загальній концепції проєкту. Відмова від використання готових бібліотек дала можливість повністю контролювати структуру сторінок, їх зовнішній вигляд та поведінку на різних пристроях. Для побудови складніших елементів інтерфейсу використовується технологія CSS Grid, яка застосовується при створенні каталогу товарів, статистичних блоків адміністративної панелі та інших компонентів. Адаптація інтерфейсу до мобільних пристроїв реалізована за допомогою медіазапитів, які автоматично змінюють розташування елементів залежно від ширини екрана користувача.

Під час реалізації програмного продукту особлива увага приділялася питанням інформаційної безпеки. У проєкті враховано базові рекомендації OWASP щодо автентифікації, керування сесіями, захисту від XSS та SQL-ін'єкцій, що забезпечує базовий рівень захисту користувацьких даних і внутрішніх ресурсів системи [17, 18, 20, 21].

Разом із тим поточна версія системи має навчальний характер, тому окремі механізми безпеки можуть бути розширені в майбутньому. Зокрема, доцільним є впровадження CSRF-токенів для захисту всіх HTML-форм від підроблених запитів, а також перенесення конфіденційних налаштувань, паролів та ключів доступу до окремих змінних середовища (.env). Крім того, у перспективі можливе впровадження додаткових механізмів журналювання подій, моніторингу активності користувачів та розширеного аудиту безпеки системи [19].

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

Таблиця 1.4

Порівняння серверних технологій для реалізації вебзастосунку

Критерій	PHP + MySQL	Node.js + PostgreSQL	Django + PostgreSQL
Складність запуску	Низька	Середня	Середня
Серверний рендеринг	Природний, без збірника	Потребує шаблонізатора	Вбудовані шаблони
Навчальна прозорість	Висока один файл = один сценарій	Середня	Середня
Локальне розгортання	Open Server, без налаштувань	Потребує npm, збірки	Потребує virtualenv, pip
Підсумок	Обрано	Не обрано	Не обрано

Практичне застосування обраних технологій та їх роль у реалізації вебзастосунку MotoMarket PRO наведено у таблиці (табл. 1.5).

Таблиця 1.5

Використані інструменти та їх практичне обґрунтування

Інструмент	Де використано	Практичне обґрунтування
PHP	Уся серверна логіка, сесії, форми	Природна інтеграція з Apache, доступне середовище
PDO	Усі SQL-запити до MySQL	Підготовлені запити, захист від SQL-ін'єкцій, єдиний API доступу до БД
MySQL / InnoDB	Зберігання всіх даних системи	Підтримка транзакцій, зовнішніх ключів та цілісності даних
password_hash()	Реєстрація та зміна пароля	Всгурт-хешування без зберігання відкритих паролів
CSS Grid	Каталог товарів, адміністративна панель, форми	Створення адаптивної сітки без використання зовнішніх фреймворків
Open Server	Локальне розгортання та тестування системи	Пакет Apache, PHP і MySQL без необхідності ручного налаштування

Важливою перевагою обраного технологічного стеку є відсутність залежності від складної інфраструктури або великої кількості зовнішніх компонентів. Крім того, використані інструменти мають широкую документацію, активну спільноту користувачів та тривалий досвід практичного застосування у сфері веброзробки.

Висновки до першого розділу

Проведено дослідження предметної області та особливостей діяльності сучасних мотوماгазинів. Визначено основні проблеми традиційних підходів до продажу мототоварів, зокрема складність отримання актуальної інформації про продукцію, обмежені можливості пошуку та недостатній рівень автоматизації. На основі аналізу сформовано основні вимоги до майбутнього вебзастосунку та визначено його ключові функціональні можливості.

Виконано аналіз існуючих рішень у сфері продажу мототоварів, серед яких універсальні маркетплейси, спеціалізовані інтернет-магазини та платформи продажу в соціальних мережах. Проведене порівняння дозволило визначити їхні переваги й недоліки, а також сформуванати перелік функцій, необхідних для реалізації власного програмного продукту.

Обґрунтовано вибір архітектури вебзастосунку MotoMarket PRO та доцільність використання класичної трирівневої моделі побудови системи. Визначено основні компоненти архітектури, принципи їх взаємодії та механізми розмежування доступу між користувачами й адміністраторами, що забезпечують надійність і зручність супроводу системи.

Також виконано обґрунтування вибору інструментальних засобів для реалізації вебзастосунку. Розглянуто можливості PHP, MySQL, PDO, HTML5, CSS3 та JavaScript, а також основні механізми забезпечення безпеки даних. Проведений аналіз підтвердив доцільність використання обраного технологічного стеку для реалізації поставлених завдань та подальшої розробки системи.

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА MOTOMARKET PRO

2.1. Визначення варіантів використання та структури системи

Перш ніж проєктувати базу даних і реалізовувати модулі, необхідно чітко визначити, хто користується системою і що саме кожен із них може робити. У MotoMarket PRO передбачено три рівні взаємодії: відвідувач, зареєстрований покупець та адміністратор. Межі між ролями закладені не лише на рівні інтерфейсу, а безпосередньо в серверній логіці через перевірки сесії та ролі.

Відвідувач це будь-яка людина, яка відкрила сайт без авторизації. Йому доступний повний перегляд каталогу, пошук і фільтрація товарів за категорією, брендом і ціною, сортування результатів, перегляд детальної картки товару з описом і рейтингом, а також надсилання контактного звернення через форму. Операції, що створюють персональні записи, для відвідувача недоступні система перенаправляє його на сторінку входу.

Покупець це зареєстрований користувач із роллю user. Після входу він успадковує всі можливості відвідувача і додатково отримує доступ до персональних сценаріїв: керування профілем із збереженням телефону та адреси доставки, формування списку бажань, роботи з постійним кошиком, оформлення замовлення з вибором способу доставки та оплати, перегляду історії власних замовлень і залишення відгуків на куплені товари. Всі записи покупця прив'язані до його user_id, тому різні користувачі працюють виключно зі своїми даними [6].

Адміністратор це обліковий запис із роллю admin у таблиці users. Він отримує доступ до окремого захищеного каталогу admin/, недоступного для покупців. Його функції охоплюють повний CRUD товарів із контролем залишків, перегляд усіх замовлень і зміну їхніх статусів, перегляд зареєстрованих користувачів та обробку контактних звернень. Будь-яка спроба відкрити адміністративну сторінку без відповідної ролі завершується перенаправленням на сторінку входу [17], наведено у таблиці (табл. 2.1).

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

Таблиця 2.1

Ролі та їхні можливості в системі

Актор	Права та можливості	Технічна реалізація
Відвідувач	Каталог, пошук, фільтри, картка товару, контактна форма	Доступ без сесії
Покупець	Профіль, кошик, обране, checkout, відгуки, замовлення	is_logged_in(), \$_SESSION['user_id']
Адміністратор	Товари, замовлення, користувачі, звернення	is_admin(), role = 'admin'

Після визначення ролей необхідно описати конкретні сценарії взаємодії. Кожен такий сценарій складається з трьох елементів: актор, який виконує дію, обов'язкова передумова для початку та результат, який система зберігає або повертає.

Реєстрація старт для кожного нового покупця. Користувач заповнює форму з іменем, email і паролем. Сервер перевіряє унікальність email, хешує пароль через password_hash() і створює запис у таблиці users з роллю user. Роль примусово встановлюється сервером, тому відвідувач не може самостійно зареєструвати адміністративний обліковий запис [5].

Вхід в акаунт відбувається через форму на login.php. Користувач вводить email і пароль, сервер знаходить запис за email через підготовлений PDO-запит і перевіряє пароль функцією password_verify(). У разі успіху до сесії записуються user_id, ім'я та роль, які визначають доступні функції та сторінку перенаправлення [5, 6].

Додавання товару до кошика виконується через add-to-cart.php. Перед збереженням сервер перевіряє автентифікацію, існування товару та актуальний залишок stock. Якщо така позиція вже є в кошику, кількість збільшується в межах доступного залишку; інакше створюється новий запис у таблиці cart [8].

Оформлення замовлення найбільш критичний сценарій системи. Checkout.php виконує всі операції у межах однієї транзакції PDO: створює запис замовлення в orders, додає позиції до order_items, зменшує залишок кожного товару в products та очищає кошик користувача. Якщо будь-яка операція завершується помилкою, виконується rollBack() і база повертається до початкового стану [4, 10].

Відгук на товар може залишити будь-який авторизований користувач. Сервер перевіряє діапазон рейтингу від 1 до 5 і непорожній текст коментаря, після чого додає запис до таблиці reviews із поточним user_id і датою, наведено у таблиці (табл. 2.2). Діаграму варіантів використання MotoMarket PRO наведено на рисунку (рис. 2.1).

Таблиця 2.2

Варіанти використання системи MotoMarket PRO

Варіант використання	Актор	Передумова	Результат
Реєстрація	Відвідувач	Унікальний email, валидні поля	Створено запис users, відкрито сесію
Вхід	Користувач	Існуючий email, правильний пароль	Встановлено сесію, виконано redirect
Перегляд каталогу	Будь-хто	—	Відображено відфільтровані товари
Додавання до кошика	Покупець	Автентифікація, товар у наявності	Створено або оновлено запис cart
Додавання до обраного	Покупець	Автентифікація	Створено або видалено запис wishlist
Оформлення замовлення	Покупець	Непорожній кошик, заповнені поля	Створено orders та order_items, оновлено stock, очищено cart
Залишення відгуку	Покупець	Автентифікація, коректний рейтинг	Створено запис reviews
Редагування товару	Адміністратор	Роль admin	Оновлено запис products

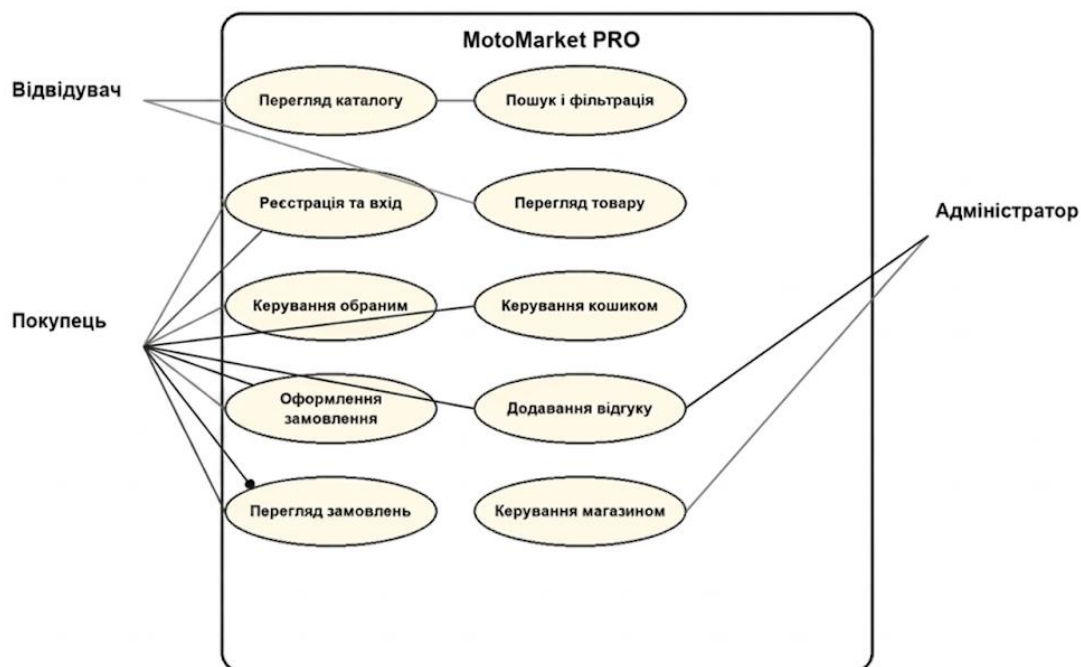


Рис. 2.1 – Діаграма варіантів використання MotoMarket PRO

Діаграма варіантів використання відображає основні сценарії взаємодії користувачів із системою та демонструє функціональні можливості MotoMarket PRO. За допомогою даної діаграми було сформовано загальне уявлення про структуру майбутньої системи, визначено основних учасників процесів та перелік операцій, які вони можуть виконувати відповідно до своїх повноважень. Крім того, діаграма дозволяє наочно представити межі системи, взаємозв'язки між окремими функціями та особливості реалізації бізнес-логіки. Її використання на етапі проєктування сприяє кращому розумінню вимог замовника та забезпечує узгодженість між функціональними потребами користувачів і технічними рішеннями, що впроваджуються в програмному продукті. Дотримання зазначених вимог забезпечує стабільне функціонування MotoMarket PRO та створює необхідні умови для ефективного використання системи в реальних умовах експлуатації. Для подальшого посилення основним сценаріям покупця присвоєно такі ідентифікатори: UC-Auth (реєстрація та автентифікація), UC-Catalog (перегляд тематичного каталогу), UC-Search (пошук і фільтрація товарів), UC-Cart (керування кошиком), UC-Wishlist

(керування списком бажань) та UC-Review (додавання відгуку); ключовій сутності бази даних товарів відповідає позначення ERD-Products, наведено у таблиці (табл. 2.3).

Таблиця 2.3

Нефункціональні вимоги до системи MotoMarket PRO

Критерій	Вимога	Очікуваний результат
Сприйняття	Інтерфейс зрозумілий без додаткового навчання	Освоєння основних функцій покупцем до 10 хв
Надійність	Стабільна обробка замовлень і операцій із кошиком	Транзакційна атомарність checkout, відсутність втрат даних
Цілісність даних	Залишки товарів завжди відповідають фактичним	Оновлення stock у межах тієї самої транзакції, що й створення замовлення
Безпека	Захист від несанкціонованого доступу та ін'єкцій	PDO prepared statements, перевірка ролі на кожній сторінці
Розгортання	Запуск без складного середовища збірки	Копіювання файлів + імпорт SQL достатньо для старту

Наведені в таблиці 2.3 нефункціональні вимоги визначають ключові характеристики якості вебзастосунку MotoMarket PRO та впливають на вибір архітектурних і технологічних рішень під час його розробки. Забезпечення зрозумілого інтерфейсу спрямоване на швидке освоєння системи користувачами без необхідності додаткового навчання. Вимоги до надійності та цілісності даних реалізуються за рахунок використання транзакційного механізму обробки замовлень і контролю складських залишків. Безпека системи досягається шляхом застосування підготовлених SQL-запитів, механізмів автентифікації та розмежування прав доступу користувачів.

2.2. Проектування бази даних

Реляційна база даних є центральним елементом MotoMarket PRO, оскільки вся бізнес-логіка системи від формування каталогу до оформлення замовлення базується на узгодженому стані даних у взаємопов'язаних таблицях. При проектуванні структури ставилася мета відобразити реальні предметні зв'язки між сутностями, забезпечити цілісність даних на рівні СУБД і мінімізувати надлишковість [8, 9].

База даних містить вісім таблиць. Центральними сутностями є `users` і `products`. Перша описує облікові записи, контактні дані та ролі користувачів, друга повний каталог товарів із усіма атрибутами, необхідними для відображення, фільтрації та контролю залишків. Усі інші таблиці утворюють зв'язки навколо цих двох центральних сутностей.

Таблиці `cart` та `wishlist` реалізують персональні зв'язки між користувачем і товаром. `Cart` зберігає поточний вміст кошика з кількістю кожної позиції, яка перевіряється відносно актуального `stock` під час додавання та оновлення. `Wishlist` реалізує список бажань із обмеженням `UNIQUE(user_id, product_id)`, що унеможливорює дублювання однієї позиції в обраному одного користувача. Таблиця `reviews` доповнює зв'язок між покупцем і товаром числовим рейтингом від 1 до 5 та текстовим коментарем [8].

Замовлення свідомо розділено на дві таблиці: заголовок `orders` і позиції `order_items`. Така структура є загальноприйнятою практикою проектування систем електронної торгівлі, оскільки одне замовлення може містити довільну кількість товарів [27]. У таблиці `orders` зберігаються контактні дані покупця, спосіб доставки, спосіб оплати, коментар, загальна сума та поточний статус. У таблиці `order_items` кожен рядок відповідає одній товарній позиції з фіксованою кількістю та ціною на момент оформлення. Копіювання ціни до `order_items` є принциповим рішенням: подальша зміна ціни в каталозі не впливає на історичну вартість уже оформлених замовлень, що забезпечує коректність фінансової звітності.

Таблиця contacts спроектована як незалежна сутність без зовнішніх ключів, оскільки контактне звернення може надіслати будь-який відвідувач сайту без реєстрації. Вона зберігає ім'я, email, тему та текст повідомлення і використовується виключно адміністратором для обробки вхідних запитів.

Усі залежні таблиці використовують зовнішні ключі з директивою ON DELETE CASCADE [9]. Це означає, що при видаленні користувача автоматично видаляються його записи кошика, обраного, замовлень і відгуків. При видаленні товару так само очищаються пов'язані позиції кошика, обраного та замовлень. Такий підхід забезпечує референційну цілісність на рівні СУБД без необхідності додаткової логіки на рівні PHP [9].

Вся схема створюється єдиним файлом database.sql. Файл спочатку видаляє залежні таблиці у безпечній послідовності від дочірніх до батьківських, щоб уникнути конфліктів зовнішніх ключів, потім створює повну структуру, додає два демонстраційні облікові записи з хешованими паролями, каталог із 25 товарами різних категорій, тестові відгуки та контактне звернення. Кодування utf8mb4 для всіх таблиць і полів забезпечує коректне зберігання українського тексту, спеціальних символів і емої [11]. Структуру бази даних наведено на рисунку (рис. 2.2).

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

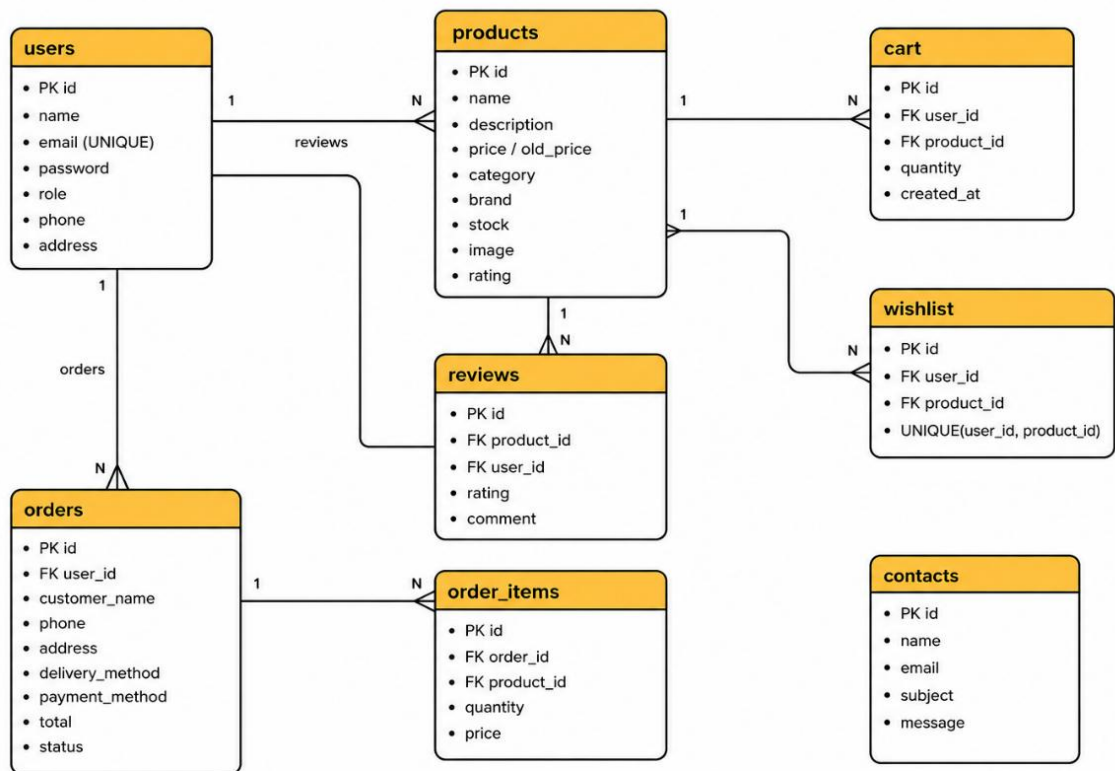


Рис 2.2 – ER-модель бази даних MotoMarket PRO

Склад таблиць бази даних, їх призначення та основні зв'язки наведено у (табл. 2.4). Реляційна структура побудована таким чином, щоб відобразити ключові сутності предметної області інтернет-магазину та забезпечити коректну взаємодію між користувачами, товарами та процесом оформлення замовлень. Центральними сутностями виступають таблиці users та products, навколо яких формуються всі інші зв'язки системи.

Таблиця 2.4

Таблиці бази даних та їх призначення

Таблиця	Призначення	Ключові зв'язки
users	Облікові записи та ролі	1:N з cart, wishlist, orders, reviews
products	Каталог і залишки	1:N з cart, wishlist, order_items, reviews
cart	Поточні покупки користувача	FK user_id, product_id

продовж. табл. 2.4

wishlist	Список бажань	UNIQUE(user_id, product_id), FK обидва
orders	Загальні дані замовлення	FK user_id
order_items	Позиції замовлення з фіксованою ціною	FK order_id, product_id
reviews	Оцінки та текстові коментарі	FK product_id, user_id
contacts	Контактні звернення відвідувачів	Незалежна таблиця

Структура таблиці users наведена у (табл. 2.5). Дана таблиця є основною для ідентифікації користувачів системи та реалізації механізмів авторизації, а також розмежування доступу між різними ролями. Саме ця таблиця забезпечує персоналізацію роботи користувача в системі, оскільки дозволяє зберігати контактні дані, такі як номер телефону та адреса доставки, які автоматично підтягуються під час оформлення замовлення. Крім того, зв'язки цієї таблиці з іншими сутностями бази даних дозволяють відстежувати історію дій користувача, включаючи замовлення, відгуки та використання функціоналу кошика.

Таблиця 2.5

Структура таблиці users

Поле	Тип	Призначення
id	INT, PK, AUTO_INCREMENT	Унікальний ідентифікатор користувача
name	VARCHAR(120), NOT NULL	Ім'я для відображення в інтерфейсі та замовленнях
email	VARCHAR(160), UNIQUE	Логін користувача, унікальний у межах системи
password	VARCHAR(255), NOT NULL	Хеш пароля, сформований password_hash()
phone	VARCHAR(40)	Контактний номер для замовлень
address	TEXT	Збережена адреса доставки для автозаповнення checkout

Таблиця products, наведена у (табл. 2.6), є основою каталогу товарів системи MotoMarket PRO. Вона містить повний набір атрибутів, необхідних для відображення товару в інтерфейсі, його фільтрації, сортування та контролю складських залишків. Окремі поля забезпечують підтримку маркетингових функцій, таких як знижки та рейтинги.

Таблиця 2.6

Структура таблиці products

Поле	Тип	Призначення
id	INT, PK, AUTO_INCREMENT	Унікальний ідентифікатор товару
name	VARCHAR(180), NOT NULL	Назва товару для відображення і пошуку
description	TEXT, NOT NULL	Детальний опис для картки товару
price	DECIMAL(10,2), NOT NULL	Поточна ціна продажу
old_price	DECIMAL(10,2), NULL	Попередня ціна для відображення знижки
category	VARCHAR(80), NOT NULL	Категорія для фільтрації каталогу
brand	VARCHAR(80), NOT NULL	Бренд для фільтрації та відображення
stock	INT, NOT NULL	Кількість одиниць на складі
image	VARCHAR(160)	Ім'я файлу зображення товару
badge	VARCHAR(40)	Мітка на картці: «Хіт», «Новинка», «Акція» тощо
rating	DECIMAL(2,1)	Рейтинг для відображення на картці товару

Структура таблиці orders наведена у (табл. 2.7). Вона відповідає за збереження інформації про оформлені замовлення та є ключовою для реалізації бізнес-процесу покупки. Таблиця містить як дані користувача на момент оформлення, так і параметри доставки та оплати, що дозволяє зберігати історичну точність навіть у разі подальшої зміни профілю користувача.

Структура таблиці orders

Поле	Тип	Призначення
id	INT, PK, AUTO_INCREMENT	Унікальний номер замовлення
user_id	INT, FK	Посилання на покупця в таблиці users
customer_name	VARCHAR(120)	Ім'я отримувача на момент оформлення
phone	VARCHAR(40)	Контактний телефон для доставки
address	TEXT	Адреса доставки
delivery_method	VARCHAR(80)	Обраний спосіб доставки
payment_method	VARCHAR(80)	Обраний спосіб оплати
comment	TEXT	Довільний коментар покупця до замовлення
total	DECIMAL(10,2)	Загальна сума замовлення
status	VARCHAR(40)	Поточний статус: «Нове», «В обробці», «Відправлено» тощо

Наведені в таблицях 2.4–2.7 характеристики демонструють структуру основних сутностей бази даних MotoMarket PRO та їхнє призначення в межах функціонування системи. Центральне місце займають таблиці users, products та orders, які забезпечують зберігання інформації про користувачів, каталог товарів і процес оформлення замовлень. Допоміжні таблиці cart, wishlist, reviews, order_items і contacts реалізують додаткові функціональні можливості вебзастосунку та забезпечують підтримку бізнес-процесів мотомагазину.

2.3. Проєктування та реалізація алгоритмів роботи

Логіка роботи MotoMarket PRO не зосереджена в одному модулі, а розподілена між окремими PHP-файлами відповідно до їхньої відповідальності. Алгоритм автентифікації реалізовано в login.php, логіка кошика в add-to-cart.php та cart.php, а критичний алгоритм оформлення замовлення повністю інкапсульований у checkout.php. Такий підхід дозволяє змінювати код одного сценарію без ризику вплинути на інші, а також робить кожен файл зрозумілим для самостійного аналізу [1].

Алгоритм автентифікації починається з отримання email із POST-запиту та пошуку відповідного запису в таблиці users через підготовлений PDO-запит

[2, 3]. Якщо запис знайдено, функція password_verify() порівнює введений пароль із хешем у базі [5]. У разі успіху до сесії записуються три значення: user_id, ім'я користувача та його роль. Саме роль визначає сторінку перенаправлення після входу адміністратор потрапляє до admin/index.php, покупець до головної сторінки. Якщо пошук не повернув запису або хеш не збігся, користувач отримує повідомлення про помилку без уточнення причини, що унеможлиблює визначення існування акаунту методом перебору [17].

Реєстрація виконує мінімальну серверну перевірку: обов'язковість заповнення полів, мінімальна довжина пароля та унікальність email. Роль нового користувача примусово встановлюється як user незалежно від будь-яких вхідних даних, тому відвідувач не може самостійно зареєструвати адміністративний обліковий запис [5].

Діаграму активності процесу автентифікації та реєстрації користувачів наведено на рисунку (рис. 2.3).

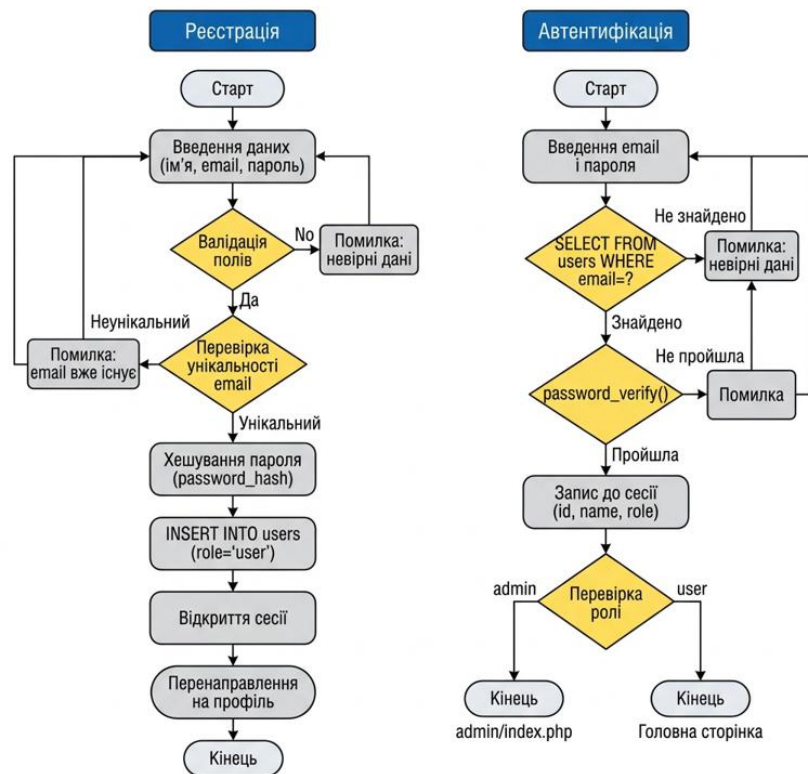


Рис. 2.3 – Схема процесу реєстрації та автентифікації користувачів

Алгоритм формування каталогу динамічно будує SQL-запит залежно від переданих GET-параметрів: search, category, brand, min_price та max_price. Для кожного непорожнього параметра до масиву умов WHERE додається відповідний вираз, а значення до масиву параметрів для передачі в prepare/execute [2, 3]. Текстовий пошук охоплює три поля одночасно: назву товару, опис і бренд через конструкцію LIKE з символом підстановки. Поле ORDER BY обирається виключно з наперед визначеного асоціативного масиву допустимих варіантів сортування, тому довільне значення від користувача не потрапляє до SQL-запиту без перевірки [21]. Такий whitelist-підхід є рекомендованою практикою захисту від ін'єкцій у динамічних частинах запиту, які не можна параметризувати стандартним способом.

Діаграму активності алгоритму пошуку та фільтрації каталогу наведено на рисунку (рис. 2.4).

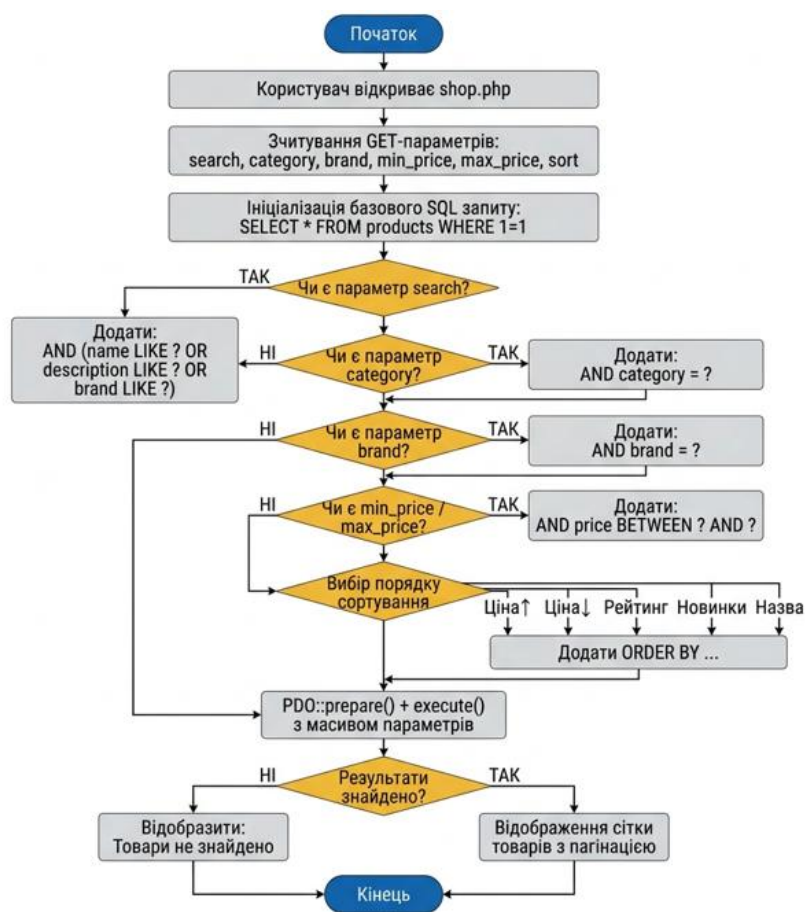


Рис. 2.4 – Схема алгоритму пошуку і фільтрації товарів

Алгоритм роботи з кошиком складається з двох основних операцій: додавання товарної позиції та оновлення її кількості. Під час додавання товару сервер послідовно виконує три обов'язкові перевірки: наявність автентифікованої сесії користувача, існування відповідного товару в базі даних та перевірку його фактичної наявності на складі через значення поля `stock` [8]. Такий підхід дозволяє запобігти додаванню до кошика неіснуючих або недоступних товарів. Якщо вибраний товар уже присутній у кошику поточного користувача, система не створює новий запис, а збільшує кількість наявної позиції. При цьому максимальне значення кількості обмежується актуальним залишком товару на складі. Якщо ж відповідного запису в кошику не існує, створюється новий рядок у таблиці `cart` із посиланням на користувача та обраний товар. Сторінка `cart.php` повторно застосовує аналогічну перевірку під час ручної зміни кількості товару користувачем. Це забезпечує додатковий рівень контролю та унеможливорює оформлення замовлення на більшу кількість товару, ніж фактично доступна на складі, навіть у випадках, коли залишки змінилися після початкового додавання товару до кошика. Завдяки цьому забезпечується актуальність даних і зменшується ризик виникнення конфліктних ситуацій під час оформлення покупки.

Найбільш критичним з точки зору цілісності даних є алгоритм оформлення замовлення. Він повинен одночасно зафіксувати нове замовлення, зберегти всі його товарні позиції, скоригувати складські залишки та очистити кошик користувача таким чином, щоб усі операції були виконані як єдине ціле або ж не були виконані взагалі [4, 10]. Саме тому під час реалізації цього механізму особлива увага приділяється забезпеченню узгодженості даних, контролю можливих помилок і гарантуванню коректного виконання всіх взаємопов'язаних операцій. Крім того, алгоритм повинен враховувати можливість виникнення непередбачених ситуацій, таких як втрати з'єднання з базою даних, помилки запису або зміни залишків товарів у процесі оформлення покупки.

					ІІЗ.КР.Б – 121 – 26 - ІІЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Для цього всі дії виконуються в межах однієї транзакції PDO через метод `beginTransaction()`. Послідовність кроків включає валідацію вхідних даних форми, створення нового запису в таблиці `orders` із отриманням ідентифікатора замовлення через метод `lastInsertId()`, послідовне додавання всіх позицій кошика до таблиці `order_items` із фіксацією актуальної вартості товару на момент покупки, оновлення таблиці `products` для коригування складських залишків, очищення кошика користувача через видалення записів із таблиці `cart` та завершення транзакції за допомогою `commit()`. Використання транзакційного механізму забезпечує атомарність усіх операцій і гарантує, що база даних не перейде в неконсистентний стан. Якщо на будь-якому етапі виконання виникає помилка або виняткова ситуація, блок `catch` виконує метод `rollback()`, після чого всі зміни скасовуються, а база даних повертається до стану, який існував до початку транзакції [4]. Такий підхід дозволяє уникнути ситуацій, коли замовлення було створене лише частково, деякі товарні позиції не були збережені або залишки товарів перестали відповідати фактичним даним системи. У результаті забезпечується висока надійність роботи модуля замовлень і збереження цілісності всієї інформації, що обробляється вебзастосунком, наведено діаграму активності на рисунку (рис. 2.5).

					ПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

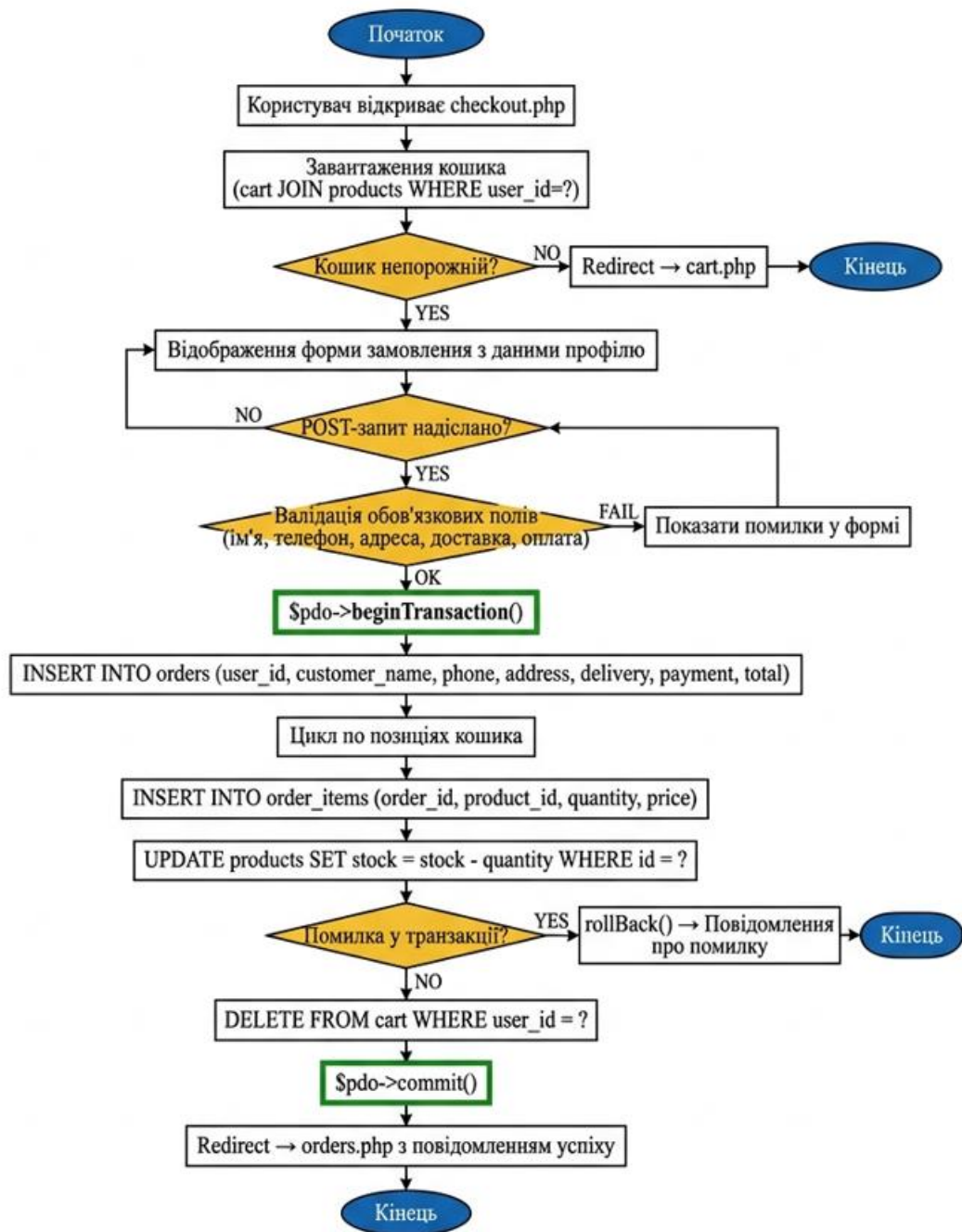


Рис. 2.5 – Схема алгоритму оформлення замовлення

Для забезпечення цілісності даних процес оформлення замовлення розділено на послідовні кроки. Основні етапи розробленого алгоритму та реалізовані на кожному етапі захисні механізми наведено у (табл. 2.8).

Таблиця 2.8

Етапи алгоритму оформлення замовлення та захисні механізми

Етап	Операція	Захисний механізм
1	Завантаження позицій кошика	Фільтр за user_id поточної сесії
2	Перевірка непорожнього кошика	Перенаправлення до cart.php

3	Валідація контактних даних	trim() і перевірка обов'язкових полів
4	beginTransaction()	Початок атомарної операції
5	INSERT до orders	Підготовлений запит із параметрами
6	INSERT до order_items у циклі	Фіксація ціни на момент оформлення
7	UPDATE stock у products	Зменшення залишку за product_id
8	DELETE з cart	Очищення кошика після фіксації
9	commit() або rollBack()	Атомарність усіх змін разом

Алгоритм роботи адміністративної панелі базується на обов'язковій перевірці ролі на початку кожного файлу каталогу admin/. Функція is_admin() перевіряє одночасно наявність активної сесії та значення role = 'admin' у \$_SESSION [17]. Будь-яка спроба відкрити адміністративну сторінку без відповідної ролі одразу завершується перенаправленням на login.php незалежно від того, чи є користувач авторизованим покупцем. Видалення товару додатково супроводжується клієнтським підтвердженням у JavaScript і повторною серверною перевіркою ролі, щоб унеможливити виконання деструктивної операції через підроблений запит [19].

2.4. Особливості програмної реалізації вебзастосунку

У цьому підрозділі детально розглянуто архітектурну логіку роботи, ключові алгоритми та специфіку програмного коду, який безпосередньо відповідає за забезпечення головних функцій серверної частини вебзастосунку MotoMarket PRO. Оскільки повний вихідний код розробленого проєкту наводиться у додатках до кваліфікаційної роботи, у цьому розділі зосереджено увагу виключно на найбільш важливих і критичних технічних рішеннях. До них належать концепція централізованого підключення до бази даних, реалізація спільних системних функцій, модуль інтерактивного каталогу з інтегрованими механізмами пошуку та динамічної фільтрації, алгоритм транзакційного оформлення замовлень, а також комплексний захист адміністративного контуру системи.

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

Конфігураційний файл `config.php` виступає головною точкою входу для всіх сторінок системи та виконує роль ядра застосунку. Програмне підключення створюється з обов'язковим атрибутом `PDO::ERRMODE_EXCEPTION`, що дозволяє ефективно перехоплювати будь-які помилки бази даних у блоці `try/catch`, а використання режиму `PDO::FETCH_ASSOC` гарантує отримання результатів вибірки у вигляді оптимізованих асоціативних масивів. Спеціалізована функція `safe()` автоматично застосовує вбудований інструмент `htmlspecialchars()` до будь-якого динамічного значення перед його безпосереднім виведенням у HTML-розмітку, що формує надійну єдину точку захисту від міжсайтового скриптингу (XSS) у всіх шаблонах системи [7, 20]. Водночас глобальні функції `is_logged_in()` та `is_admin()` повністю інкапсулюють у собі перевірку стану поточної сесії, завдяки чому кожна захищена сторінка робить один зрозумілий виклик замість прямого та небезпечного звернення до суперглобального масиву `$_SESSION`.

Спільні структурні шаблони `header.php` та `footer.php` виконують значно ширший спектр задач, ніж просте виведення статичної розмітки сайту. Шаблон шапки (`header`) при кожному завантаженні або оновленні сторінки виконує два легкі оптимізовані SQL-запити до бази даних: перший здійснює підрахунок наявних позицій у кошику, а другий визначає кількість товарів у списку бажань поточного користувача, після чого динамічно відображає ці лічильники у верхній навігаційній панелі. Крім того, цей модуль автоматично формує користувацьке меню залежно від поточного стану сесії. Звичайний гість бачить лише стандартні кнопки входу та реєстрації, авторизований покупець отримує посилання на персональний кабінет та кошик, а адміністратор системи додатковий захищений пункт переходу до панелі керування. Шаблон підвалу (`footer`) логічно завершує загальну HTML-структуру документів та здійснює підключення головного клієнтського файлу скриптів `script.js` [12].

Модуль інтерактивного каталогу складається з трьох окремих файлів, кожен з яких націлений на вирішення власних специфічних задач. Головна сторінка `index.php` виконує два окремі незалежні запити до бази даних: перший отримує акційні товари зі знижкою через порівняння умови `old_price IS NOT NULL`, а другий вибирає останні нові надходження за допомогою сортування за полем `created_at DESC` із жорстким обмеженням ліміту. Такий підхід дозволяє автоматично наповнювати головну сторінку сайту актуальним і свіжим вмістом без жодних додаткових ручних налаштувань з боку адміністратора [8]. Сторінка `shop.php` повністю реалізує складну логіку пошуку, сортування та фільтрації товарів через динамічне формування масиву умов `WHERE` залежно від переданих GET-параметрів. Кожне заповнене користувачем значення додає відповідну умову до загального тіла запиту, а всі вхідні дані передаються виключно через параметризований метод `execute()`, що повністю виключає можливість проведення SQL-ін'єкцій [21]. Сторінка детального перегляду `product.php` завантажує інформацію про конкретний товар за його унікальним ідентифікатором `id` та одним додатковим запитом отримує всі пов'язані відгуки з іменами їхніх авторів через об'єднання `JOIN` з таблицею користувачів `users`.

Важливою архітектурною особливістю реалізації модуля покупок є те, що критична перевірка залишку товарів на складі виконується системою двічі. Вперше це відбувається під час безпосереднього додавання товару до кошика у файлі `add-to-cart.php`, а повторно під час ручного чи автоматичного оновлення кількості у файлі `cart.php` [8]. Це гарантовано захищає систему від комерційних помилок і накладок, коли фактичний залишок товару встиг змінитися між двома відвідуваннями сторінки або під час тривалого перебування користувача на сайті. Модуль `wishlist-toggle.php` реалізує універсальний перемикач стану обраних товарів через один єдиний файл. Якщо відповідний запис у базі даних уже існує, система миттєво видаляє його, а якщо ні створює новий. Такий підхід дозволяє одній кнопці на сторінці

картки товару коректно виконувати обидві протилежні дії без дублювання програмного коду.

Найбільш технічно складним та відповідальним у системі є модуль оформлення замовлення. Файл `checkout.php` одночасно виконує дві важливі ролі, розділені за типом запиту: при отриманні GET-запиту він відображає інтерфейсну форму, заповнюючи її поточними даними профілю як початковими значеннями, а при отриманні POST-запиту запускає процес обробки та перевірки форми. Всі операції, пов'язані зі зміною або внесенням даних, виконуються виключно всередині ізольованої транзакції PDO [4, 10, 22].

Усі адміністративні файли та скрипти керування ізольовані й розміщені в окремому захищеному каталозі `admin/`, і кожен із них обов'язково починається з жорсткого виклику функції безпеки `is_admin()` [17]. Сторінка керування номенклатурою товарів реалізує повний класичний CRUD-цикл, що включає зручну форму додавання позицій, форму редагування з попередньо заповненими полями, а також механізм видалення. Видалення записів реалізовано з обов'язковим подвійним підтвердженням: клієнтським через інтерфейсне вікно JavaScript та серверним через повторну жорстку перевірку ролі поточного користувача [19]. Сторінка конкретного замовлення демонструє адміністратору повні контактні дані покупця, склад придбаних позицій і містить окрему форму для оперативної зміни статусу замовлення через безпечний POST-запит.

Увесь програмний код проєкту надійно зберігається у системі контролю версій Git із попередньо налаштованою локальною ідентичністю автора. Історія розробки чітко поділена на послідовні логічні коміти, які повністю відповідають завершеним функціональним блокам системи, таким як проєктування схеми бази даних, створення базової структури, написання модуля автентифікації, реалізація каталогу, кошика, транзакційного замовлення, адміністративної панелі та фінальна документація проєкту [24].

					ІІЗ.КР.Б – 121 – 26 - ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Висновки до другого розділу

Було визначено ролі користувачів системи MotoMarket PRO та описано основні сценарії їхньої взаємодії з вебзастосунком. Для відвідувача, зареєстрованого покупця та адміністратора було встановлено чіткий перелік доступних функцій і обмежень доступу. Також сформовано варіанти використання системи, що дозволило формалізувати бізнес-логіку та визначити межі відповідальності кожної ролі. Додатково було проаналізовано взаємозв'язки між ролями та відповідними функціональними модулями системи, що забезпечило більш чітке розуміння структури майбутнього програмного продукту.

Було виконано проєктування бази даних вебзастосунку MotoMarket PRO. Розроблена реляційна модель із восьми взаємопов'язаних таблиць забезпечує зберігання інформації про користувачів, товари, замовлення та супутні сутності. Використання зовнішніх ключів і каскадних зв'язків дозволило забезпечити цілісність даних і коректну взаємодію між усіма елементами системи. Крім того, було враховано принципи нормалізації даних, що мінімізує надлишковість інформації та підвищує ефективність роботи бази даних при виконанні запитів.

Розроблено основні алгоритми роботи вебзастосунку, що охоплюють процеси автентифікації, реєстрації, пошуку та фільтрації товарів, роботи з кошиком і оформлення замовлень. Також враховано сценарії обробки помилок і перевірки вхідних даних, що підвищує стабільність роботи системи в умовах некоректних запитів.

Розглянуто особливості програмної реалізації вебзастосунку MotoMarket PRO. Описано структуру основних модулів, принципи організації коду, централізоване підключення до бази даних, механізми роботи каталогу, кошика, оформлення замовлень та адміністративної панелі. Додатково наведено реалізацію базових підходів до безпеки, включаючи захист від SQL-ін'єкцій, XSS-атак та перевірку прав доступу.

					ПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ

3.1. Порядок встановлення та налаштування системи

MotoMarket PRO реалізовано на основі класичної клієнт-серверної архітектури з використанням вебсервера Apache, серверної мови PHP та СУБД MySQL. Для розробки та тестування проєкту використовувалося локальне середовище Open Server під керуванням операційної системи Windows. Водночас архітектура застосунку не залежить від конкретної операційної системи та може бути розгорнута як у середовищі Windows, так і на серверах Linux за наявності відповідних компонентів програмної платформи. На відміну від проєктів із мікросервісною архітектурою, система не потребує контейнеризації, складного процесу збірки чи налаштування кількох паралельних процесів. Для навчальної демонстрації достатньо пакету Open Server або аналогічного рішення, яке об'єднує Apache, PHP і MySQL в єдиному інсталяторі [25]. Після копіювання файлів проєкту та імпорту SQL-схеми система повністю готова до запуску.

Конфігурація підключення до бази даних зосереджена в одному файлі config.php. За замовчуванням використовуються хост localhost, база motomarketv2pro1, користувач root і послідовна спроба паролів root та порожній рядок для сумісності з різними профілями локального середовища [2]. Для реального розгортання ці значення необхідно винести до змінних середовища та надати окремому користувачу бази даних мінімально необхідні права замість використання root. Для наочного представлення фізичної структури проєкту та взаємозв'язків між його основними вузлами побудовано відповідну схему. Діаграму розгортання, яка відображає взаємодію між компонентами локального середовища, наведено на рисунку (рис. 3.1).

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

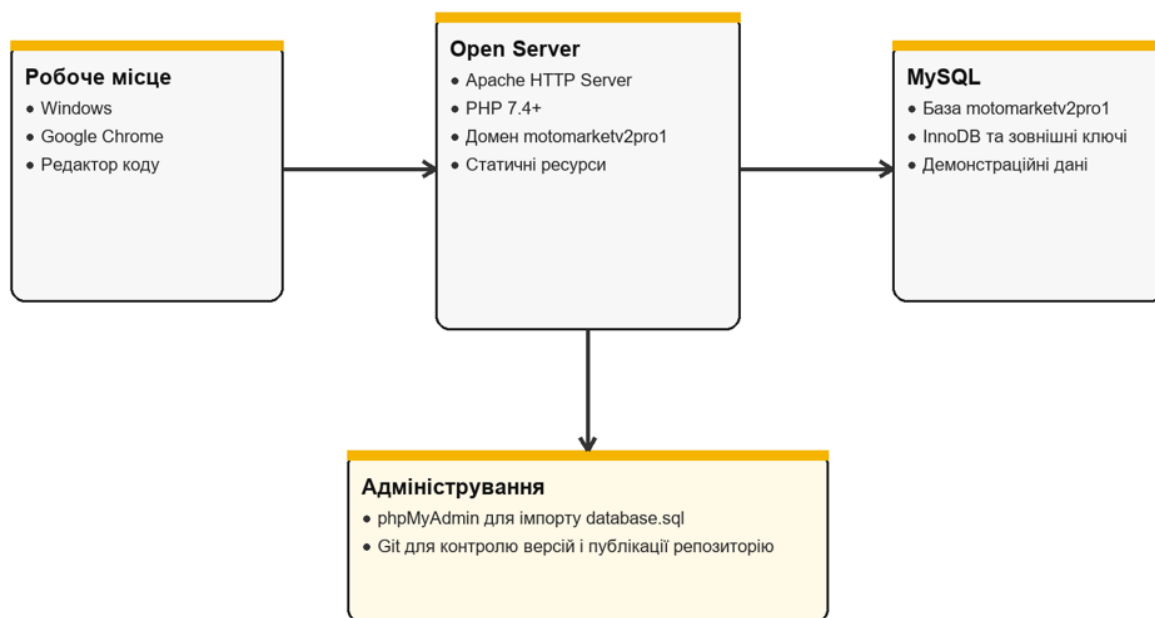


Рис. 3.1 – Схема локального розгортання MotoMarket PRO

Для забезпечення стабільної роботи вебзастосунку використовується класичний стек вебтехнологій. Основні компоненти програмного середовища, необхідні для розгортання та роботи MotoMarket PRO, наведено у (табл. 3.1).

Таблиця 3.1

Компоненти середовища MotoMarket PRO

Компонент	Технологія	Призначення
Вебсервер	Apache HTTP Server	Обробка HTTP-запитів, передача керування PHP-файлам
Серверна мова	PHP	Бізнес-логіка, сесії, формування HTML-відповідей
СУБД	MySQL / InnoDB	Зберігання всіх даних системи, транзакції, зовнішні ключі
Локальний пакет	Open Server	Об'єднує Apache, PHP і MySQL в єдиному середовищі
Управління БД	phpMyAdmin	Імпорт SQL-схеми, перегляд і редагування даних [26]

Забезпечення стабільного зв'язку між серверною логікою вебзастосунку та системою управління базами даних вимагає точного налаштування параметрів доступу. Для зручності адміністрування та можливості швидкого перенесення проєкту між різними середовищами (наприклад, з локального сервера на робочий хостинг), усі ключові налаштування винесено в централізований конфігураційний файл. Основні параметри конфігурації, необхідні для підключення вебзастосунку до бази даних, налаштування кодування та коректної маршрутизації системи, представлено в (табл. 3.2).

Таблиця 3.2

Параметри конфігурації підключення

Параметр	Типове значення	Призначення
host	localhost	Адреса MySQL-сервера
dbname	motomarketv2pro1	Назва бази даних проєкту
username	root	Локальний користувач бази даних
password	root / порожній рядок	Сумісність із різними профілями Open Server
charset	utf8mb4	Коректне зберігання українського тексту та Unicode [11]
BASE_URL	/	Префікс URL для формування посилань у шаблонах

Перенесення вебзастосунку на нове робоче місце або сервер вимагає точного відтворення його програмної інфраструктури. Для коректної роботи системи недостатньо простого копіювання вихідного коду — необхідно виконати комплексну базову ініціалізацію, яка включає налаштування віртуального домену вебсервера, створення відповідної бази даних та імпорт початкового набору даних. Дотримання правильної послідовності цих кроків гарантує відсутність конфліктів та забезпечує повну працездатність усього функціоналу. Детальна послідовність основних дій, необхідних для розгортання, налаштування та запуску проєкту в локальному середовищі, подано в (табл. 3.3).

Таблиця 3.3

Порядок розгортання системи

Крок	Дія	Очікуваний результат
1	Скопіювати папку проєкту до каталогу доменів Open Server	Apache бачить проєкт за локальним доменом
2	Відкрити phpMyAdmin і створити базу motomarketv2pro1	Порожня база з кодуванням utf8mb4 готова до імпорту
3	Імпортувати файл database.sql	Створено 8 таблиць, демо-товари, облікові записи та тестові дані
4	Відкрити локальний домен у браузері	Відображається головна сторінка з каталогом
5	Увійти під обліковим записом user@moto.test	Доступні кошик, обране, профіль і замовлення
6	Увійти під обліковим записом admin@moto.test	Відкривається адміністративна панель admin/index.php

Після імпорту файлу database.sql система автоматично отримує повністю підготовлений набір початкових даних, необхідних для демонстрації та тестування всіх основних функціональних можливостей вебзастосунку. Для спрощення перевірки роботи системи передбачено два демонстраційні облікові записи користувачів. Перший обліковий запис має адресу електронної пошти admin@moto.test та роль адміністратора, що надає доступ до адміністративної панелі, управління товарами, перегляду замовлень і обробки контактних звернень. Другий обліковий запис user@moto.test призначений для тестування функціоналу звичайного покупця та дозволяє перевірити процес перегляду каталогу, додавання товарів до кошика, створення списку бажань, оформлення замовлень і залишення відгуків. З метою забезпечення безпеки облікових записів паролі користувачів не зберігаються у відкритому вигляді. У файлі SQL вони представлені виключно у вигляді bcrypt-хешів, сформованих за допомогою функції password_hash(), яка є стандартним засобом безпечного хешування паролів у PHP [5].

Файл database.sql розроблено таким чином, щоб його можна було багаторазово імпортувати без виникнення помилок або конфліктів структури бази даних. Перед створенням нових таблиць скрипт автоматично видаляє існуючі залежні таблиці у правильній послідовності від дочірніх до батьківських. Такий порядок дозволяє уникнути порушення обмежень зовнішніх ключів та забезпечує коректне очищення бази даних перед повторним створенням структури [9]. Після видалення попередніх таблиць виконується повне створення всієї схеми бази даних разом із необхідними індексами, зв'язками та початковими записами. Це значно спрощує процес розробки, тестування та демонстрації проєкту, оскільки середовище можна швидко повернути до початкового стану в будь-який момент.

Після успішного виконання всіх етапів встановлення система повністю готова до експлуатації. Для перевірки коректності налаштування бази даних можна скористатися інструментом phpMyAdmin та переконатися у наявності всіх восьми таблиць, а також демонстраційних записів у них. Наявність таблиць і тестових даних свідчить про правильне виконання SQL-скрипта та успішне підключення застосунку до СУБД [26].

Додатково рекомендується перевірити працездатність основних бізнес-процесів системи. Для цього необхідно увійти під обліковим записом покупця, додати один або декілька товарів до кошика та пройти повний процес оформлення замовлення. У разі коректної роботи транзакційного механізму після підтвердження замовлення кошик користувача повинен автоматично очиститися, кількість товарів на складі має відповідно зменшитися, а інформація про нове замовлення повинна з'явитися в адміністративній панелі. Успішне виконання цих дій підтверджує правильність роботи бази даних, механізму транзакцій та ключових модулів вебзастосунку MotoMarket PRO.

3.2. Структура інтерфейсу та порядок взаємодії з платформою

Головна сторінка виконує презентаційну й навігаційну функції. У верхній частині розміщено контакти, основне меню, пошук, лічильники

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

обраного й кошика та кнопку кабінету. Головну сторінку вебплатформи MotoMarket PRO зображено на рисунку (рис. 3.2).

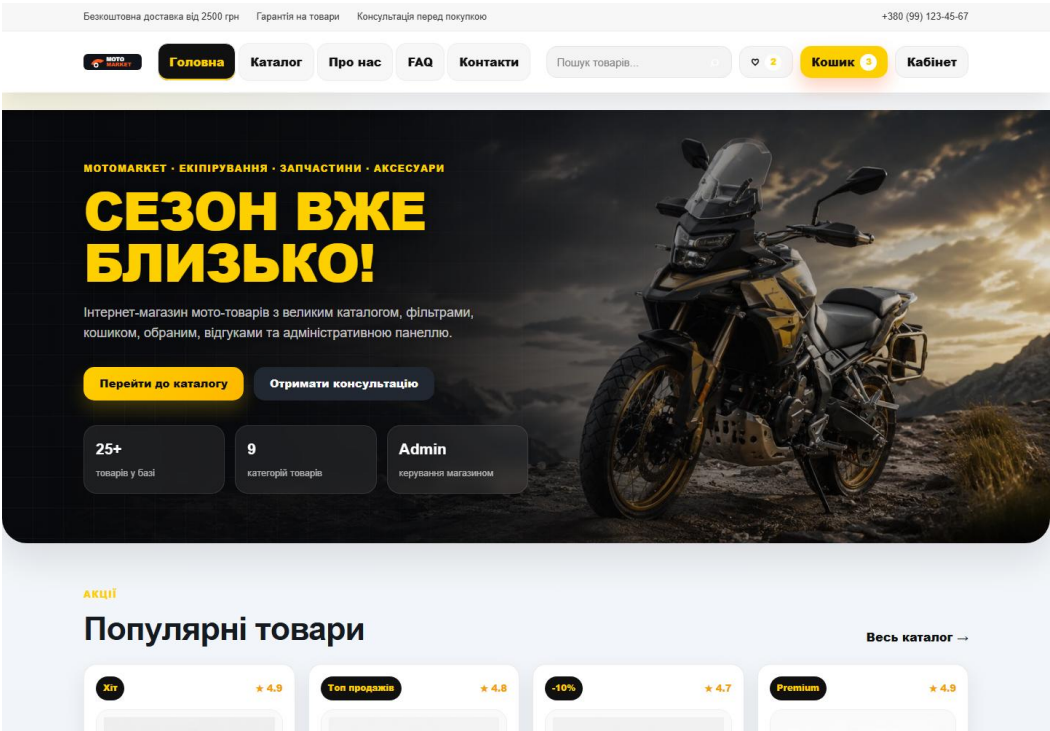


Рис. 3.2 – Головна сторінка MotoMarket PRO

Структура головної сторінки спроектована з урахуванням сучасних вимог до зручності користувацького інтерфейсу. Для швидкого залучення уваги клієнта одразу під навігаційним блоком розміщено секції, що демонструють ключові переваги магазину, популярні категорії, а також спеціальні добірки акційних та нових товарів. Базовим елементом представлення асортименту є стандартизована картка товару, яка містить усю необхідну для прийняття рішення інформацію: якісне зображення, назву, категорію, бренд, візуалізований рейтинг, а також стару та актуальну ціну. Важливим архітектурним рішенням є використання єдиного стилю (компонентного підходу) для відображення карток товарів у загальному каталозі, списку бажань та блоках рекомендацій, що забезпечує візуальну цілісність усього застосунку.

Розділ каталогу починається з інформаційного банера, під яким інтегровано комплексну панель фільтрації. Зовнішній вигляд сторінки

каталогу товарів із реалізованими інструментами пошуку та багатокритеріальної фільтрації наведено на рисунку (рис. 3.3).

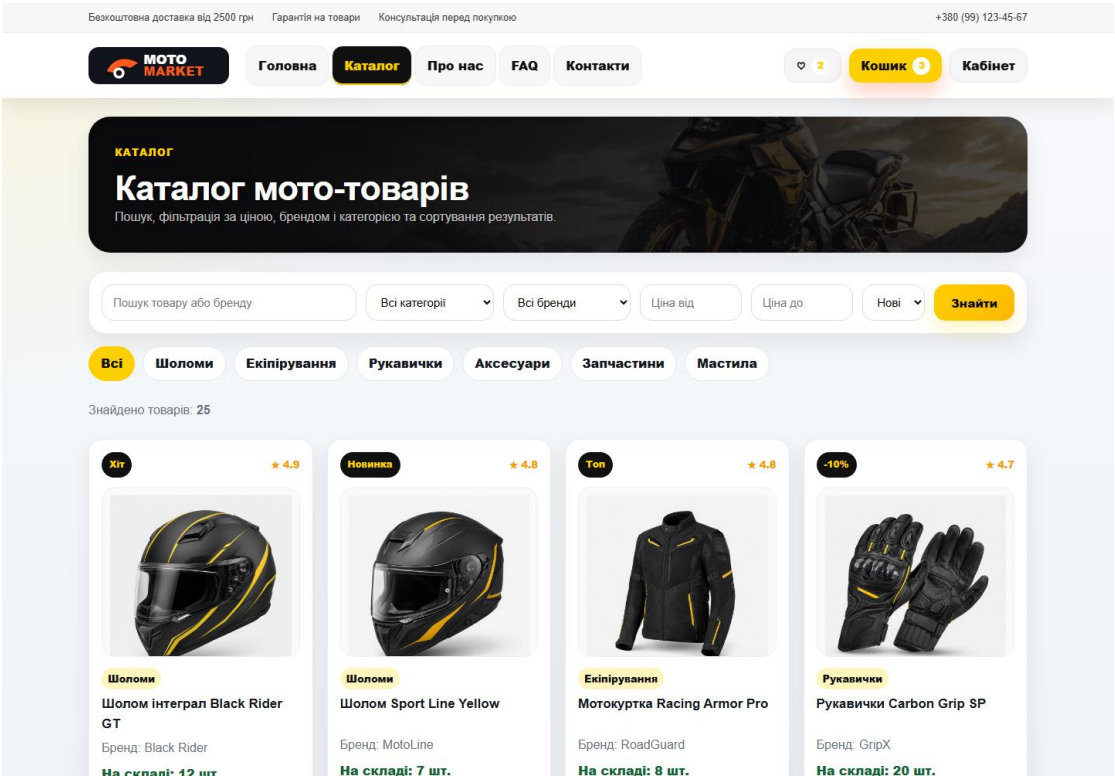


Рис. 3.3 – Каталог, пошук і фільтрація товарів

Сторінка детального перегляду товару побудована таким чином, щоб максимально зосередити увагу користувача на інформації про конкретну позицію. Візуальна частина із фотографією товару відокремлена від інформаційного блоку, що покращує сприйняття контенту та підвищує зручність користування сторінкою. Покупець має можливість переглянути категорію товару, його назву, бренд, рейтинг, спеціальні позначки, актуальну ціну, детальний опис та інформацію про наявний залишок на складі. Для оформлення покупки реалізовано спеціальну форму вибору кількості товару, де максимальне значення quantity автоматично обмежується доступним залишком stock. Це дозволяє уникнути оформлення замовлень на відсутній товар. Окремо реалізована функція додавання товару до списку бажань, яка працює незалежно від механізму кошика та дає можливість зберігати цікаві

товари для майбутніх покупок. Сторінку детального перегляду товару та основні елементи її інтерфейсу показано на рисунку (рис. 3.4).

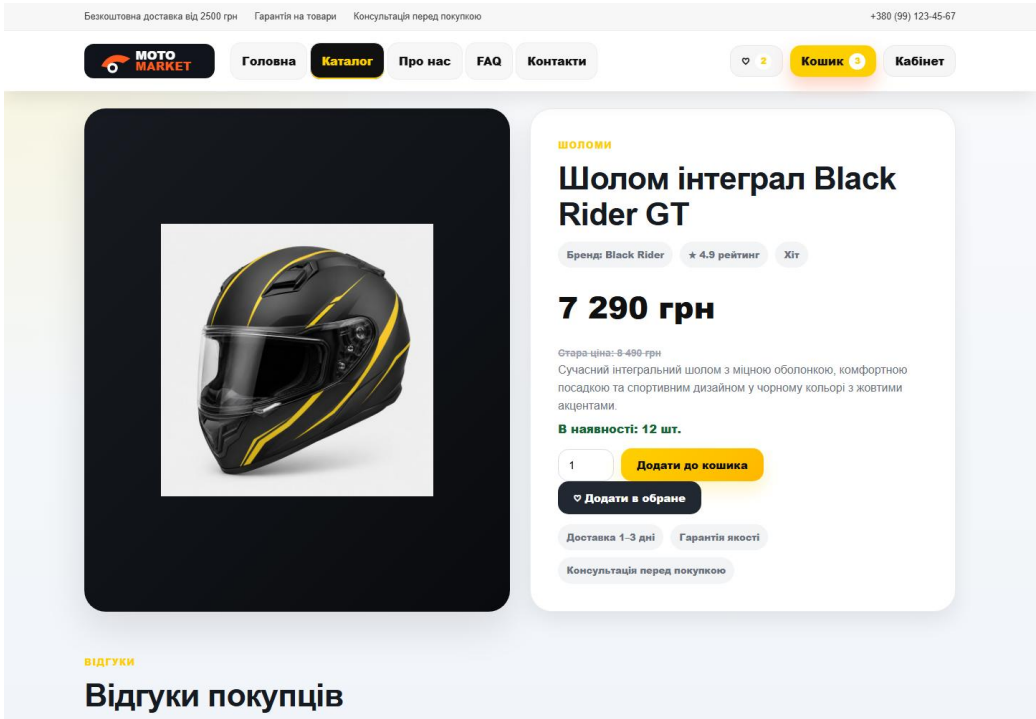


Рис. 3.4 – Сторінка детальної інформації про товар

У нижній частині картки товару відображаються відгуки покупців. Авторизований користувач може поставити оцінку від 1 до 5 та залишити коментар. Сервер перевіряє, чи рейтинг входить у допустимий діапазон і чи текст не порожній, після чого додає запис із поточним user_id. Це дозволяє формувати реальні відгуки, що впливають на довіру до товару.

Кошик реалізовано у вигляді таблиці, де для кожного товару показано назву, ціну, кількість, суму та дію «видалити». Зміна кількості виконується через окрему форму прямо в рядку, що робить процес швидким і зрозумілим. Загальна вартість автоматично обчислюється сервером на основі актуальної ціни та кількості, тому користувач завжди бачить точну суму без необхідності оновлювати сторінку вручну. Кошик зареєстрованого покупця представлено на рисунку (рис. 3.5).

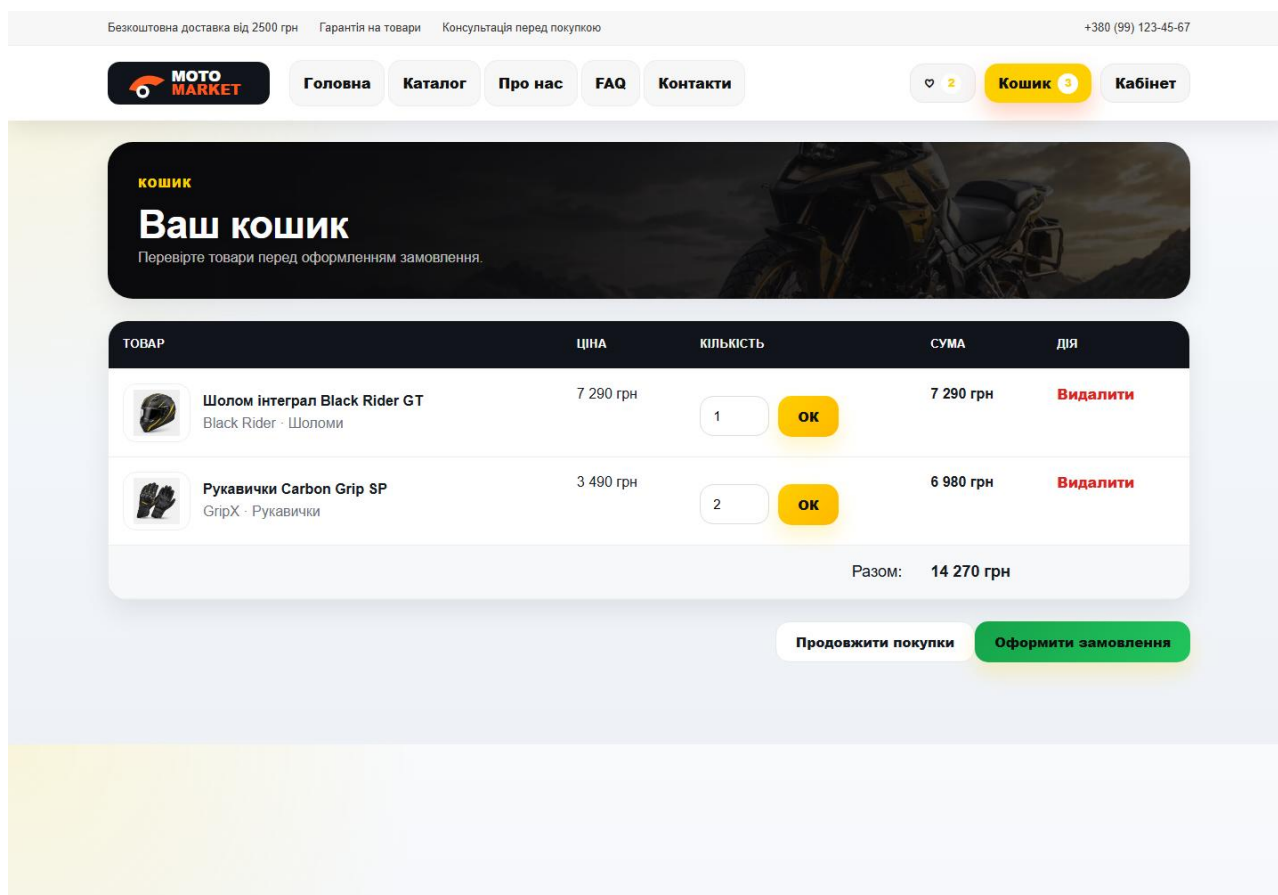


Рис. 3.5 – Кошик зареєстрованого покупця

Для доступу до персоналізованих можливостей платформи користувач спершу проходить реєстрацію або автентифікацію відповідно до сценарію UC-Auth. Користувач ініціює перехід до стану Authenticated виконанням однієї з двох дій: реєстрацією нового облікового запису або введенням облікових даних існуючого. Кожна дія обробляється окремою формою на спільній сторінці. Введені дані перевіряються як на клієнтській стороні (щоб одразу підсвітити помилки й підказати користувачу), так і на сервері (для гарантії безпеки та коректності): сервер перевіряє унікальність email під час реєстрації та відповідність пароля хешу під час входу, після чого створює сесію користувача й присвоює відповідну роль. Такий підхід забезпечує швидкий і зрозумілий процес авторизації, а також захищає систему від некоректних або шкідливих запитів. Сторінку реєстрації та автентифікації користувачів зображено на рисунку (рис. 3.6).

ВХІД

Увійти до кабінету

Отримайте доступ до замовлень, кошика та персональних даних для покупок.

Email

Пароль

Увійти

Після входу ви одразу потрапите до особистого кабінету або в адмін-панель.

РЕЄСТРАЦІЯ

Створити новий акаунт

Зареєструйтесь, щоб зберігати обране, швидше оформлювати замовлення та мати власний профіль.

Ім'я

Email

Пароль

Створити акаунт

Реєстрація займає кілька секунд, після чого акаунт одразу буде активний.

Рис. 3.6 – Сторінка реєстрації та автентифікації користувачів

В особистому кабінеті покупець переглядає облікові дані та змінює ім'я, номер телефону й адресу доставки. Праворуч розташовано швидкі переходи до історії замовлень, списку бажань і каталогу. Особистий кабінет користувача наведено на рисунку (рис. 3.7).

КАБІNET

Особистий кабінет

Новий функціонал: користувач може зберігати контактні дані для замовлень.

МІЙ АКАУНТ

Користувач
user@moto.test

0 замовлень

1 товарів в обраному

0 позицій у кошику

НАЛАШТУВАННЯ

Мої дані

Ім'я

РОЗДІЛИ

Швидкий доступ

Мої замовлення

Обране

Рис. 3.7 – Особистий кабінет користувача

Список бажань дає змогу користувачеві зберігати цікаві товари без необхідності одразу оформлювати покупку. Це зручно, коли він хоче

повернутися до вибору пізніше або порівняти кілька варіантів. Кожну позицію можна швидко додати до кошика чи вилучити зі списку, не залишаючи поточної сторінки. Такий механізм робить процес взаємодії легким і безперервним: користувач не відволікається на додаткові переходи, а платформа зберігає його інтереси й допомагає планувати майбутні покупки.

Таким чином, список бажань виступає як персональний перелік відкладених товарів, які користувач може швидко повернути до кошика або переглянути пізніше. Список бажаних товарів користувача показано на рисунку (рис. 3.8).

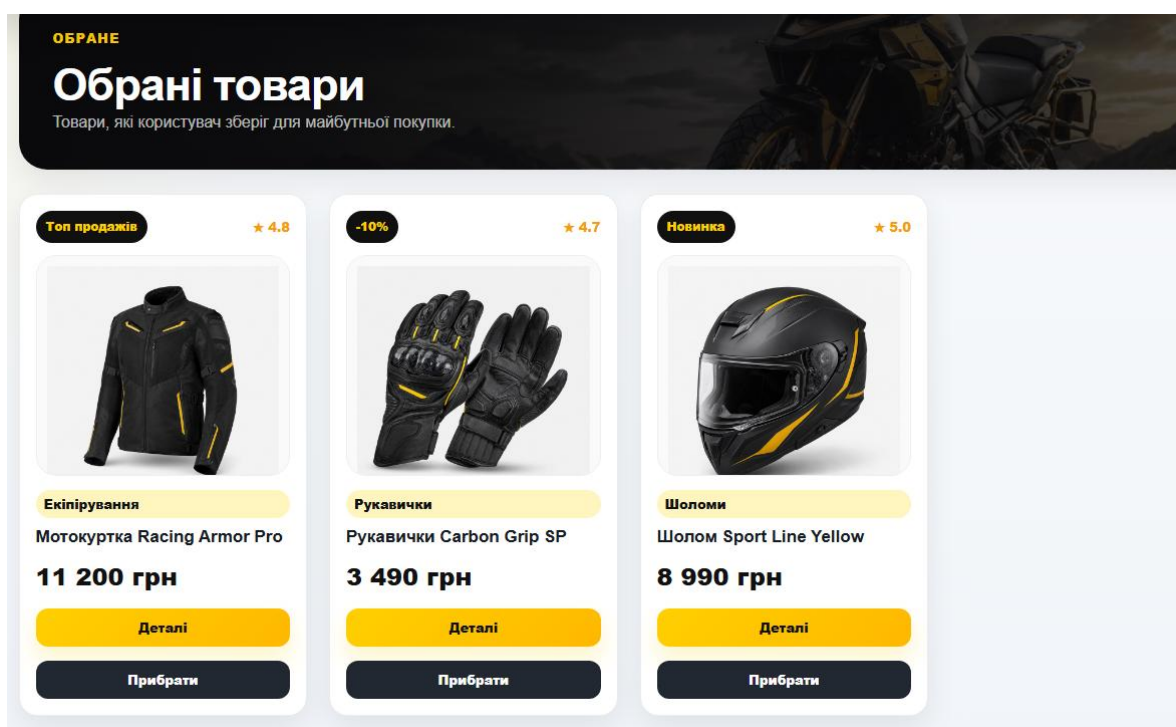


Рис. 3.8 – Список бажаних товарів

Під час оформлення замовлення покупець має можливість уважно перевірити свої контактні дані, щоб уникнути помилок у доставці. Далі він обирає зручний спосіб доставки та форму оплати: це може бути кур'єр, поштове відділення чи самовивіз, а також картка, готівка чи онлайн-сервіс. У процесі він контролює склад замовлення: бачить перелік товарів, їх кількість та ціну. Підсумкова сума автоматично обчислюється системою ще до

підтвердження операції, що дозволяє покупцеві оцінити витрати та прийняти остаточне рішення. Сторінку оформлення замовлення представлено на рисунку (рис. 3.9).

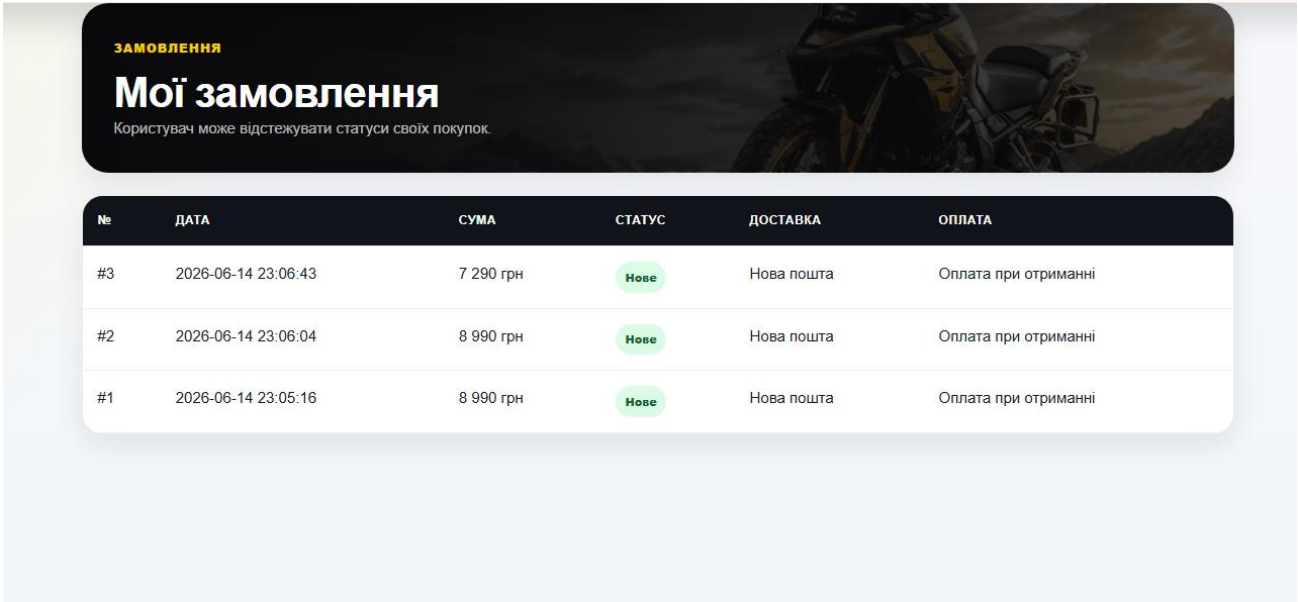
Рис. 3.9 – Сторінка оформлення замовлення

Після підтвердження покупки система автоматично створює новий запис у персональному розділі «Мої замовлення». Це дозволяє користувачеві в будь-який момент повернутися до перегляду історії своїх попередніх покупок, яка надійно зберігається в базі даних без обмежень за часом. Для кожного окремого замовлення в загальному списку відображаються ключові дані: унікальний номер (ідентифікатор), точна дата та час оформлення, підсумкова сума до сплати, актуальний статус виконання, вибраний спосіб доставки, а також метод оплати.

Завдяки такому структурованому підходу покупець має повний контроль над усім логістичним та фінансовим процесом. Він може самостійно й у реальному часі відстежувати, на якому саме етапі перебуває його замовлення від обробки менеджером до відправки та прибуття у пункт

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

призначення. Це мінімізує потребу у зверненнях до служби підтримки, підвищує рівень довіри, а також забезпечує абсолютну прозорість та надійність сервісу. Поточний інтерфейс історії замовлень користувача зображено на рисунку (рис. 3.10).



№	ДАТА	СУМА	СТАТУС	ДОСТАВКА	ОПЛАТА
#3	2026-06-14 23:06:43	7 290 грн	Нове	Нова пошта	Оплата при отриманні
#2	2026-06-14 23:06:04	8 990 грн	Нове	Нова пошта	Оплата при отриманні
#1	2026-06-14 23:05:16	8 990 грн	Нове	Нова пошта	Оплата при отриманні

Рис. 3.10 – Замовлення користувача

Адміністративна панель зберігає той самий візуальний стиль, що й користувацька частина платформи, але має окреме меню з інструментами для керування. На першому екрані адміністратор бачить ключові показники: кількість товарів у системі, загальну кількість замовлень, зареєстрованих користувачів та отриманих повідомлень. Додатково відображається список останніх замовлень, що дозволяє швидко оцінити поточну активність.

Кнопки керування відкривають відповідні таблиці наприклад, перелік товарів із можливістю редагування, список замовлень із деталями та статусами, базу користувачів із контактними даними, а також журнал повідомлень. Це забезпечує адміністратору централізований доступ до всіх важливих даних і дозволяє оперативно реагувати на зміни. Головну сторінку адміністративної панелі MotoMarket PRO наведено на рисунку (рис. 3.11).

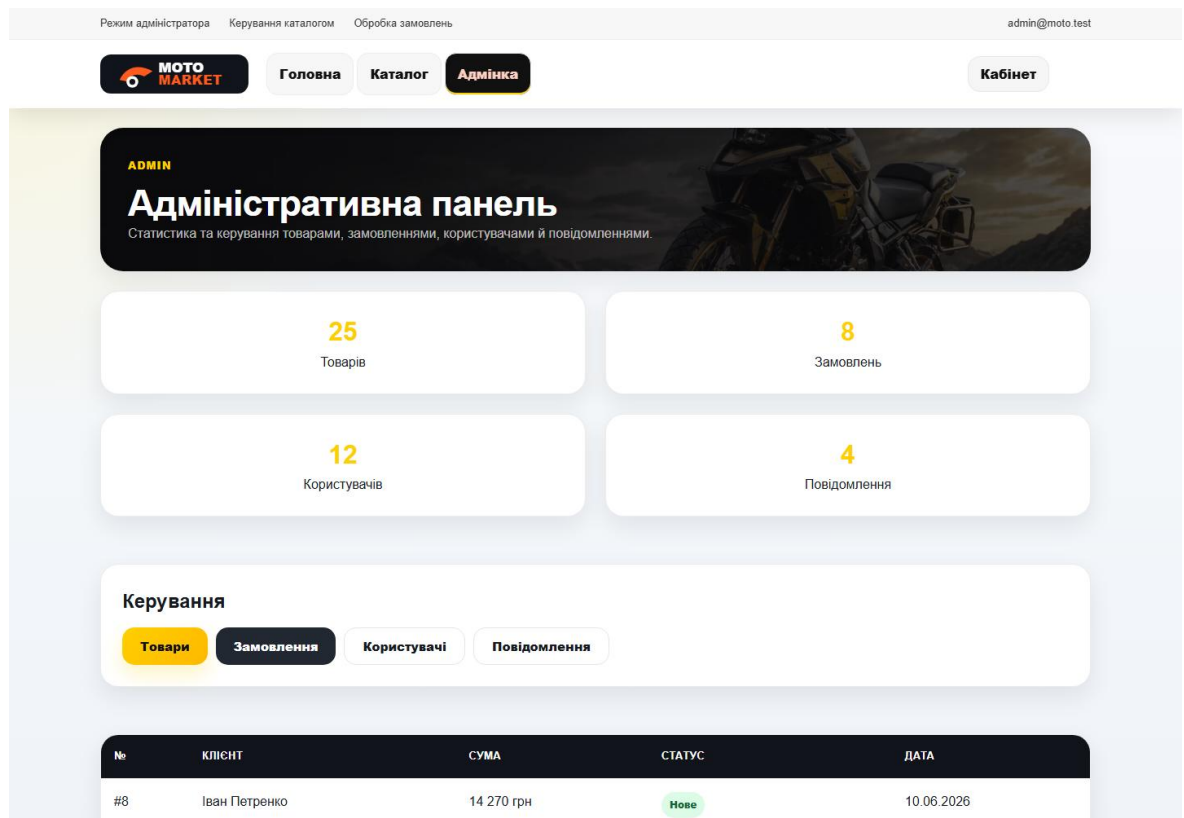


Рис. 3.11 – Головна сторінка адміністративної панелі

Сторінка товарів дозволяє додавати позицію або перейти до редагування. Форма містить назву, опис, ціни, категорію, бренд, залишок, ім'я файлу зображення, badge і рейтинг. Видалення супроводжується клієнтським підтвердженням і повторною серверною перевіркою ролі.

Для замовлення адміністратор переглядає контактні дані, доставку, оплату, коментар і всі позиції. Статус оновлюється окремою POST-формою. Список користувачів є інформаційним, а контактні повідомлення відображаються в хронологічному порядку.

3.3. Тестування роботи програмного продукту

Перевірку сформовано за сценарним підходом. Для кожної функції визначено передумову, послідовність дій та результат, який безпосередньо впливає з реалізованої серверної логіки. Окремо перевіряються негативні випадки: доступ без сесії, перевищення складського залишку, порожні поля та звернення до адміністративних сторінок користувачем із роллю user. Такий

підхід дозволяє перевірити не лише позитивні сценарії, а й коректність захисних механізмів системи [17, 21].

Перевірка автентифікації охоплює чотири сценарії. Успішна реєстрація з валідними даними створює запис у таблиці users із хешованим паролем і одразу відкриває сесію. Спроба зареєструватися з уже існуючим email повертає повідомлення про помилку без створення дублікату завдяки обмеженню UNIQUE на рівні бази даних [8]. Успішний вхід встановлює сесію та виконує перенаправлення залежно від ролі. Введення неправильного пароля відхиляє вхід без уточнення причини, щоб не розкривати інформацію про існування облікового запису [17].

Перевірка каталогу та пошуку підтверджує коректну роботу фільтрів. При виборі категорії система повертає виключно товари цієї категорії. При одночасному застосуванні кількох фільтрів категорії, бренду та діапазону ціни результат містить лише позиції, що задовольняють усі умови одночасно. Текстовий пошук знаходить товари за збігом у назві, описі та бренді. Порожній результат при некоректному запиті відображає відповідне повідомлення без помилки сервера.

Перевірка кошика охоплює додавання, оновлення та граничні випадки. При першому додаванні товару створюється новий запис у cart. При повторному додаванні того самого товару кількість збільшується, але не перевищує значення stock [8]. Спроба вручну встановити кількість більшу за залишок через форму оновлення обмежується до максимально допустимого значення. Видалення позиції з кошика одразу відображається в лічильнику навігації без перезавантаження сторінки.

Перевірка оформлення замовлення є найбільш розгорнутою, оскільки цей сценарій зачіпає одразу чотири таблиці. Після заповнення форми та підтвердження система створює запис в orders, додає позиції до order_items із фіксованою поточною ціною, зменшує stock кожного товару та очищає кошик усе в межах однієї транзакції [4, 10], наведено у таблиці (табл. 3.4).

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

Таблиця 3.4

Функціональні сценарії перевірки системи

Сценарій	Дії	Очікуваний результат	Стан
Реєстрація	Ввести нові валідні дані	Створено запис users, відкрито сесію	Реалізовано
Повторний email	Зареєструвати існуючий email	Показано повідомлення про помилку	Реалізовано
Вхід покупця	Ввести коректний email і пароль	Встановлено сесію, redirect на головну	Реалізовано
Вхід адміністратора	Ввести дані admin@moto.test	Redirect на admin/index.php	Реалізовано
Неправильний пароль	Ввести неправильний пароль	Вхід відхилено без уточнення причини	Реалізовано
Фільтр каталогу	Обрати категорію і діапазон ціни	Показано лише відповідні товари	Реалізовано
Текстовий пошук	Ввести назву або бренд	Знайдено товари за збігом у трьох полях	Реалізовано
Додавання до кошика	Додати товар двічі	Кількість збільшено в межах stock	Реалізовано
Перевищення залишку	Встановити quantity > stock	Кількість обмежено до доступного залишку	Реалізовано
Список бажань	Двічі натиснути toggle	Запис створено, потім видалено	Реалізовано
Checkout	Заповнити форму з непорожнім кошиком	Створено замовлення, кошик очищено, stock зменшено	Реалізовано
Порожній checkout	Відкрити checkout без товарів	Перенаправлення на cart.php	Реалізовано
Відгук	Надіслати рейтинг і коментар	Відгук відображено на сторінці товару	Реалізовано
Некоректний рейтинг	Надіслати рейтинг поза межами 1–5	Відгук відхилено з повідомленням	Реалізовано
Admin-доступ як user	Відкрити admin/ із роллю user	Перенаправлення на login.php	Реалізовано
Admin-доступ без сесії	Відкрити admin/ без авторизації	Перенаправлення на login.php	Реалізовано
Контактна форма	Надіслати звернення без авторизації	Запис збережено в таблиці contacts	Реалізовано

Статичний аналіз коду показує послідовне використання підготовлених запитів PDO для всіх значень, що надходять від користувача [2, 3]. Сортування каталогу реалізоване через whitelist допустимих варіантів, а не пряме підставлення GET-параметра до SQL [21]. Весь динамічний HTML проходить через функцію safe(), яка застосовує htmlspecialchars() перед виведенням [7, 20]. Серверна перевірка ролі виконується на початку кожного файлу адміністративного каталогу незалежно від клієнтської навігації [17].

Окремо проведено аналіз безпекових ризиків системи. Для кожної виявленої загрози визначено наявний захисний механізм та рекомендації щодо посилення перед промисловим розгортанням, наведено у таблиці (табл. 3.5).

Таблиця 3.5

Аналіз безпекових ризиків системи

Ризик	Наявний захист	Рекомендоване посилення
SQL-ін'єкція	PDO prepared statements для всіх user-input значень	Зберегти whitelist для всіх динамічних частин запиту
XSS	safe() / htmlspecialchars() на всьому динамічному виводі	Впровадити Content Security Policy через HTTP-заголовки
Злом пароля	password_hash() / password_verify() з bcrypt	Додати політику складності та rate limiting на вхід
Несанкціонований admin	is_admin() на початку кожної admin-сторінки	Додати журнал адміністративних операцій із timestamp
CSRF	POST-запити для більшості змін даних	Додати CSRF-токени до всіх форм зміни стану [19]
Витік конфігурації	Параметри у локальному config.php	Винести секрети до .env-файлу поза репозиторієм
Негативний залишок	Перевірка quantity відносно stock у PHP	Атомарний UPDATE з умовою WHERE stock >= quantity
Перебір облікових записів	Єдине повідомлення про помилку входу	Додати тимчасове блокування після N невдалих спроб

Основним напрямом подальшого посилення захисту є впровадження CSRF-токенів для операцій зміни даних, зокрема роботи з кошиком, списком бажань, відгуками та замовленнями.

Перевірка адаптивності підтвердила коректну роботу інтерфейсу на різних розмірах екранів. У подальшому доцільно виконати аудит інтерфейсу на відповідність стандарту WCAG 2.2 [14, 23].

Висновки до третього розділу

Було розглянуто процес встановлення, налаштування та запуску вебзастосунку MotoMarket PRO у локальному середовищі. Описано архітектуру клієнт-серверної взаємодії, основні програмні компоненти системи (Apache, PHP, MySQL) та порядок їх використання в рамках Open Server. Також визначено параметри конфігурації підключення до бази даних і наведено послідовність кроків розгортання системи, що забезпечує її коректний запуск і готовність до тестування.

Проаналізовано структуру користувацького та адміністративного інтерфейсів вебзастосунку MotoMarket PRO. Розглянуто основні функціональні екрани системи, зокрема головну сторінку, каталог товарів із пошуком і фільтрацією, сторінку товару, кошик, список бажань, особистий кабінет та процес оформлення замовлення. Окрему увагу приділено адміністративній панелі, яка забезпечує керування товарами, замовленнями, користувачами та зверненнями, формуючи повноцінний інструмент управління системою.

Виконано тестування функціональних можливостей вебзастосунку за сценарним підходом, включаючи як позитивні, так і негативні сценарії. Перевірено роботу основних модулів системи: реєстрації, авторизації, каталогу, кошика, оформлення замовлень, відгуків та адміністративного доступу. Додатково проведено аналіз безпекових ризиків і перевірено ефективність реалізованих механізмів захисту, що підтвердило коректність роботи системи та її готовність до подальшого використання.

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

ВИСНОВКИ

У кваліфікаційній роботі розроблено вебзастосунок MotoMarket PRO для спеціалізованого продажу мотоекіпірування, запчастин, мастил, електроніки та аксесуарів.

Під час аналізу визначено потребу у структурованому каталозі, фільтрації, персональному кошику, обліку замовлень і окремому адміністративному контурі. Порівняння з маркетплейсами, тематичними магазинами й дошками оголошень дозволило обмежити систему функціями одного продавця.

Спроектовано трирівневу архітектуру з серверним формуванням сторінок. PHP реалізує сценарії та сесії, PDO забезпечує доступ до MySQL, а HTML, CSS і JavaScript формують адаптивний клієнтський інтерфейс. Структура проєкту не потребує окремої системи збірки.

Реляційна база даних складається з таблиць users, products, cart, wishlist, orders, order_items, reviews і contacts. Зовнішні ключі відображають предметні зв'язки, а транзакція оформлення забезпечує узгоджене створення замовлення, позицій, зміну залишку й очищення кошика.

Реалізовано реєстрацію та вхід, профіль, каталог із комбінованими фільтрами, деталі товару, відгуки, список бажань, кошик, checkout, історію замовлень і контактну форму. Адміністративна панель підтримує керування товарами, перегляд і зміну статусу замовлень, користувачів і звернень.

Для базового захисту використано хешування паролів, серверні перевірки ролей, параметризовані SQL-запити та екранування HTML. Визначено напрями подальшого розвитку: CSRF-захист, env-конфігурація, автоматизоване тестування, інтеграція платіжного шлюзу, поштового API та електронних повідомлень.

					ПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PHP Group. PHP Manual [Електронний ресурс]. — 2026. — Режим доступу: <https://www.php.net/manual/en/> (дата звернення: 18.06.2026).
2. PHP Group. PDO extension [Електронний ресурс]. — 2026. — Режим доступу: <https://www.php.net/manual/en/book.pdo.php> (дата звернення: 18.06.2026).
3. PHP Group. PDO::prepare [Електронний ресурс]. — 2026. — Режим доступу: <https://www.php.net/manual/en/pdo.prepare.php> (дата звернення: 18.06.2026).
4. PHP Group. PDO transactions [Електронний ресурс]. — 2026. — Режим доступу: <https://www.php.net/manual/en/pdo.transactions.php> (дата звернення: 18.06.2026).
5. PHP Group. password_hash [Електронний ресурс]. — 2026. — Режим доступу: <https://www.php.net/manual/en/function.password-hash.php> (дата звернення: 18.06.2026).
6. PHP Group. Session handling [Електронний ресурс]. — 2026. — Режим доступу: <https://www.php.net/manual/en/book.session.php> (дата звернення: 18.06.2026).
7. PHP Group. htmlspecialchars [Електронний ресурс]. — 2026. — Режим доступу: <https://www.php.net/manual/en/function htmlspecialchars.php> (дата звернення: 18.06.2026).
8. Oracle. MySQL 8.4 Reference Manual [Електронний ресурс]. — 2026. — Режим доступу: <https://dev.mysql.com/doc/refman/8.4/en/> (дата звернення: 18.06.2026).
9. Oracle. InnoDB foreign keys [Електронний ресурс]. — 2026. — Режим доступу: <https://dev.mysql.com/doc/refman/8.4/en/create-table-foreign-keys.html> (дата звернення: 18.06.2026).
10. Oracle. COMMIT and ROLLBACK [Електронний ресурс]. — 2026. — Режим доступу: <https://dev.mysql.com/doc/refman/8.4/en/commit.html> (дата звернення: 18.06.2026).

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

- 11.Oracle. utf8mb4 character set [Електронний ресурс]. — 2026. — Режим доступу:<https://dev.mysql.com/doc/refman/8.4/en/charset-unicode-utf8mb4.html> (дата звернення: 18.06.2026).
- 12.Mozilla Developer Network. HTML: HyperText Markup Language [Електронний ресурс]. — 2026. — Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 18.06.2026).
- 13.Mozilla Developer Network. CSS: Cascading Style Sheets [Електронний ресурс]. — 2026. — Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 18.06.2026).
- 14.Mozilla Developer Network. Responsive web design [Електронний ресурс]. — 2026. — Режим доступу: https://developer.mozilla.org/en-US/docs/Learn_web_development (дата звернення: 18.06.2026).
- 15.Mozilla Developer Network. CSS grid layout [Електронний ресурс]. — 2026. — Режим доступу: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_grid_layout (дата звернення: 18.06.2026).
- 16.Mozilla Developer Network. Using Fetch [Електронний ресурс]. — 2026. — Режим доступу: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch (дата звернення: 18.06.2026).
- 17.OWASP Foundation. Authentication Cheat Sheet [Електронний ресурс]. — 2026. — Режим доступу: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html (дата звернення: 18.06.2026).
- 18.OWASP Foundation. Session Management Cheat Sheet [Електронний ресурс]. — 2026. — Режим доступу: https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html (дата звернення: 18.06.2026).
- 19.OWASP Foundation. CSRF Prevention Cheat Sheet [Електронний ресурс]. — 2026. — Режим доступу: <https://cheatsheetseries.owasp.org/cheatsheets/Cross->

[Site Request Forgery Prevention Cheat Sheet.html](#) (дата звернення: 18.06.2026).

- 20.OWASP Foundation. XSS Prevention Cheat Sheet [Електронний ресурс]. — 2026. — Режим доступу: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html (дата звернення: 18.06.2026).
- 21.OWASP Foundation. SQL Injection Prevention Cheat Sheet [Електронний ресурс]. — 2026. — Режим доступу: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html (дата звернення: 18.06.2026).
- 22.Fielding R., Nottingham M., Reschke J. HTTP Semantics [Електронний ресурс]. — 2022. — Режим доступу: <https://www.rfc-editor.org/rfc/rfc9110> (дата звернення: 18.06.2026).
- 23.W3C Web Accessibility Initiative. WCAG 2.2 [Електронний ресурс]. — 2023. — Режим доступу: <https://www.w3.org/TR/WCAG22/> (дата звернення: 18.06.2026).
- 24.Chacon S., Straub B. Git documentation [Електронний ресурс]. — 2026. — Режим доступу: <https://git-scm.com/doc> (дата звернення: 18.06.2026).
- 25.Apache Friends. XAMPP [Електронний ресурс]. — 2026. — Режим доступу: <https://www.apachefriends.org/index.html> (дата звернення: 18.06.2026).
- 26.phpMyAdmin Development Team. phpMyAdmin documentation [Електронний ресурс]. — 2026. — Режим доступу: <https://docs.phpmyadmin.net/en/latest/> (дата звернення: 18.06.2026).
- 27.Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. 7th ed. Sebastopol: O'Reilly Media, 2024. 832 p.
- 28.Lockhart J. Modern PHP: The Complete Guide. New York: Apress, 2023. 420 p.
- 29.Duckett J. HTML and CSS: Design and Build Websites. Updated Edition. Indianapolis: Wiley, 2021. 490 p.
- 30.Duckett J. JavaScript and jQuery: Interactive Front-End Web Development. Updated Edition. Indianapolis: Wiley, 2022. 640 p.

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А

У додатку наведено основну частину SQL-сценарію створення таблиць MotoMarket PRO. Повний файл database.sql також містить демонстраційні облікові записи, товари, відгуки та контактне повідомлення.

Обробка А.1 – Файл database.sql

```
CREATE DATABASE IF NOT EXISTS motomarketv2pro1 CHARACTER SET utf8mb4  
COLLATE utf8mb4_unicode_ci;  
USE motomarketv2pro1;
```

```
DROP TABLE IF EXISTS reviews;  
DROP TABLE IF EXISTS wishlist;  
DROP TABLE IF EXISTS order_items;  
DROP TABLE IF EXISTS orders;  
DROP TABLE IF EXISTS cart;  
DROP TABLE IF EXISTS contacts;  
DROP TABLE IF EXISTS products;  
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(120) NOT NULL,  
    email VARCHAR(160) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    role ENUM('user','admin') NOT NULL DEFAULT 'user',  
    phone VARCHAR(40) DEFAULT '',  
    address TEXT,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE products (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(180) NOT NULL,  
    description TEXT NOT NULL,  
    price DECIMAL(10,2) NOT NULL,  
    old_price DECIMAL(10,2) NULL,  
    category VARCHAR(80) NOT NULL,  
    brand VARCHAR(80) NOT NULL,  
    stock INT NOT NULL DEFAULT 0,  
    image VARCHAR(160) DEFAULT '',  
    badge VARCHAR(40) DEFAULT '',  
    rating DECIMAL(2,1) NOT NULL DEFAULT 4.8,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE cart (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT NOT NULL,
```

					ІПЗ.КР.Б – 121 – 26 - ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

```

        product_id INT NOT NULL,
        quantity INT NOT NULL DEFAULT 1,
        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
        FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE
CASCADE
);

CREATE TABLE wishlist (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    product_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE KEY uniq_wishlist (user_id, product_id),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE
CASCADE
);

CREATE TABLE orders (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    customer_name VARCHAR(120) NOT NULL,
    phone VARCHAR(40) NOT NULL,
    address TEXT NOT NULL,
    delivery_method VARCHAR(80) NOT NULL,
    payment_method VARCHAR(80) NOT NULL,
    comment TEXT,
    total DECIMAL(10,2) NOT NULL,
    status VARCHAR(40) NOT NULL DEFAULT 'HoBe',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);

CREATE TABLE order_items (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE
CASCADE
);

CREATE TABLE reviews (
    id INT AUTO_INCREMENT PRIMARY KEY,
    product_id INT NOT NULL,
    user_id INT NOT NULL,
    rating INT NOT NULL,
    comment TEXT NOT NULL,

```

					ІІЗ.КР.Б – 121 – 26 - ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

```

        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE
CASCADE,
        FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
    );

```

```

CREATE TABLE contacts (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(120) NOT NULL,
    email VARCHAR(160) NOT NULL,
    subject VARCHAR(180) DEFAULT '',
    message TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

```

INSERT INTO users (name, email, password, role, phone, address)
VALUES
('Адміністратор', 'admin@moto.test',
'$2y$12$hAqaQ0xmWjP74tVni/XEeuEd24uXTnH8HA9x15WmJbF0sVZZ1bKXe',
'admin', '+380991234567', 'Київ, Україна'),
('Користувач', 'user@moto.test',
'$2y$12$h6s3kBIqMiX0Sgr0TZFzm.QTeuaVVtwNbs73Q6ephF48ni1J5cinu',
'user', '+380671112233', 'Львів, відділення Нової пошти №5');

```

```

INSERT INTO products (name, description, price, old_price, category,
brand, stock, image, badge, rating) VALUES
('Шолом інтеграл Black Rider GT', 'Сучасний інтегральний шолом з
міцною оболонкою, комфортною посадкою та спортивним дизайном у
чорному кольорі з жовтими акцентами.', 7290, 8490, 'Шоломи', 'Black
Rider', 12, 'helmet-black-yellow.jpg', 'Хіт', 4.9),
('Шолом Sport Line Yellow', 'Легкий спортивний шолом для міста і
траси. Має вентиляційні канали, прозорий візор та агресивний
дизайн.', 8990, 9990, 'Шоломи', 'MotoLine', 7, 'helmet-sport-
lines.jpg', 'Новинка', 4.8),
('Мотокуртка Racing Armor Pro', 'Захисна мотокуртка з посиленими
плечима, ліктями та вентиляційними вставками. Підійде для щоденної
їзди.', 11200, 12900, 'Екіпування', 'RoadGuard', 8, 'jacket-black-
yellow.jpg', 'Топ продажів', 4.8),
('Рукавички Carbon Grip SP', 'Спортивні мотоперчатки з карбоновим
захистом кісточок, міцною долонею та зручними застібками.', 3490,
3990, 'Рукавички', 'GripX', 20, 'gloves-racing-yellow.jpg', '-10%',
4.7),
('Рукавички Armor Black Touring', 'Міцні чорні рукавички для
подорожей і міста. Захищають пальці, долоню та зап'ястя.', 2890,
3390, 'Рукавички', 'KomfortRide', 18, 'gloves-armor-black.jpg',
'Акція', 4.6),
('Мотоботи Forma Adventure Pro', 'Високі мотоботи для подорожей з
посиленням носком, п'ятою та захистом гомілки.', 9800, 10900,
'Екіпування', 'Forma', 7, 'boots-touring-yellow.jpg', 'Новинка',
4.7),

```

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

('Моторюкзак Aero Shell 25L', 'Аеродинамічний мото-рюкзак з твердою спинкою, дощовим захистом та відділенням для ноутбука.', 4100, 4890, 'Акcesуари', 'AeroPack', 14, 'backpack-moto-yellow.jpg', 'Хіт', 4.7),
 ('Вихлопна система Carbon Slip-On', 'Легка вихлопна система з металевим корпусом і карбоновим завершенням. Покращує звук і вигляд мотоцикла.', 18900, 20500, 'Акcesуари', 'AkraStyle', 5, 'exhaust-carbon-yellow.jpg', 'Premium', 4.9),
 ('Глушник Metal Performance', 'Металевий глушник для покращення звучання та стилю мотоцикла. Підходить для спортивного тюнінгу.', 14600, 15900, 'Акcesуари', 'MotoTech', 6, 'exhaust-metal-yellow.jpg', '-8%', 4.6),
 ('Комплект зірок та ланцюга Gold DID', 'Надійний комплект ланцюга та зірок для трансмісії. Має міцні золотисті ланки та довгий ресурс.', 5400, 6200, 'Запчастини', 'DID', 15, 'chain-sprocket-gold.jpg', 'Топ', 4.6),
 ('Зірка передня Racing Sprocket', 'Металева передня зірка для стабільної передачі крутного моменту та активної експлуатації.', 1350, NULL, 'Запчастини', 'SteelDrive', 32, 'sprocket-metal.jpg', 'В наявності', 4.5),
 ('Набір моторних деталей Service Kit', 'Комплект деталей для обслуговування мотоцикла. Підійде для планового ремонту та заміни.', 3200, 3700, 'Запчастини', 'ServiceMoto', 13, 'parts-engine-studio.jpg', 'Сервіс', 4.5),
 ('Моторне масло MotoOil 10W-40', 'Якісне синтетичне моторне масло для сучасних мотоциклів. Забезпечує стабільну роботу двигуна.', 780, 920, 'Мастила', 'MotoOil', 40, 'engine-oil-yellow.jpg', 'Хіт', 4.8),
 ('Масло Racing 15W-50', 'Синтетичне масло для спортивних мотоциклів з підвищеним захистом двигуна при високих навантаженнях.', 890, NULL, 'Мастила', 'RacingOil', 28, 'engine-oil-yellow.jpg', 'Новинка', 4.7),
 ('Колесо Adventure Gold Rim', 'Колесо з золотим ободом, спицями та гальмівним диском для adventure-мотоциклів.', 12400, 13900, 'Шини та диски', 'WheelPro', 6, 'wheel-gold-spokes.jpg', '-10%', 4.9),
 ('Шина Dual Sport з диском', 'Міцне колесо для дорожніх і легких позашляхових поїздки. Добре тримає зчеплення.', 11800, 13200, 'Шини та диски', 'TrailMax', 5, 'wheel-gold-rim.jpg', 'Топ', 4.8),
 ('Мотогарнітура Intercom Pro', 'Комплект гарнітури для шолома з мікрофоном і динаміками. Підходить для зв'язку під час поїздки.', 4590, 5290, 'Електроніка', 'RideTalk', 16, 'intercom-headset-yellow.jpg', 'Новинка', 4.8),
 ('Bluetooth гарнітура Rider Connect', 'Компактна гарнітура для спілкування з пасажиром, навігації та прийому дзвінків.', 3890, NULL, 'Електроніка', 'RiderConnect', 12, 'intercom-headset-yellow.jpg', 'Хіт', 4.7),
 ('Кофр Touring Box 47L', 'Місткий задній кофр для подорожей. Допомогає безпечно перевозити речі під час дальніх поїздки.', 8700, 9500, 'Акcesуари', 'TourBox', 9, 'backpack-moto-yellow.jpg', 'Топ продажів', 4.7),
 ('Захист спини Armor Back Pro', 'Легкий захист спини для використання під курткою. Зменшує ризик травм при падінні.', 2650, NULL, 'Захист', 'SafeRide', 18, 'jacket-black-yellow.jpg', 'Топ', 4.6)

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

Фрагмент checkout.php демонструє завантаження кошика, перевірку форми, створення заголовка й позицій замовлення, зміну складських залишків та очищення кошика у межах транзакції.

Обробка Б.1 – Серверний сценарій checkout.php

```
<?php
require_once 'config.php';
if (!is_logged_in()) redirect('login.php');

$user_id = (int)$_SESSION['user_id'];

$stmt = $pdo->prepare('SELECT c.*, p.name, p.price, p.stock FROM cart
c JOIN products p ON c.product_id = p.id WHERE c.user_id = ?');
$stmt->execute([$user_id]);
$items = $stmt->fetchAll();

if (!$items) {
    redirect('cart.php');
}

$user_stmt = $pdo->prepare('SELECT * FROM users WHERE id = ?');
$user_stmt->execute([$user_id]);
$user = $user_stmt->fetch();

$total = 0;
foreach ($items as $i) {
    $total += $i['price'] * $i['quantity'];
}
$error = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $customer_name = trim($_POST['customer_name'] ?? '');
    $phone = trim($_POST['phone'] ?? '');
    $address = trim($_POST['address'] ?? '');
    $delivery = trim($_POST['delivery'] ?? 'Нова пошта');
    $payment = trim($_POST['payment'] ?? 'Оплата при отриманні');
    $comment = trim($_POST['comment'] ?? '');

    if ($customer_name === '' || $phone === '' || $address === '') {
        $error = 'Заповніть ім'я, телефон та адресу доставки.';
    } else {
        $pdo->beginTransaction();
        try {
            $stmt = $pdo->prepare('INSERT INTO orders (user_id,
customer_name, phone, address, delivery_method, payment_method,
comment, total, status) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)');
            $stmt->execute([$user_id, $customer_name, $phone,
$address, $delivery, $payment, $comment, $total, 'Нове']);
        } catch (Exception $e) {
            $pdo->rollBack();
            $error = $e->getMessage();
        }
    }
}
```

						Арк.
					ІІЗ.КР.Б – 121 – 26 - ПЗ	72
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$order_id = (int)$pdo->lastInsertId();

foreach ($items as $item) {
    $stmt = $pdo->prepare('INSERT INTO order_items
(order_id, product_id, quantity, price) VALUES (?, ?, ?, ?)');
    $stmt->execute([$order_id, (int)$item['product_id'],
(int)$item['quantity'], (float)$item['price']]);

    $stmt = $pdo->prepare('UPDATE products SET stock =
stock - ? WHERE id = ?');
    $stmt->execute([(int)$item['quantity'],
(int)$item['product_id']]);
}

$stmt = $pdo->prepare('DELETE FROM cart WHERE user_id = ?')->execute([$user_id]);
$stmt->commit();
redirect('orders.php?success=1');
} catch (Throwable $e) {
    $pdo->rollBack();
    $error = 'Помилка оформлення замовлення.';
}
}

$page_title = 'Оформлення замовлення - MotoMarket PRO';
require_once 'includes/header.php';
?>

<div class="catalog-hero page-head">
    <span class="section-kicker">Checkout</span>
    <h1>Оформлення замовлення</h1>
    <p class="light-muted">У новій версії додано спосіб доставки,
спосіб оплати та коментар до замовлення.</p>
</div>

<?php if ($error): ?><div class="alert error"><?php echo
safe($error); ?></div><?php endif; ?>

<div class="checkout-grid">
    <div class="card">
        <h2>Дані покупця</h2>
        <form method="POST">
            <label>Ім'я</label>
            <input type="text" name="customer_name" value="<?php echo
safe($user['name']) ?>" required>
            <label>Телефон</label>
            <input type="text" name="phone" value="<?php echo
safe($user['phone']) ?>" required>
            <label>Адреса доставки</label>

```

```

        <textarea name="address" required><?php echo
safe($user['address'] ?? ''); ?></textarea>
        <label>Спосіб доставки</label>
        <select name="delivery">
            <option>Нова пошта</option>
            <option>Укрпошта</option>
            <option>Кур'єр по місту</option>
            <option>Самовивіз</option>
        </select>
        <label>Спосіб оплати</label>
        <select name="payment">
            <option>Оплата при отриманні</option>
            <option>Оплата картою</option>
            <option>Переказ на рахунок</option>
        </select>
        <label>Коментар</label>
        <textarea name="comment" placeholder="Наприклад:
зателефонувати перед відправкою"></textarea>
        <button class="btn success" type="submit">Підтвердити
замовлення</button>
    </form>
</div>
<div class="card">
    <h2>Ваше замовлення</h2>
    <?php foreach ($items as $item): ?>
        <div class="order-line">
            <span><?php echo safe($item['name']); ?> × <?php echo
(int)$item['quantity']; ?></span>
            <strong><?php echo money_format_ua($item['price'] *
$item['quantity']); ?></strong>
        </div>
    <?php endforeach; ?>
    <hr>
    <div class="order-line total">
        <span>Разом</span>
        <strong><?php echo money_format_ua($total); ?></strong>
    </div>
</div>

<?php require_once 'includes/footer.php'; ?>

```

Обробка Б.2 – Серверний сценарій profile.php

```

<?php
require_once 'config.php';

if (!is_logged_in()) {
    redirect('login.php');
}

```

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$user_id = (int)$_SESSION['user_id'];

$stmt = $pdo->prepare('SELECT * FROM users WHERE id = ?');
$stmt->execute([$user_id]);
$user = $stmt->fetch();

if (!$user) {
    redirect('logout.php');
}

$error = '';
$success = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $name = trim($_POST['name'] ?? '');
    $phone = trim($_POST['phone'] ?? '');
    $address = trim($_POST['address'] ?? '');

    if ($name === '' || $phone === '' || $address === '') {
        $error = 'Заповніть усі обов'язкові поля.';
    } else {
        try {
            $stmt = $pdo->prepare('UPDATE users SET name = ?, phone =
?, address = ? WHERE id = ?');
            $stmt->execute([$name, $phone, $address, $user_id]);

            $_SESSION['user_name'] = $name;
            $success = 'Дані профілю успішно оновлено.';

            $stmt = $pdo->prepare('SELECT * FROM users WHERE id = ?');
            $stmt->execute([$user_id]);
            $user = $stmt->fetch();
        } catch (Throwable $e) {
            $error = 'Не вдалося оновити профіль.';
        }
    }
}

$page_title = 'Особистий кабінет - MotoMarket PRO';
require_once 'includes/header.php';
?>

<div class="catalog-hero page-head">
    <span class="section-kicker">Profile</span>
    <h1>Особистий кабінет</h1>
    <p class="light-muted">Користувач може переглядати й оновлювати
контактні дані для оформлення замовлень.</p>
</div>

<?php if ($error): ?>
    <div class="alert error"><?php echo safe($error); ?></div>

```

					ІПЗ.КР.Б – 121 – 26 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

```

<?php endif; ?>

<?php if ($success): ?>
    <div class="alert success"><?php echo safe($success); ?></div>
<?php endif; ?>

<div class="checkout-grid">
    <div class="card">
        <h2>Редагування профілю</h2>
        <form method="POST">
            <label>Ім'я</label>
            <input type="text" name="name" value="<?php echo
safe($user['name'] ?? ''); ?>" required>

            <label>Телефон</label>
            <input type="text" name="phone" value="<?php echo
safe($user['phone'] ?? ''); ?>" required>

            <label>Адреса доставки</label>
            <textarea name="address" required><?php echo
safe($user['address'] ?? ''); ?></textarea>

            <button class="btn success" type="submit">Зберегти
зміни</button>
        </form>
    </div>

    <div class="card">
        <h2>Дані користувача</h2>
        <div class="order-line">
            <span>Email</span>
            <strong><?php echo safe($user['email'] ?? ''); ?></strong>
        </div>
        <div class="order-line">
            <span>Роль</span>
            <strong><?php echo safe($user['role'] ?? 'user');
?></strong>
        </div>
        <div class="order-line">
            <span>Телефон</span>
            <strong><?php echo safe($user['phone'] ?? '-');
?></strong>
        </div>
    </div>
</div>

<?php require_once 'includes/footer.php'; ?>

```

Обробка Б.3 – Обробка додавання/видалення товару з обраного

wishlist-toggle.php

```
<?php
require_once 'config.php';

if (!is_logged_in()) {
    redirect('login.php');
}

$user_id = (int)$_SESSION['user_id'];
$product_id = (int)($_POST['product_id'] ?? 0);

if ($product_id <= 0) {
    redirect($_SERVER['HTTP_REFERER'] ?? 'wishlist.php');
}

$stmt = $pdo->prepare('SELECT id FROM wishlist WHERE user_id = ? AND
product_id = ?');
$stmt->execute([$user_id, $product_id]);
$existing = $stmt->fetch();

if ($existing) {
    $del = $pdo->prepare('DELETE FROM wishlist WHERE id = ?');
    $del->execute([(int)$existing['id']]);
} else {
    $ins = $pdo->prepare('INSERT INTO wishlist (user_id, product_id)
VALUES (?, ?)');
    $ins->execute([$user_id, $product_id]);
}

$back = $_SERVER['HTTP_REFERER'] ?? 'wishlist.php';
redirect($back);
```

Обробка Б.4 – Виведення історії замовлень користувача orders.php

```
<?php
require_once 'config.php';

if (!is_logged_in()) {
    redirect('login.php');
}

$user_id = (int)$_SESSION['user_id'];

$stmt = $pdo->prepare('SELECT * FROM orders WHERE user_id = ? ORDER BY
created_at DESC');
$stmt->execute([$user_id]);
$orders = $stmt->fetchAll();

$page_title = 'Мої замовлення - MotoMarket PRO';
```

					ІІЗ.КР.Б – 121 – 26 - ІІЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

require_once 'includes/header.php';
?>

<div class="catalog-hero page-head">
    <span class="section-kicker">Orders</span>
    <h1>Мої замовлення</h1>
    <p class="light-muted">Тут відображається історія покупок, поточні
статуси та склад кожного замовлення.</p>
</div>

<?php if (isset($_GET['success'])): ?>
    <div class="alert success">Замовлення успішно оформлено.</div>
<?php endif; ?>

<div class="card">
    <?php if (!$orders): ?>
        <p>У вас ще немає замовлень.</p>
    <?php else: ?>
        <?php foreach ($orders as $order): ?>
            <div class="order-card">
                <div class="order-line">
                    <strong>Замовлення №<?php echo (int)$order['id'];
?></strong>
                    <span><?php echo safe($order['created_at']);
?></span>
                </div>
                <div class="order-line">
                    <span>Статус</span>
                    <strong><?php echo safe($order['status']);
?></strong>
                </div>
                <div class="order-line">
                    <span>Доставка</span>
                    <strong><?php echo
safe($order['delivery_method']); ?></strong>
                </div>
                <div class="order-line">
                    <span>Оплата</span>
                    <strong><?php echo safe($order['payment_method']);
?></strong>
                </div>
                <div class="order-line total">
                    <span>Сума</span>
                    <strong><?php echo
money_format_ua((float)$order['total']); ?></strong>
                </div>
            </div>
        <?php endforeach; ?>
    <?php endif; ?>
</div>

```

					ІІЗ.КР.Б – 121 – 26 - ІІЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		