

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення

комп'ютерних технологій

Наталія СТЕФУРАК / _____ /

підпис

«__» _____ 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

освітньо-професійного ступеня «фаховий молодший бакалавр»

зі спеціальності 122 «Комп'ютерні науки»

на тему: «Онлайн-система продажу товарів для манікюру»

Студентка групи КН-41

Валентина КУЛЬЧИНСЬКА

_____ /
(підпис)

Керівник роботи

Оксана СИРОТЮК

_____ /
(підпис)

Консультанти:

з техніко-економічного
обґрунтування

Любов МЕЛЕНЧУК

_____ /
(підпис)

нормоконтролер

Надія ГАВРИШКІВ

_____ /
(підпис)

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділення
комп'ютерних технологій

Наталія СТЕФУРАК / _____ /
підпис

« ___ » _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»
студентці Кульчинській Валентині Андріївні

(прізвище, ім'я та по-батькові студента)

1. Тема роботи Онлайн-система продажу товарів для манікюру

затверджена наказом по коледжу від "25" листопада 2024 р., № 253а-н

2. Термін здачі студентом завершеної роботи " ___ " _____ 202_ р.

3. Вихідні дані до роботи Результати аналізу web-базованих інформаційних систем електронної комерції та засобів їхньої реалізації

4. Перелік питань, які повинні бути розроблені:

а) основна частина Аналіз предметної області та постановка завдання, Проектування інформаційної системи, Реалізація та тестування інформаційної системи

б) техніко-економічне обґрунтування Аналіз ринку, Економічний розрахунок проектних витрат, Стратегічне обґрунтування доцільності проекту

5. Перелік графічного матеріалу Діаграма діяльності, діаграма варіантів використання, діаграма послідовності, ER-діаграма

6. Консультації роботи

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
Техніко-економічне обґрунтування	<u>Меленчук Л.І.</u> Вчений ступінь, звання, ПІП консультанта		

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1	Вибір теми, ознайомлення з вимогами.	20.11.24	25.11.24
2	Дослідження предметної області, аналіз типових рішень.	26.11.24	29.11.24
3	Проектування системи: формалізація вимог, виділення функціональних та нефункціональних вимог до ПП та проектування структури системи.	30.11.24	15.12.24
4	Побудова моделі даних, виділення основних сутностей та зв'язків між ними, проектування бази даних	16.12.24	27.01.25
5	Аналіз засобів та технологій реалізації проєкту. Встановлення налаштування необхідного програмного забезпечення	28.01.25	11.02.25
6	Реалізація бази даних.	12.02.25	15.03.25
7	Розробка програмного продукту.	16.03.25	27.04.25
8	Доопрацювання модулів, підсистем і т.д.	28.04.25	06.05.25
9	Опрацювання питань з техніко-економічного обґрунтування	07.05.25	20.05.25
10	Тестування на налагодження програмного продукту	21.05.25	25.05.25
11,	Робота над основним розділом пояснювальної записки	26.05.06.25	10.06.25
12	Оформлення пояснювальної записки, додатків, перевірка на унікальність	11.06.25	17.06.25
13	Попередній захист кваліфікаційної роботи.	13.06.25	
14	Підготовка до захисту кваліфікаційної роботи	19.06.25	23.06.25
15	Захист кваліфікаційної роботи	24.06.25	

7. Дата видачі завдання “ ___ ” _____ 2024 р. Керівник _____ /

Завдання прийняв до виконання _____ / _____ /

Реферат

Онлайн-система продажу товарів для манікюру. Кваліфікаційна робота. Кульчинська Валентина Андріївна. Галицький фаховий коледж імені В'ячеслава Чорновола, відділення комп'ютерних технологій. Спеціальність 122 «Комп'ютерні науки», 2025. Сторінок – 113, рисунків – 49, додатків – 8.

Об'єкт дослідження – web-базовані інформаційні системи електронної комерції та засоби їхньої реалізації.

Метою роботи є розробка онлан-системи продажу товарів для манікюру із реалізацією функцій каталогізації товарів, обліку замовлень, авторизації користувачів та адміністрування через веб-інтерфейс із застосуванням PHP, JavaScript, Bootstrap і MySQL.

Система повинна забезпечити введення та зберігання інформації про товари, категорії, користувачів, замовлення та способи доставки; формування каталогу товарів із відображенням зображень; реалізацію функцій реєстрації, авторизації, кошика та профілю; а також керування даними через панель адміністратора.

Для реалізації поставленої задачі було використано мову PHP, JavaScript, CSS, HTML, фреймворк Bootstrap, середовище XAMPP із MySQL (phpMyAdmin) і бібліотеку jQuery.

Результатом розробки стала завершена онлайн-система продажу товарів для манікюру, готова до впровадження.

Abstract

Online Manicure Products Sales System. Qualification Work. Kulchynska Valentyna Andriivna. Halytsky Applied College named after Vyacheslav Chornovil, Department of Computer Technologies. Specialty 122 “Computer Sciences”, 2025. Pages – 113, Figures – 49, Appendices – 8.

Object of Study: Web-based e-commerce information systems and the tools used for their implementation.

Objective: To develop an online manicure products sales system implementing product cataloging, order management, user authentication, and administration via a web interface using PHP, JavaScript, Bootstrap, and MySQL.

System Requirements: Input and storage of information on products, categories, users, orders, and delivery methods

Generation of a product catalog with image display

User registration, authentication, shopping cart, and profile functionality

Data management through an administrator panel

Tools & Technologies: PHP, JavaScript, CSS, HTML; Bootstrap framework; XAMPP environment with MySQL (phpMyAdmin); jQuery library.

Outcome: A fully developed online manicure products sales system, ready for deployment.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та постановка завдання.....	9
1.1 Аналіз предметної області.....	9
1.2 Обґрунтування доцільності розробки	10
1.3 Аналіз існуючих рішень	12
1.4 Аналіз вимог та постановка завдання	18
2 Проєктування інформаційної системи.....	20
2.1 Формалізація вимог до системи.....	20
2.2 Проєктування структури системи	24
2.2 Проєктування інтерфейсу інформаційної системи.....	25
2.4 Проєктування бази даних	30
2.5 Аналіз технологій реалізації	35
3 Реалізація та тестування інформаційної системи	43
3.1 Опис технологій та засобів реалізації	43
3.2 Реалізація бази даних	45
3.3 Реалізація основних функцій інформаційної системи	48
3.4. Тестування інформаційної системи.....	67
4 Техніко-економічне обґрунтування.....	77
4.1 Аналіз ринку	77
4.2 Економічний розрахунок проєктних витрат	78
4.3 Обґрунтування доцільності проєкту	80
Висновки	81
Додатки.....	83

					КР.КН 25.593.10.000 ПЗ			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Кульчинська В. А.			Онлайн-система продажу товарів для манікюру	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		Сидотюк О.Б.					5	115
<i>рецензен</i>		Гавришків Н.Г.				ГФК.ВКТ КН-41		
<i>Н.контр.</i>		Гавришків Н.Г.						
<i>Зав. від.</i>		Стефурак Н.А.						

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – Application Programming Interface;

CSS – Cascading Style Sheets;

cURL – Client URL;

HTML – HyperText Markup Language;

HTTP – Hypertext Transfer Protocol;

IDE – Integrated Development Environment;

JS – JavaScript;

MVC – Model-View-Controller;

MySQL – My Structured Query Language;

Node.js – Node JavaScript.

ORM – Object-Relational Mapping;

PDO – PHP Data Objects;

PHP – Hypertext Preprocessor;

REST – Representational State Transfer;

SQL – Structured Query Language;

БД – база даних;

ЄСВ – єдиний соціальний внесок;

СУБД – система управління базами даних.

					КР.КН 25.593.10.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Стрімкий розвиток цифрових технологій та електронної комерції відкриває нові горизонти для підприємницької діяльності у вузькоспеціалізованих сегментах ринку. Індустрія краси, зокрема сфера нейл-арту та професійного манікюру, демонструє стабільне зростання попиту як серед професійних майстрів, так і серед споживачів, які надають перевагу домашньому догляду за нігтями.

Сучасний ринок товарів для манікюру характеризується широким асортиментом продукції – від базових інструментів та косметичних засобів до високотехнологічного обладнання та інноваційних матеріалів. Водночас спостерігається тенденція до зростання вимог споживачів щодо якості онлайн-сервісів: інтуїтивності навігації, персоналізації пропозицій, швидкості обробки замовлень та надійності логістичних процесів.

Аналіз поточного стану ринку електронної торгівлі у даному сегменті виявив ряд проблем, а саме фрагментарність пропозицій, недостатню спеціалізацію існуючих платформ, обмежені можливості для порівняння товарів та отримання професійних консультацій, а також відсутність комплексних рішень, які б задовольняли потреби як професійних салонів краси, так і приватних користувачів.

Мета кваліфікаційної роботи полягає у створенні комплексної інформаційної системи електронної торгівлі товарами для манікюру, яка забезпечить ефективну взаємодію між постачальниками та споживачами через інтеграцію сучасних веб-технологій, адаптивного користувацького інтерфейсу та функціональних можливостей електронної комерції.

Для досягнення поставленої мети визначено наступні завдання:

- Проведення аналізу ринку у сфері онлайн-продажу товарів для манікюру.
- Формалізація функціональних та нефункціональних вимог до інформаційної системи.

					КР.КН 25.593.10.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

- Розробка архітектури системи з урахуванням принципів масштабованості, безпеки та продуктивності.
- Проектування інтуїтивного користувацького інтерфейсу з адаптивним дизайном.
- Моделювання структури бази даних для ефективного зберігання та обробки інформації.
- Обґрунтування вибору технологічного стеку для реалізації веб-додатку.
- Інтеграція платіжних систем та логістичних сервісів.
- Впровадження механізмів забезпечення інформаційної безпеки та захисту персональних даних.

Актуальність цієї роботи полягає в розробці онлайн-платформи для продажу манікюрних товарів із врахуванням специфіки ринку та сучасного рівня розвитку інформаційних технологій. Така система об'єднує сучасні веб-технології (PHP, MySQL, Bootstrap, JavaScript) та інтеграцію з сервісами доставки (Нова Пошта, УкрПошта), що забезпечує інтуїтивно зрозумілий та безпечний інтерфейс як для професійних майстрів, так і для звичайних споживачів.

Практична значущість роботи полягає в тому, що розроблювана інформаційна система може бути безпосередньо використана для створення комерційного онлайн-магазину манікюрних товарів. Це сприятиме розширенню каналів збуту та підвищенню доступності якісних продуктів завдяки ефективній організації процесів вибору, замовлення та доставки. Крім того, сучасний технологічний стек і архітектурні рішення дають можливість подальшого масштабування та адаптації системи під інші напрями електронної торгівлі.

					КР.КН 25.593.10.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз предметної області

У сучасних умовах зростаючої популярності електронної комерції все більше підприємств прагнуть перенести свою діяльність у цифровий простір, забезпечуючи клієнтам доступ до товарів та послуг у зручному онлайн-форматі. Це явище особливо помітне у сфері роздрібної торгівлі, де швидкість обслуговування, асортимент і зручність взаємодії з покупцем відіграють ключову роль. Ринок товарів для манікюру не є винятком. У цьому сегменті споживачами виступають як професійні салони краси та майстри нігтьового сервісу, так і звичайні користувачі, які доглядають за нігтями вдома.

Протягом останніх років індустрія манікюру стрімко зростає. За даними аналітиків, ринок продуктів для нігтьового сервісу щороку збільшується на 6-8%, з'являється багато нових матеріалів та інструментів для створення дизайнів манікюру. Особливо помітне зростання в сегменті гель-лаків та інструментів для домашнього використання. Пандемія лише посилила цю тенденцію, оскільки багато людей почали самостійно робити манікюр вдома. Створення спеціалізованого інтернет-магазину дозволить зайняти нішу на цьому перспективному ринку.

Зручна, ефективна та доступна онлайн-система продажу товарів для манікюру дозволяє забезпечити постійний зв'язок із клієнтами, розширити аудиторію, автоматизувати процеси обробки замовлень і покращити якість обслуговування.

Ринок манікюрної продукції включає широкий асортимент товарів, таких як матеріали для покриття нігтів, інструменти для манікюру, фрезери, лампи, витяжки та різноманітні аксесуари. Для ефективної торгівлі таким різноманіттям продукції важливо мати систему, яка дозволяє: класифікувати товари за категоріями, фільтрувати за брендами, ціною, наявністю, надавати детальну

					КР.КН 25.593.10.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

інформацію та фото товарів, реалізувати функції кошика, обліку замовлень і реєстрації користувачів.

Функціональність системи має забезпечувати зручність користування для клієнтів. Це передбачає наявність детального каталогу з зображеннями та описами, інтуїтивно зрозумілу навігацію та ефективну систему пошуку й фільтрації. Користувачам необхідно надати можливість зберігати обрані товари, мати власний особистий кабінет та декілька зручних способів оплати.

Технічна реалізація системи має забезпечувати її стабільну роботу на різних пристроях, швидке завантаження сторінок та надійний захист даних користувачів. Оптимізація для пошукових систем допоможе залучати нових клієнтів.

1.2 Обґрунтування доцільності розробки

У зв'язку із зростанням попиту виникає потреба в ефективному інструменті, що дозволяє зручно здійснювати замовлення, переглядати асортимент, обирати способи оплати та доставки, а також отримувати своєчасну інформацію про новинки й акції продукції для манікюру. Онлайн-система продажу товарів для манікюру стає не просто веб-сайтом із каталогом — це повноцінна платформа, яка забезпечує постійний зв'язок між продавцем і покупцем, створює комфортні умови для придбання продукції, оптимізує логістику замовлень та дозволяє бізнесу працювати безперервно 24/7 без необхідності клієнтам відвідувати фізичний магазин.

Окрім цього, для клієнтів важлива можливість детально вивчити характеристики товарів, порівняти ціни. Також інтернет-магазин дозволяє швидко поповнювати запаси витратних матеріалів без відриву від роботи.

Використання такої системи сприяє розширенню цільової аудиторії, оскільки дає змогу охопити покупців з різних регіонів, у тому числі віддалених. Крім того, автоматизація процесів – таких як облік товарів, обробка замовлень, управління залишками на складі та аналітика продажів дозволяє значно знизити

					КР.КН 25.593.10.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

витрати ресурсів і мінімізувати вплив людського фактору. Впровадження онлайн-системи є вигідним рішенням як для підприємців, що прагнуть масштабувати свою діяльність, так і для клієнтів, які очікують високої якості обслуговування та швидкої доставки необхідних товарів.

У порівнянні з фізичним магазином, онлайн-система не потребує оренди приміщення, що значно знижує постійні витрати. При таких обставинах не потрібно витрачати кошти на оформлення вітрин, торгового обладнання та великої кількості персоналу для обслуговування клієнтів. Замість цього, можна інвестувати в якісний склад, систему обробки замовлень та доставку. Це дозволяє встановлювати більш конкурентні ціни.

Онлайн-формат дозволяє надати вичерпну інформацію про кожен товар: фотографії з різних ракурсів, відео огляди, детальні характеристики, інструкції із застосування.

Сучасні системи електронної комерції дозволяють аналізувати поведінку користувачів і формувати індивідуальні рекомендації. Також можливе налаштування персоналізованих знижок для постійних клієнтів або професійних майстрів, які роблять великі оптові закупівлі.

Інтернет-магазин дозволяє автоматизувати значну частину операцій: прийом замовлень, обробку платежів, формування документів для відправки. Система може автоматично контролювати залишки на складі та сповіщати про необхідність поповнення асортименту. Це значно підвищує ефективність роботи і зменшує ймовірність людських помилок.

Впровадження такої системи дозволить не лише ефективно вести бізнес у сфері манікюрних матеріалів, але й створить надійну та зручну основу для подальшого масштабування та розширення асортименту супутніх товарів.

Таким чином, онлайн-система продажу продукції для манікюру є актуальним і затребуваним інструментом, що поєднує в собі переваги сучасних технологій, зручність електронної торгівлі та потреби ринку у спеціалізованому, функціональному та доступному сервісі.

					КР.КН 25.593.10.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1.3 Аналіз існуючих рішень

Детальний аналіз функціоналу існуючих інтернет-магазинів дозволяє виявити їхні сильні та слабкі сторони. Це створює можливість розробити систему, яка пропонує кращі рішення і заповнює прогалини конкурентів.

Вивчаючи недоліки існуючих платформ, можна заздалегідь уникнути типових проблем і створити більш досконалий продукт з першої спроби. Розуміння, які функції справді використовуються користувачами, а які залишаються не потрібними, допомагає оптимізувати бюджет і зосередитись на розробці дійсно необхідних елементів.

Для дослідження було обрано сайт-магазин матеріалів для манікюру «Furman» <https://furman.ua/uk/>.

Відкривши цю онлайн-систему одразу можна побачити усі необхідні пункти меню на головній сторінці (рис. 1.1).

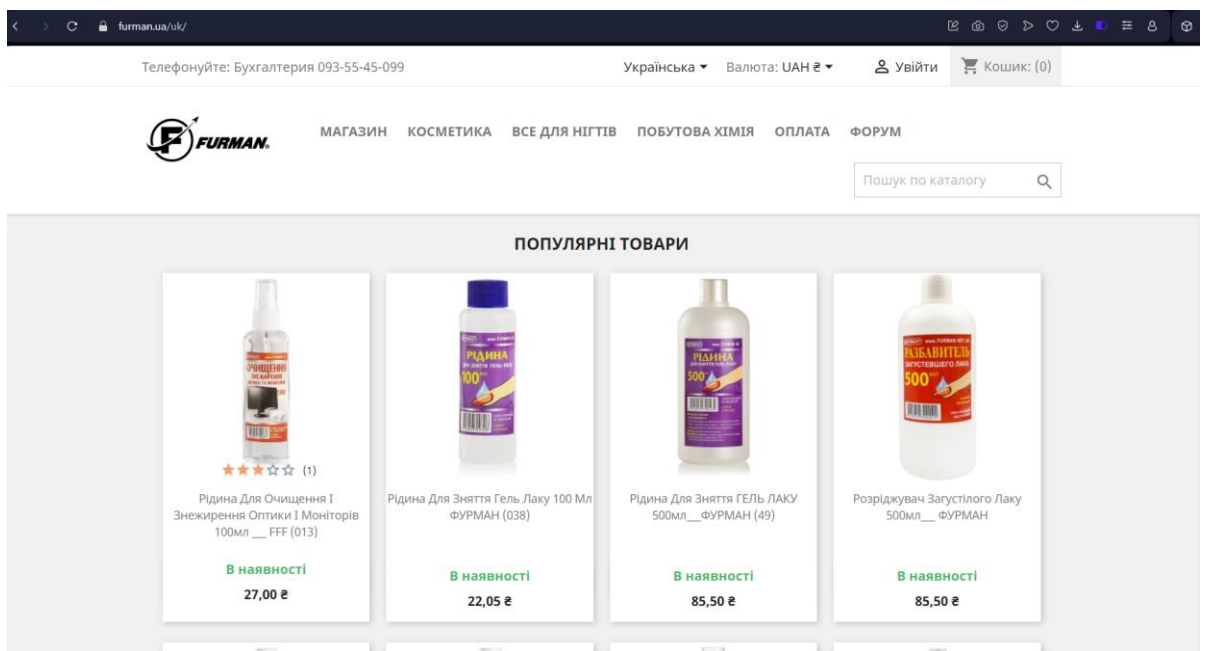


Рисунок 1.1 – Головна сторінка сайту «Furman»

Звернувши увагу на стиль сайту можна зробити висновок, що обрана саме така кольорова гама щоб зосередити увагу на обранні товару. Проте як відомо більшість людей обираючи де здійснити покупки, звертають першочергово увагу

					КР.КН 25.593.10.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

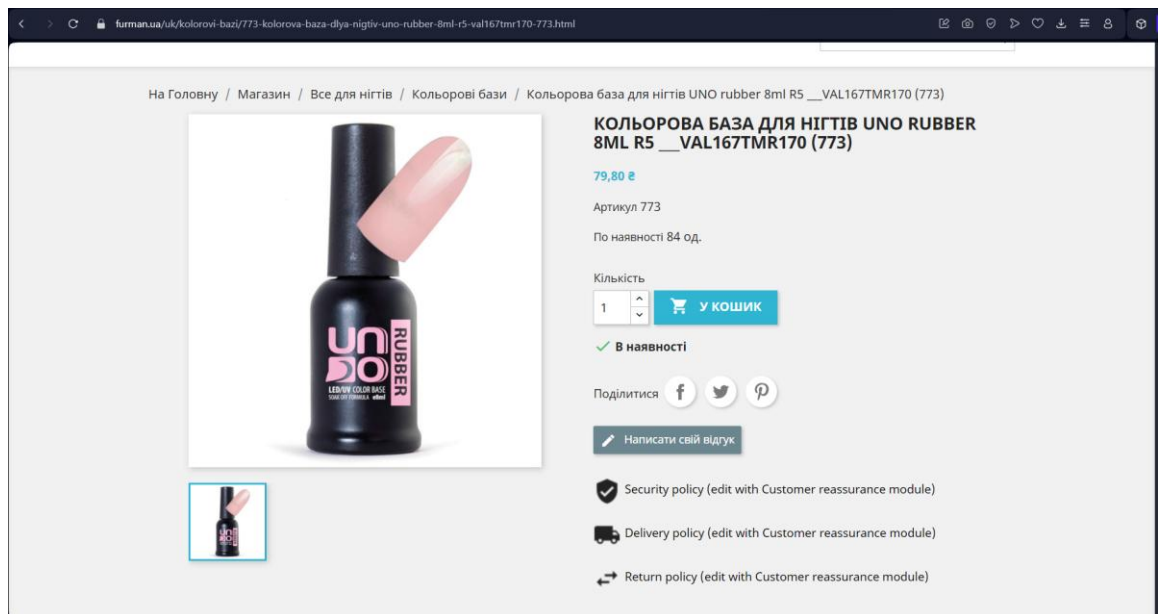


Рисунок 1.3 – Сторінка обраного товару

Після натиснення на кнопку додати у кошик товар відображається у відповідному списку товарів до покупки. Одразу у кошику є можливість перейти до оформлення замовлення (рис. 1.4).

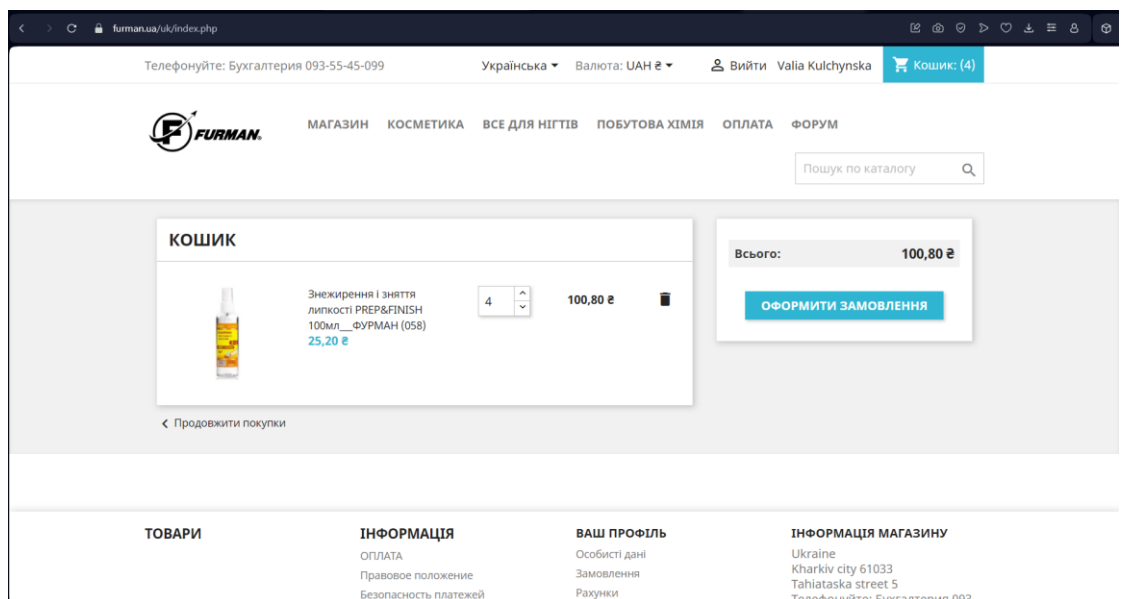


Рисунок 1.4 – Кошик товарів

Відкривається сторінка на якій користувач вводить необхідні дані для відправки, у тому числі спосіб доставки та місце (рис 1.5).

					КР.КН 25.593.10.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Область

Місто

Відділення

Адресна доставка

Рисунок 1.5 – Вибір місця доставки

Наступним кроком буде вибір способу оплати (рис 1.6).

Рисунок 1.6 – Способи оплати

Після введення усієї важливої інформації для відправки натиснувши на кнопку завершення оформлення замовлення відкривається сторінка із платіжною системою (рис 1.7).

Рисунок 1.7 – Сторінка з платіжною системою

Змн.	Арк.	№ докум.	Підпис	Дата

КР.КН 25.593.10.000 ПЗ

Арк.

15

перевантаженим рекламними вставками й спливаючими вікнами, які відволікають увагу від головного – перегляду і купівлі продукту (рис. 1.9).

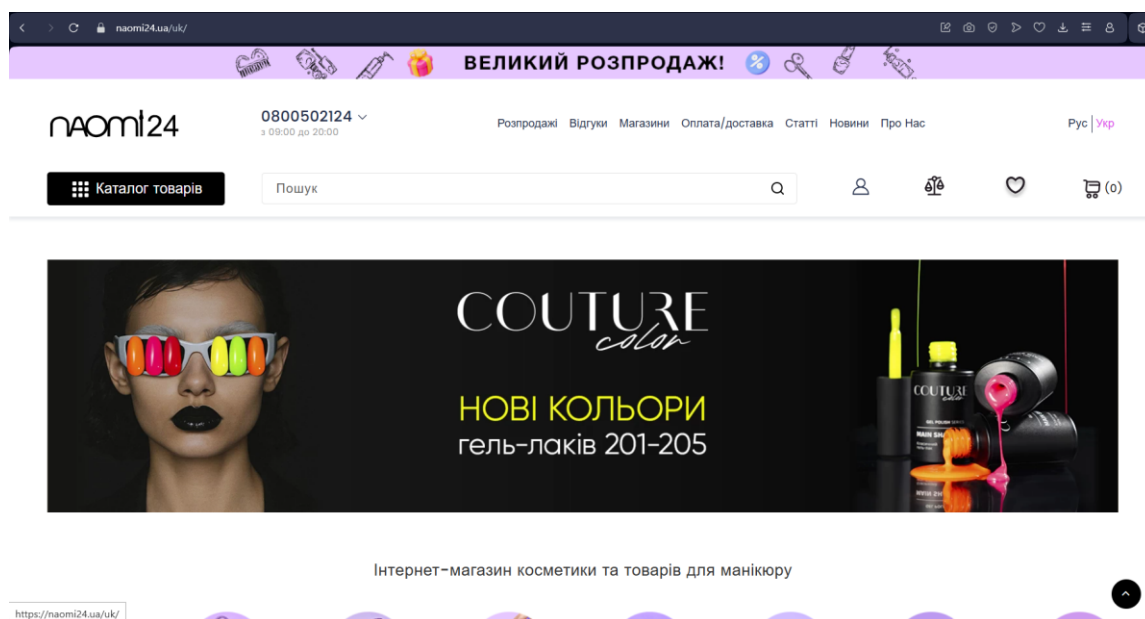


Рисунок 1.9 –Головна сторінка

Під час прокручування мобільна версія іноді підвантажує великі зображення без «лінивої» загрузки, що призводить до затримок і неплвної роботи.

При оформленні замовлення селекції для вибору відділення Нової Пошти часто залишаються порожніми, а помилки JavaScript змушують повертатися назад і повторювати спробу. Блок із кнопкою «Оформити замовлення» розташований занадто низько, і користувачі змушені багато скролити. Загалом кольорова гама та розмітка елементів інтерфейсу не завжди узгоджуються між собою: місцями контрастні банери конфліктують із стриманими тонами каталогу, а шрифти й відступи потребують уніфікації.

Власне, для нової системи важливо зробити інтерфейс легким та інтуїтивно зрозумілим. Варто зосередитися на мінімалістичному дизайні без зайвих відволікаючих елементів, оптимізувати завантаження картинок і забезпечити надійну роботу форм із вибором способу доставки та відділення, аби клієнт міг швидко і безперешкодно завершити покупку.

					КР.КН 25.593.10.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

1.4 Аналіз вимог та постановка завдання

Для клієнтської частини інформаційної системи важливо реалізувати зручну систему реєстрації та авторизації, яка дозволить користувачам швидко створювати облікові записи.

Каталог товарів потребує продуманої структури – важливо забезпечити інтуїтивну навігацію між різними групами товарів: гель-лаками, базами і топами, інструментами, акриловими системами, засобами для дизайну та аксесуарами. Кожна категорія повинна мати фільтри, що дозволять користувачам швидко знаходити потрібні товари за брендом, ціною, кольором, властивостями та іншими характеристиками. Важливо впровадити систему пошуку товарів для користувачів з конкретними побажаннями.

Процес оформлення замовлення має бути максимально спрощеним, з мінімальною кількістю кроків. Система повинна підтримувати різні способи доставки, зокрема через популярні в Україні служби. Щодо оплати, необхідно інтегрувати різноманітні методи: банківські карти, електронні гаманці та безконтактні методи оплати.

Особистий кабінет користувача має слугувати центром управління всією взаємодією з магазином. Він повинен містити історію замовлень з можливістю повторного замовлення одним кліком, збережені адреси доставки, список бажань для відкладених товарів.

Адміністративна частина системи має забезпечувати повний контроль над усіма аспектами управління магазином. Інтерфейс для керування товарами повинен дозволяти легко додавати нові позиції, редагувати існуючі та керувати їх доступністю на сайті.

Система управління замовленнями потребує зручного інтерфейсу для обробки нових замовлень, зміни їх статусу та формування документів для відправки.

Безпека системи є не менш важливим аспектом, особливо враховуючи необхідність збору та зберігання персональних даних клієнтів та інформації про

					КР.КН 25.593.10.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

платежі. Система повинна повністю відповідати вимогам законодавства щодо захисту персональних даних, включаючи можливість для користувачів контролювати свої дані та видаляти їх за запитом.

Інтерфейс має бути інтуїтивно зрозумілим для різних категорій користувачів, від досвідчених майстрів манікюру до початківців, які лише пробують себе в цій галузі. Дизайн повинен адаптуватися під різні пристрої – від настільних комп'ютерів до мобільних телефонів, забезпечуючи однаково зручний доступ до всіх функцій сайту.

Цільова аудиторія проекту є досить різноманітною і включає декілька основних груп з різними потребами. Професійні майстри манікюру становлять значну частину потенційних клієнтів – вони потребують регулярного поповнення професійних матеріалів, цікавляться новинками та прагнуть оптимізувати свої витрати через оптові закупівлі. Любителі домашнього манікюру шукають доступні товари, а також прагнуть простого інтерфейсу та зрозумілих описів.

					КР.КН 25.593.10.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Формалізація вимог до системи

Онлайн-система продажу товарів для манікюру повинна забезпечувати зручну взаємодію між користувачами (клієнтами, адміністраторами, комплектувальниками) та базою даних через веб-інтерфейс. Алгоритм її функціонування умовно поділяється на кілька ключових етапів:

1) Доступ до системи.

Користувач переходить на головну сторінку сайту, де доступний каталог товарів, фільтрація, пошук, реєстрація та вхід у профіль. Перегляд товарів можливий навіть без авторизації, однак для оформлення замовлення потрібна реєстрація або вхід.

2) Авторизація та автентифікація.

Користувач вводить логін (email) і пароль. Дані перевіряються на сервері шляхом запиту до бази даних. У разі успішної авторизації відкривається доступ до особистого кабінету, кошика, історії замовлень тощо.

3) Робота з каталогом.

Користувач переглядає товари, обирає категорії, бренди, застосовує фільтри та сортування. Товари відображаються у вигляді карток із фото, назвою, ціною та кнопкою «Додати до кошика» або «В обране».

4) Оформлення замовлення.

Обрані товари додаються до кошика. Користувач переходить до оформлення замовлення, де: перевіряє вміст кошика; обирає спосіб доставки (наприклад, «Нова Пошта», «Укрпошта»); обирає спосіб оплати (наприклад, післяплата, онлайн-оплата); підтверджує замовлення.

5) Збереження замовлення.

Система створює новий запис у таблиці бази даних, а також відповідні записи для кожного товару. Замовлення набуває початкового статусу (наприклад, «Очікує обробки»).

					КР.КН 25.593.10.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

6) Комплектування замовлень.

Комплектувальник отримуватиме інформацію про нові замовлення через адміністративну панель. Він повинен бачити список замовлених товарів і деталі доставки, після чого змінювати статус на «Виконується» або «Відправлено».

7) Функціонал адміністратора.

Адміністратор повинен мати доступ до додавання, редагування та видалення товарів; управління категоріями; перегляду всіх замовлень і користувачів; оновлення контенту на сайті.

8) Підтримка та завершення циклу замовлення товару.

Користувач отримуватиме повідомлення про статус замовлення (на сайті або через email). Після завершення замовлення дані зберігаються в історії, і користувач може повторно переглядати або дублювати замовлення.

Загальний алгоритм функціонування системи подано на рисунку 2.1.

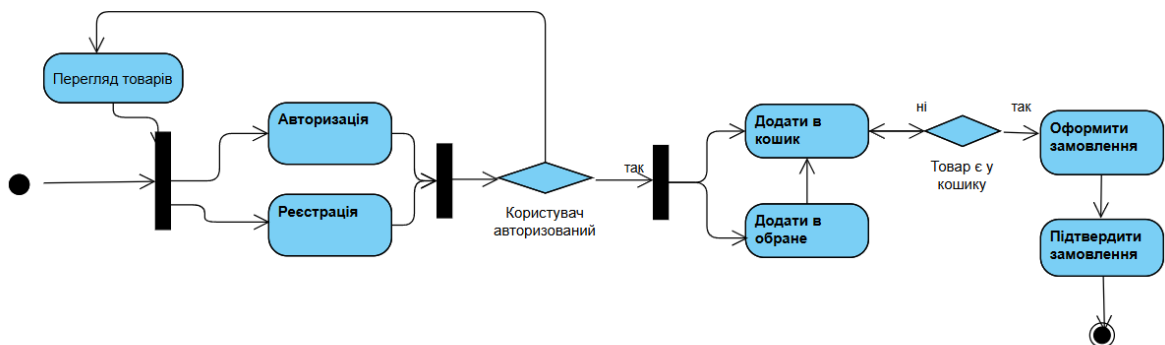


Рисунок 2.1 – Загальний алгоритм функціонування системи

Для реалізації онлайн-системи продажу товарів для манікюру необхідно здійснити аналіз функціональних і нефункціональних вимог. В процесі визначено такі основні функції:

- реєстрація та авторизація користувачів для забезпечення безпечного доступу до особистого кабінету;
- управління каталогом товарів для структурування продукції за категоріями та брендами;

- обробка замовлень, яка включає механізм формування кошика, оформлення та відстеження замовлень;
- робота користувача з особистим кабінетом, де можна переглядати свої дані, замовлення та збережені товари;
- адміністративна панель, що дозволяє керувати товарами, замовленнями та користувачами.

З огляду на велику кількість взаємопов'язаних функцій, для кращого розуміння структури системи було створено UML-діаграму прецедентів, яка відображає взаємодію між основними користувачами системи (акторами) та її функціональними можливостями. На діаграмі наведено дії, доступні кожному типу користувача: клієнту – перегляд товарів, реєстрація і додавання до кошика, оформлення замовлень, доступ до історії покупок та редагування профілю; комплектувальнику замовлень – перегляд списку активних замовлень, деталей замовлення, зміна статусу замовлення (наприклад, «збирається», «відправлено»), адміністратору — повний доступ до управління базою товарів, управління користувачами та контентом сайту тощо (рис. 2.2).



Рисунок 2.2 – Діаграма прецедентів

Побудова даної моделі забезпечує основу для подальшого етапу реалізації системи. Завдяки графічному представленню функціональних вимог спрощується розробка та тестування, а також зменшується ризик непорозумінь між замовником і розробником.

Усі типи користувачів мають доступ до персоналізованого інтерфейсу відповідно до своїх ролей. Діаграма демонструє залежності між прецедентами, зокрема використання підтвердження електронної пошти після реєстрації, а також спадкування ролей користувачів у межах адміністративного контролю.

Функціональність інформаційної системи ґрунтується на забезпеченні зручної та захищеної взаємодії між користувачем і платформою. Основна функція – обробка замовлень – реалізується через кошик, що зберігає вибрані товари навіть без реєстрації. Автоматичний перерахунок вартості з урахуванням знижок, вибір способу доставки та способу оплати, а також можливість залишити коментар – усе це формує логіку покупки. Додатково система автоматично генерує рахунок на оплату.

Каталог товарів підтримує багаторівневу структуру категорій, розширену систему фільтрації (за брендом, кольором, ціною тощо), а також пошук з автодоповненням і можливість сортування результатів. Функція порівняння характеристик дозволяє приймати більш обґрунтовані рішення при виборі.

Особистий кабінет користувача реалізує можливість перегляду, редагування персональних даних, повторного замовлення, перегляду історії покупок та керування списком бажань. Особлива увага приділяється збереженню даних доставки для подальшого використання, а також реалізації права на видалення облікового запису.

Адміністративна частина надає повний контроль над вмістом сайту. Адміністратор може додавати, редагувати або приховувати товари, змінювати категорії, параметри фільтрів, обробляти замовлення, оновлювати контент та керувати доступом інших адміністраторів.

					КР.КН 25.593.10.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2 Проектування структури системи

В результаті аналізу вимог до системи було здійснено проектування її структури, яка визначає архітектуру, основні модулі, взаємозв'язки між компонентами та логіку взаємодії користувача із системою.

На основі виділених функціональних вимог було розроблено логічну модель системи, що включає ключові підсистеми: модуль реєстрації та авторизації, модуль управління каталогом, модуль обробки замовлень, особистий кабінет користувача та адміністративну панель.

Кожен компонент має чітко визначене призначення і взаємодіє з іншими елементами системи через серверну частину, яка забезпечує збереження, обробку та передачу даних. Для візуалізації логічної структури була створена діаграма компонентів, яка наочно демонструє архітектуру системи та зв'язки між її частинами.

Система спроектована за класичною трирівневою архітектурою, що включає:

Рівень презентації (клієнтський інтерфейс) – веб-сторінки, з якими взаємодіє користувач.

Рівень логіки (серверна частина) – обробляє запити, взаємодіє із базою даних, керує авторизацією, обробкою замовлень тощо.

Рівень даних (база даних) – зберігає інформацію про товари, замовлення, користувачів, категорії тощо.

Такий підхід дозволяє досягти високого рівня модульності, полегшує підтримку й масштабування проекту, а також створює надійний фундамент для реалізації функціоналу на етапі програмної реалізації.

Для ефективного функціонування системи необхідно реалізувати взаємодію між модулями через чітко визначені запити та відповідні обробники на сервері. Наприклад: модуль авторизації передаватиме ідентифікатор користувача в сесію, який надалі використовуватиметься для отримання персоналізованих даних; кошик передаватиме дані в модуль замовлення, який створюватиме

					КР.КН 25.593.10.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

відповідні записи в таблицях orders та order_items; адміністративна панель повинна взаємодіяти з модулями товарів, користувачів і замовлень через CRUD-операції тощо. Основні процеси взаємодії подані на діаграмі послідовності (рис. 2.3).

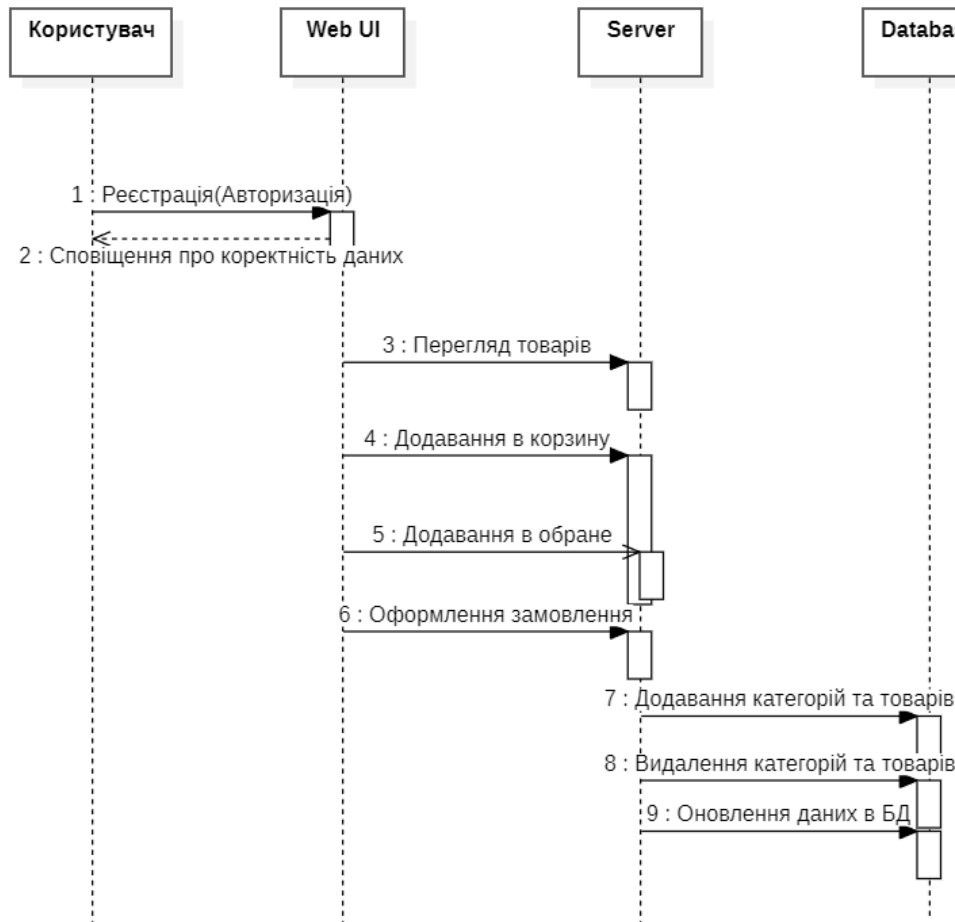


Рисунок 2.3 – Діаграма послідовності

Проектування структури системи дозволило визначити її логічну побудову, функціональні модулі та принципи взаємодії між ними. Це створює основу для подальшої реалізації функціоналу та забезпечує зрозумілу і підтримувану архітектуру.

2.2 Проектування інтерфейсу інформаційної системи

Інтерфейс системи має бути інтуїтивно зрозумілим та доступним на всіх типах пристроїв. Інтерфейс покупця складається з головної сторінки, каталогу,

сторінки товару, кошика, сторінки оформлення замовлення, особистого кабінету та списку бажань.

Дизайн побудований на принципах UI/UX. Планується використання адаптивної верстки для забезпечення коректної роботи на смартфонах і планшетах [4].

Головна сторінка є вітриною сайту та першою точкою взаємодії користувача з онлайн-системою. Її структура спрямована на приваблення уваги відвідувача, ознайомлення з брендом, пропозиціями та швидкий перехід до ключових розділів. Інтерфейс головної сторінки реалізовано у зручному, адаптивному форматі, що підтримує навігацію як з ПК, так і з мобільних пристроїв. На головній сторінці буде розміщено коротку інформацію про магазин та рекомендовані товари (акційні товари та нову колекцію). Макет подано на рисунку 2.4.



Рисунок 2.4 – Макет головної сторінки

Форми реєстрації, авторизації та редагування профілю матимуть мінімальну кількість полів, підказки та валідацію введених даних (рис. 2.5).

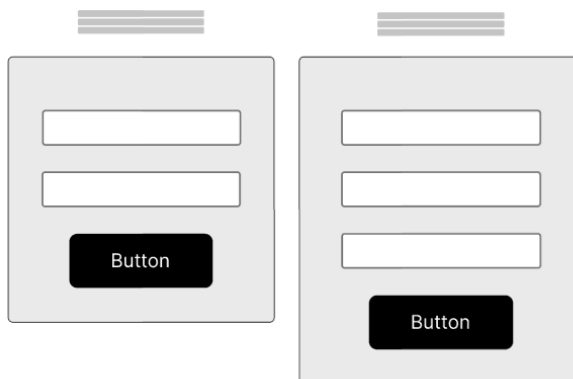


Рисунок 2.5 – Макети форм для авторизації та реєстрації

Сторінка каталогу забезпечує зручний доступ користувача до асортименту товарів для манікюру. Вона реалізована у вигляді інтерактивного списку продукції з можливістю фільтрації, сортування, переходу до сторінки товару та додавання товарів до кошика або списку бажаного (рис. 2.6).



Рисунок 2.6 – Макет сторінки каталогу товарів

Пошук розташований на видному місці сторінки та реагує на запити в реальному часі.

Макет сторінки з описом конкретного товару подано на рисунку 2.7.

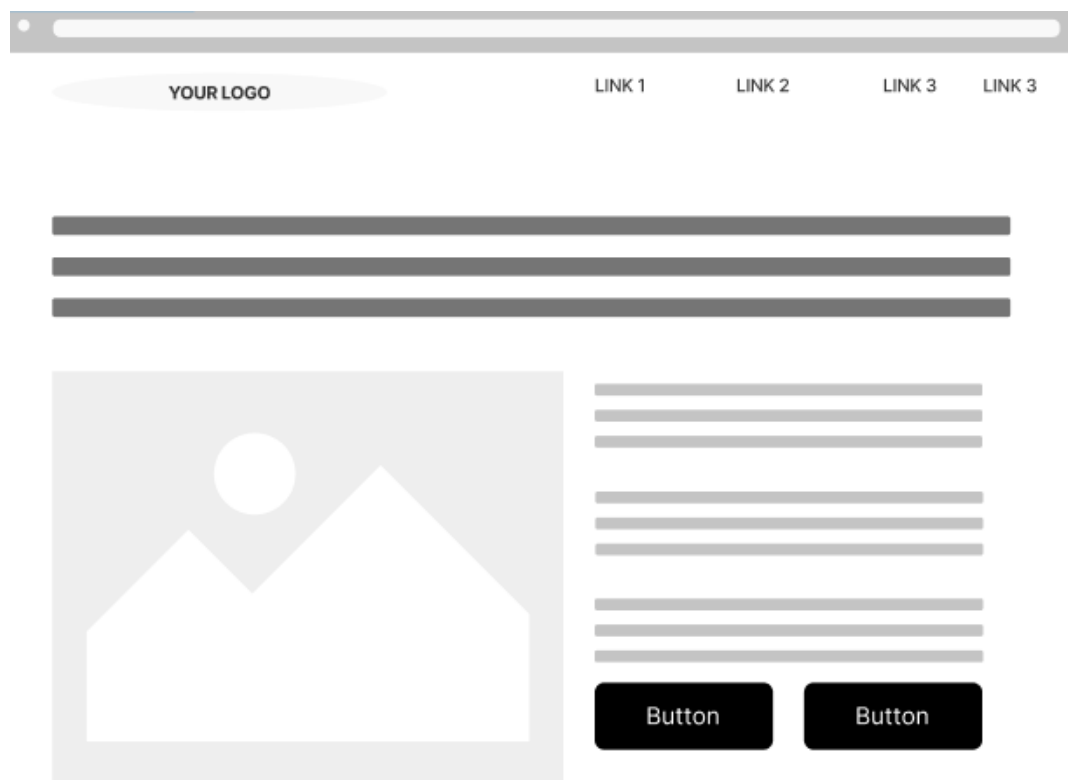


Рисунок 2.7 – Макет сторінки опису товару

На цій сторінці користувачі зможуть переглянути детальну інформацію про товари, їхню вартість та додати товар у кошик та/або бажане.

Кошик містить перелік доданих товарів, можна переходити до оформлення замовлення одним кліком.

Весь процес взаємодії побудований за принципом мінімізації кількості кліків (рис. 2.8).

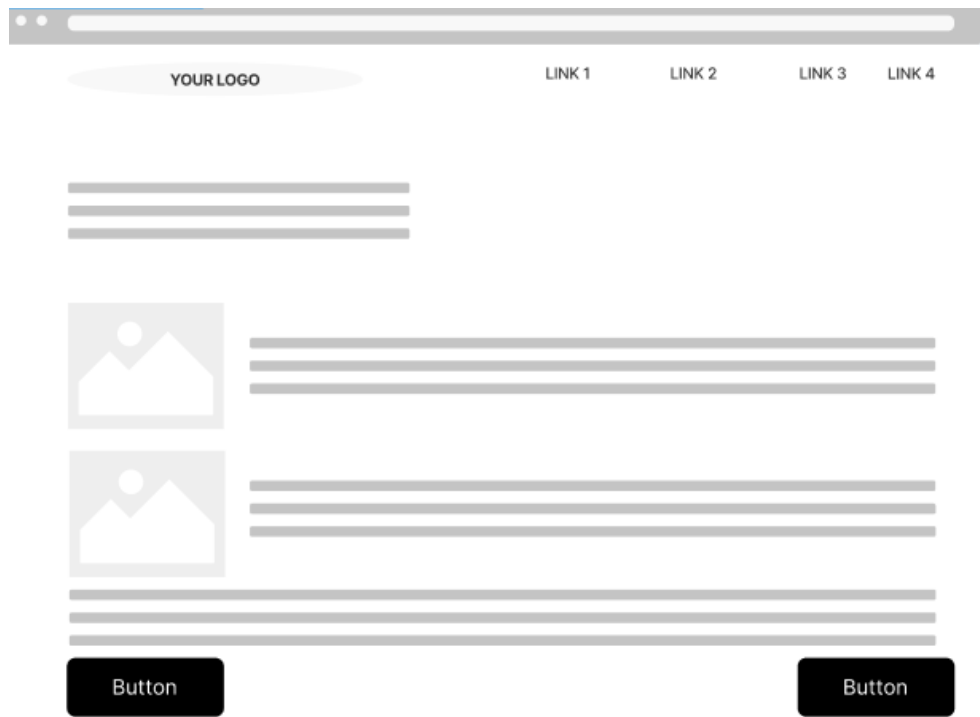


Рисунок 2.8 – Макет «кошика»

Інтерфейс адміністратора містить панель навігації з розділами керування товарами, замовленнями, користувачами, контентом.

Для кожного блоку доступні форми введення, таблиці зі статусами, пошук і фільтрація для зручності керування великим обсягом даних (рис. 2.9).

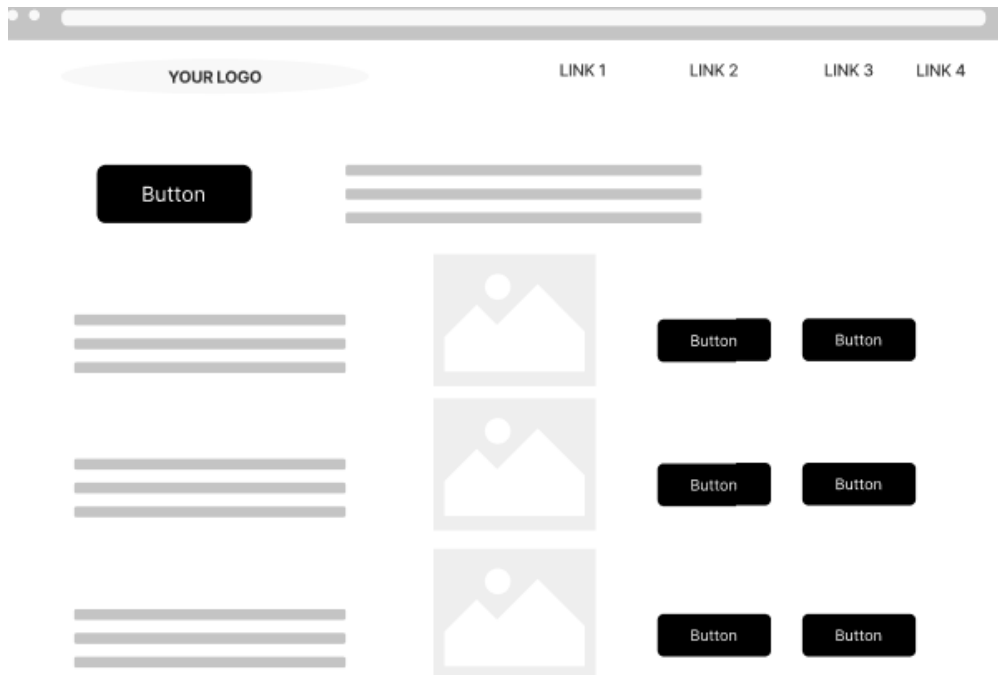


Рисунок 2.9 – Макет панелі адміністратора

Спроектований інтерфейс онлайн-системи продажу товарів для манікюру орієнтований на користувача, простий та логічно структурований з урахуванням сучасних вимог до зручності використання: всі основні функції – перегляд товарів, додавання до кошика, оформлення замовлення, авторизація та перегляд профілю – розміщені у доступних і очікуваних місцях. Меню має чітку ієрархію, адаптовану до різних ролей користувачів.

Всі елементи інтерфейсу згруповані логічно: каталог з фільтрами, картки товарів, кошик та панель користувача, що дозволяє мінімізувати кількість дій для досягнення цілі (наприклад, покупки).

Таким чином, інтерфейс забезпечує швидке ознайомлення нових користувачів із функціоналом системи та сприяє ефективній роботі постійних клієнтів.

2.4 Проектування бази даних

Проектування бази даних виконується з дотриманням принципів нормалізації для зменшення надлишковості та забезпечення цілісності даних.

Структура бази даних охоплює ключові сутності, які відповідають бізнес-логіці онлайн-магазину: користувачі, товари, категорії, замовлення, адреси доставки та інші.

Кожна таблиця має чітке призначення та забезпечує збереження важливої інформації, необхідної для функціонування системи.

Таблиця users (Користувачі).

Зберігає всі облікові записи користувачів магазину як клієнтів, так і адміністраторів з комплектувальниками.

Таблиця використовується для аутентифікації, авторизації та персоналізації функцій (табл. 2.1).

					КР.КН 25.593.10.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 – Таблиця з інформацією про користувача

Поле	Тип	Обмеження	Опис
id	int	Primary key, auto_increment	Ідентифікатор користувача
email	varchar	Unique, not null	Електронна пошта
Password_hash	varchar	Not null	Пароль
name	varchar	Nullable	Ім'я користувача
role	Enum()	Default «client», not null	Рівень користувача
Created_at	datetime	Default current_timestamp	Дата реєстрації

Встановлений параметр унікальний email при використанні якого неможливо створити декілька облікових записів із однією електронною поштою та створити запис без пароля.

Таблиця products (Товари)

Каталог продукції магазину. Забезпечує зберігання основної інформації про товар: назва, опис, бренд, ціна, залишок, категорія, тощо (таблиця 2.2).

Таблиця 2.2 – Таблиця з інформацією про товари

id	int	primary key, auto_increment	ID товару
name	varchar	not null	Назва товару
description	text	nullable	Опис товару
price	decimal	not null, check (price >= 0)	Поточна ціна
brand	varchar	nullable	Назва бренду
stock_quantity	int	default 0, check (stock_quantity >= 0)	Кількість на складі
category_id	int	foreign key categories(id)	Категорія
is_visible	boolean	default true	Показ товару

Дає змогу обрати адресу при оформленні, зручно для постійних клієнтів.

Таблиця wishlists (Списки бажань)

Зберігає вподобані товари, які користувач хоче придбати пізніше (таблиця 2.7).

Таблиця 2.7 – Таблиця з інформацією про список бажаного

id	int	primary key, auto_increment	Унікальний запис
user_id	int	foreign key → users(id)	Хто додав товар
product_id	int	foreign key → products(id)	Який товар

Дані нормалізовано аби уникнути дублювання, усі зв'язки – через ключі.

На рисунку 2.2 подано відповідну ER-діаграму.

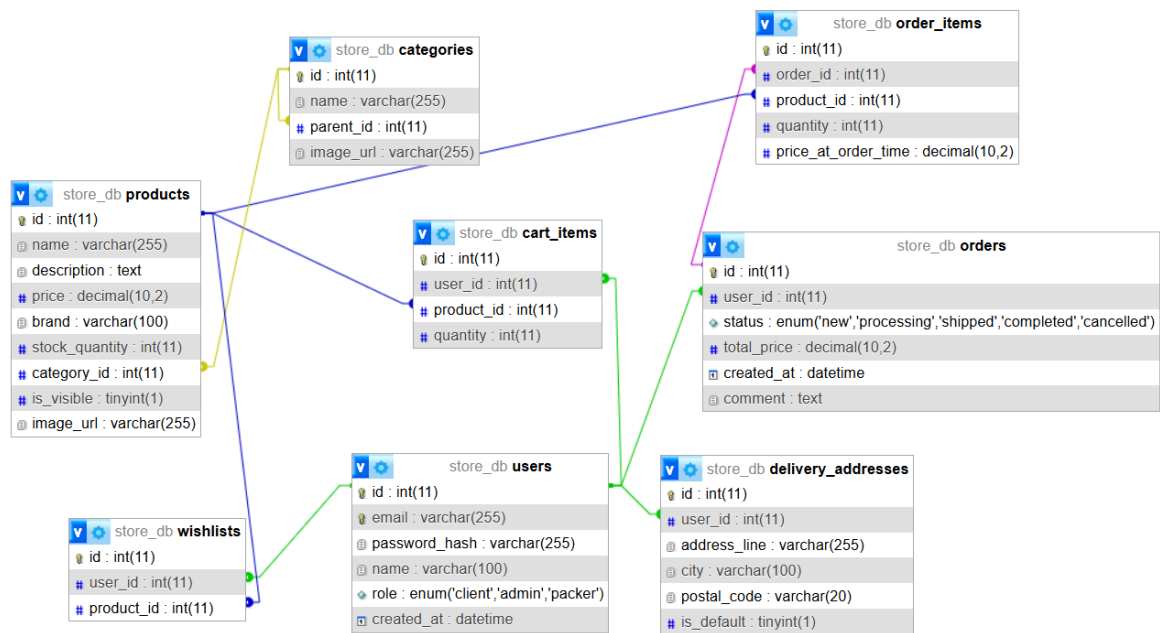


Рисунок 2.2 – ER-діаграма

У процесі проєктування бази даних для онлайн-системи продажу товарів для маніюру було розроблено чітку структуру бази даних, яка відображає всі ключові сутності предметної області: користувачі, товари, категорії, замовлення, список бажаного, кошик тощо. Логічна цілісність даних забезпечується з

допомогою зовнішніх ключів та обмежень. Наприклад, замовлення не може бути створене без відповідного користувача, а товар у замовленні повинен існувати в таблиці products.

Також враховано уникнення дублювання інформації завдяки нормалізації бази даних: кожна таблиця виконує конкретну функцію, а зв'язки між сутностями реалізовано через ідентифікатори.

Підтримується багаторівнева взаємодія користувачів завдяки зберіганню ролей (адміністратор, клієнт, комплектувальник) у таблиці users, що дозволяє реалізувати диференційований доступ до функцій системи. Оптимізація роботи з замовленнями забезпечується шляхом використання окремої таблиці – order_items для зберігання товарів у кожному замовленні, що дозволяє зберігати будь-яку кількість товарів у рамках одного замовлення.

Підвищено рівень безпеки – для зберігання паролів у таблиці users використовується поле password_hash, що дозволяє реалізувати безпечну авторизацію. Також закладено основу для аналітики, адже база підтримує зберігання історії замовлень, що дає змогу у майбутньому реалізувати модулі статистики, звітів, аналізу продажів тощо.

Таким чином, розроблена база даних повністю відповідає вимогам до сучасної комерційної системи, є функціонально повною, логічно цілісною, масштабованою та готовою до подальшої інтеграції з веб-інтерфейсом користувача.

2.5 Аналіз технологій реалізації

Оскільки типова структура веб-орієнтованої інформаційної системи включає три рівні — презентаційний (клієнтська частина), логічний (серверна частина) та рівень даних, необхідно провести аналіз засобів реалізації кожного з цих компонентів.

Клієнтська частина може бути реалізована з використанням HTML, CSS, JavaScript та сучасних фреймворків (React, Vue, Bootstrap тощо).

					КР.КН 25.593.10.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Основне завдання — забезпечити зручний та інтуїтивно зрозумілий користувацький інтерфейс.

Серверна частина відповідає за бізнес-логіку, обробку запитів, авторизацію користувачів, взаємодію з базою даних. Тут застосовуються серверні мови програмування (PHP, Python, JavaScript/Node.js) та фреймворки (Laravel, Django, Express).

Для зберігання структурованої інформації, забезпечення доступу до неї та виконання запитів необхідно проаналізувати ряд систем управління базами даних (СУБД), які надають відповідний функціонал і забезпечують ефективну взаємодію з іншими компонентами інформаційної системи.

У випадку онлайн-системи для продажу товарів висуваються додаткові вимоги:

- підтримка великої кількості транзакцій;
- захист персональних даних;
- обробка платежів;
- можливість масштабування.

У процесі вибору СУБД для онлайн-системи важливо враховувати тип проекту, обсяг даних, вимоги до швидкодії, підтримку транзакцій, наявність технічної підтримки, простоту інтеграції з іншими компонентами (таблиця 2.8).

Таблиця 2.8 – Порівняння найпопулярніших СУБД для web-систем

Параметр	MySQL	PostgreSQL	SQLite	MongoDB
Тип	Реляційна	Реляційна	Локальна	NoSQL (документальна)
Продуктивність	Висока	Висока	Обмежена	Висока
Масштабованість	Висока	Висока	Низька	Дуже висока
Транзакції	Так (InnoDB)	Так	Обмежено	Ні

Продовження таблиці 2.8

Простота	Легка	Середня	Дуже проста	Складна для SQL-зв'язків
Сумісність	PHP, Python	PHP, Python	Усі	Node.js, Python
Ліцензія	GPL	PostgreSQL	Public Domain	SSPL

MySQL пропонує широкий спектр функціональних можливостей, які забезпечують її численні переваги. Ця СУБД вирізняється багатим набором інструментів, що роблять її особливо привабливою для використання при розробці web-систем [10]:

- стабільна та перевірена роками технологія, що підтримує всі типові задачі онлайн-магазину;
- широка підтримка хостингів – більшість провайдерів вже мають встановлений MySQL;
- велика спільнота та документація – спрощує розв'язання технічних проблем;
- підтримка реляційної моделі та транзакцій (через InnoDB);
- гнучка інтеграція з PHP та популярними фреймворками, наприклад, Laravel, Yii, Symfony.

Вибір інструментів для взаємодії із системою управління базами даних MySQL має вирішальне значення для ефективної реалізації веб-системи. До таких засобів належать:

- мови програмування, які забезпечують обробку логіки на серверній частині;
- фреймворки та бібліотеки, які спрощують розробку, структурують код і забезпечують засоби безпеки;

- ORM-системи (Object-Relational Mapping), які полегшують роботу з базою даних через об'єктну модель;
- інструменти адміністрування, які дають змогу керувати БД без глибоких знань SQL.

PHP – найпоширеніша серверна мова для створення веб-систем, яка історично тісно інтегрована з MySQL. Переваги:

- вбудовані функції роботи з MySQL (mysqli, PDO);
- підтримка популярних фреймворків, таких як Laravel, Symfony, WordPress – усі працюють із MySQL;
- легка інтеграція з HTML та JavaScript;
- широка спільнота та багато навчальних ресурсів.

Недоліками є складність у забезпеченні безпеки та масштабованості (MVC, фреймворки).

Python з фреймворком Django – також потужне середовище для швидкої розробки веб-додатків.

Перевагами є висока читабельність коду, вбудований ORM, який легко підключається до MySQL, активна підтримка спільноти. Та при виборі даної мови слід врахувати такі недоліки як менша популярність для хостингу в порівнянні з PHP, більша початкова складність налаштування.

Node.js дозволяє реалізовувати як клієнтську, так і серверну частину за допомогою JavaScript. Для взаємодії з MySQL використовують бібліотеки mysql2, sequelize та фреймворк Express.js для створення REST API.

Перевагами є асинхронність, масштабованість, сучасний стек (MERN/MEVN). Але вагомим недоліком є більша складність обробки помилок та логіки у порівнянні з PHP.

Фреймворки використовують для спрощення, прискорення та структурування процесу розробки програмного забезпечення, зокрема веб-додатків. Вони надають готовий каркас (framework), в межах якого розробник створює власну логіку, не починаючи все "з нуля".

									Арк.
									38
Змн.	Арк.	№ докум.	Підпис	Дата					

Основні цілі використання фреймворків:

- швидкість розробки – фреймворки надають набір готових функцій (робота з базами даних, маршрутизація, автентифікація тощо), що значно скорочує час на написання коду;
- стандартизація коду – вони забезпечують дотримання загальноприйнятих підходів та структур (наприклад, MVC — Model-View-Controller), що робить код більш читабельним і зрозумілим для інших розробників;
- безпека – більшість сучасних фреймворків мають вбудовані засоби захисту від поширених атак: SQL-ін'єкцій, XSS, CSRF тощо;
- підтримка та масштабованість – завдяки своїй структурі, фреймворки дозволяють легше підтримувати та розширювати проект у майбутньому;
- зменшення кількості помилок – використання вже перевірених компонентів знижує ризик виникнення критичних помилок у коді.
- інтеграція з іншими технологіями – багато фреймворків легко інтегруються з бібліотеками, API, сторонніми сервісами, що розширює функціональні можливості проекту.

Laravel є найпопулярнішим фреймворком у світі PHP. Має потужну інтеграцію з MySQL і забезпечує:

- ORM Eloquent для об'єктної роботи з таблицями;
- міграції та seeders для управління схемою БД;
- Blade шаблони для відображення даних;
- готову систему автентифікації та авторизації;
- підтримку кешування, черг, веб-сокетів.

Недоліки: потребує певного рівня підготовки, особливо у роботі з міграціями та сервісами.

Фреймворк Django дозволяє швидко створювати надійні веб-додатки з підтримкою MySQL. Основні функції:

- ORM для моделювання БД через класи Python;

					КР.КН 25.593.10.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

- адмін-панель за замовчуванням;
- вбудовані засоби безпеки (XSS, CSRF, SQL-захист).

Але важливим недоліком є складність налаштування проекту на продакшн-сервері.

Express.js – мінімалістичний фреймворк для Node.js, який забезпечує створення API для фронтенду. Використовується з MySQL через mysql2 (низькорівнева бібліотека для виконання SQL-запитів) та sequelize (ORM-аналог Eloquent). Надає такі переваги як асинхронна обробка, висока продуктивність; при цьому недоліками є велика кількість налаштувань, відсутність "з коробки" рішень для авторизації, валідації.

Інструменти адміністрування MySQL використовуються для керування, налаштування, моніторингу та оптимізації роботи бази даних без необхідності постійно писати SQL-запити вручну. Вони значно полегшують життя як розробникам, так і адміністраторам баз даних (DBA):

- створення та редагування баз даних і таблиць: візуальний інтерфейс дозволяє створювати таблиці, визначати поля, типи даних, індекси без написання SQL-коду;
- виконання SQL-запитів: інтерфейс для ручного введення, збереження та виконання SQL-запитів із результатами у зручній формі;
- імпорт та експорт даних: зручно переносити дані між різними серверами або робити резервні копії (наприклад, у форматі SQL або CSV);
- керування користувачами та правами доступу: додавання нових користувачів, встановлення паролів, обмеження доступу до окремих баз або таблиць;
- моніторинг продуктивності: аналіз навантаження, кількості підключень, блокувань, часу виконання запитів тощо;
- резервне копіювання та відновлення: планування та створення бекапів бази даних, а також швидке відновлення в разі збою;

					КР.КН 25.593.10.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

– візуальне моделювання бази даних: створення ER-діаграм (Entity-Relationship), що особливо корисно при проектуванні структури БД.

Найпоширенішими інструментами адміністрування MySQL на сьогоднішній день є phpMyAdmin та MySQL Workbench.

PhpMyAdmin надає безкоштовний веб-інтерфейс, забезпечує роботу з таблицями, SQL-запитами, імпорт/експорт, також тут можна швидко створити або редагувати схему бази.

Щодо MySQL Workbench, то він є більш потужним, ніж phpMyAdmin, також підтримує візуальне моделювання БД (ER-діаграми) та автоматизацію резервного копіювання та адміністрування.

Ручна розробка інформаційної системи передбачає створення функціоналу «з нуля» із використанням базових веб-технологій — таких як HTML, CSS, JavaScript, PHP та MySQL. На відміну від розробки за допомогою фреймворків або CMS, цей підхід має низку суттєвих переваг, які особливо важливі у навчальному та індивідуальному проектах:

1) Глибоке розуміння роботи системи – розробник глибше засвоює логіку веб-програмування, самостійно реалізує механізми авторизації, роботи з базами даних, обробки форм, сесій та запитів. Це сприяє кращому розумінню архітектури веб-застосунків та алгоритмів взаємодії клієнт-сервер.

2) Повний контроль над кодом – тут відсутні приховані частини коду, що дозволяє повністю контролювати логіку, безпеку та оптимізацію. Це особливо важливо, коли потрібне точне налаштування функціоналу під потреби користувача чи бізнес-логіку.

3) Гнучкість у реалізації нестандартних рішень – фреймворки часто нав'язують власну структуру проекту або обмежують у виборі рішень. Ручна розробка дозволяє реалізувати будь-який нестандартний функціонал – наприклад, систему ролей (клієнт, адміністратор, комплектувальник), індивідуальні правила фільтрації товарів, кастомну логіку замовлень тощо.

					КР.КН 25.593.10.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Опис технологій та засобів реалізації

При розробці інформаційної системи для онлайн-продажу манікюрних товарів було обрано перевірені та широко розповсюджені веб-технології, що забезпечують оптимальне співвідношення функціональності, продуктивності та економічної ефективності. Вибір технологічного стеку базувався на принципах масштабованості, простоти підтримки та мінімізації витрат на ліцензування програмного забезпечення.

Основою серверної логіки обрано мову програмування PHP, що виправдано кількома ключовими факторами [9]. По-перше, PHP підтримується практично всіма веб-хостингами, що значно спрощує розгортання проекту як на локальному середовищі, так і на віддалених серверах. По-друге, наявність великої спільноти розробників забезпечує доступ до величезної кількості документації, навчальних матеріалів та готових рішень. PHP створено спеціально для веб-розробки, що робить його ідеальним інструментом для e-commerce проектів, а відсутність ліцензійних витрат забезпечує економічну ефективність рішення [11].

Для зберігання даних використовується реляційна база даних MySQL, вибір якої обґрунтовується її надійністю та стабільністю у сфері електронної комерції. MySQL забезпечує ACID-сумісність, що гарантує цілісність даних при проведенні фінансових операцій та обробці замовлень. Система володіє відмінними показниками масштабованості, дозволяючи обробляти великі обсяги даних та витримувати високе навантаження. Community Edition надає всі необхідні функції без додаткових витрат на ліцензування.

Керування базою даних здійснюється через phpMyAdmin, що надає зручний веб-інтерфейс для адміністрування, моніторингу продуктивності та створення резервних копій. Цей інструмент дозволяє ефективно керувати структурою бази даних, оптимізувати запити та контролювати продуктивність системи.

					КР.КН 25.593.10.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

Для створення адаптивного та сучасного інтерфейсу використано CSS-фреймворк Bootstrap 5, що забезпечує мобільну адаптивність через responsive дизайн, який працює однаково добре на всіх типах пристроїв від смартфонів до великих моніторів. Фреймворк забезпечує розробку візуального інтерфейсу завдяки єдиному стилю компонентів по всьому додатку, а готові компоненти та утиліти значно прискорюють цей процес. Bootstrap 5 забезпечує сумісність з усіма сучасними браузерами.

Динамічна функціональність реалізована за допомогою JavaScript з використанням Fetch API, що дозволило уникнути залежностей від сторонніх JavaScript-бібліотек та зменшити розмір сторінок.

Для організації доставки товарів інтегровано API Нової Пошти через спеціально розроблені проксі-скрипти на PHP, що використовують cURL для здійснення HTTP-запитів [1]. Такий підхід забезпечує високий рівень безпеки через приховування API-ключів від клієнтської частини, дозволяє реалізувати кешування відповідей для покращення продуктивності системи, забезпечує централізовану обробку помилок API.

Для розробки проєкту було обрано Visual Studio Code як основне інтегроване середовище розробки, що виправдано його потужними функціональними можливостями та гнучкістю налаштування. IDE забезпечує зручне підсвічування синтаксису для PHP, JavaScript, HTML, CSS та SQL, містить вбудований термінал для виконання команд Git, composer та інших утиліт командного рядка. Інтегрована підтримка Git надає візуальний інтерфейс для commit, merge та branch операцій, а потужна система автодоповнення та IntelliSense працює для всіх використовуваних у проєкті мов програмування.

У процесі розробки активно використовувались такі розширення як PHP IntelliSense для розширеного автодоповнення PHP-коду, MySQL для підключення та виконання SQL-запитів безпосередньо з IDE, Bootstrap розширення для автодоповнення класів фреймворка, Live Server для створення локального сервера розробки з автоматичним перезавантаженням сторінок [7, 8, 10].

					КР.КН 25.593.10.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Для локальної розробки використовується XAMPP, що включає Apache 2.4 як веб-сервер з підтримкою створення SEO-friendly URL, MySQL 8.0 з тими ж налаштуваннями, що й на продакшн-сервері, PHP з усіма необхідними розширеннями, а також phpMyAdmin для локального управління базою даних [3, 4].

Git використовується як основна система контролю версій з чітко організованою гілковою структурою. У проекті дотримуються стандарти для створення структурованих та інформативних повідомлень комітів, використовується автоматична перевірка синтаксису та форматування коду.

Такий комплексний підхід до вибору технологій, організації середовища розробки та структурування коду забезпечує створення надійної, безпечної та масштабованої онлайн-системи для продажу товарів. Використання Visual Studio Code як основного інструменту розробки, поєднане з потужним технологічним стеком PHP, MySQL, Bootstrap та JavaScript, створює оптимальні умови для ефективної розробки, тестування та подальшої підтримки системи [5]. Система здатна ефективно обслуговувати потреби інтернет-магазину манікюрних товарів та легко адаптуватися до майбутніх вимог бізнесу, забезпечуючи стабільну роботу та можливість для росту.

3.2 Реалізація бази даних

Реалізація бази даних для інформаційної системи для онлайн-продажу манікюрних товарів виконується з використанням системи керування базами даних MySQL через інструмент phpMyAdmin. Процес створення бази даних розпочався з виконання команди CREATE DATABASE, після чого структура даних формувалася поетапно через створення усіх необхідних таблиць системи.

У результаті проектування було створено комплексну базу даних, що включає наступні основні таблиці:

- users - інформація про користувачів системи (клієнти, пакувальники, адміністратори);

					КР.КН 25.593.10.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

- categories - категорії товарів для структурування асортименту;
- products - каталог товарів з повною інформацією про кожну позицію;
- cart_items - елементи кошику покупок користувачів;
- orders - замовлення клієнтів з усіма деталями;
- order_items - позиції в замовленнях з кількістю та ціною;
- wishlist - список бажань користувачів для збереження улюблених товарів.

Таблиці формувалися за допомогою запитів типу CREATE TABLE, у яких детально визначалися імена полів, типи даних (зокрема INT, VARCHAR, TEXT, DECIMAL, ENUM, DATETIME), первинні ключі (PRIMARY KEY), зовнішні ключі (FOREIGN KEY), автоматичне збільшення значень полів (AUTO_INCREMENT) та обмеження на унікальність значень (UNIQUE). Всі таблиці налаштовувалися з використанням движка InnoDB та кодування utf8mb4_unicode_ci для забезпечення коректної роботи з українським текстом.

Приклад створення таблиці користувачів:

```
CREATE TABLE users (
  id INT PRIMARY KEY AUTO_INCREMENT,
  email VARCHAR(255) UNIQUE NOT NULL,
  password_hash VARCHAR(255) NOT NULL,
  name VARCHAR(100),
  role ENUM('client', 'packer', 'admin') DEFAULT 'client',
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP);
```

Результат запиту створення таблиці користувачів зображено на рисунку 3.1.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1 id	int(11)			Hi	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2 email	varchar(255)	utf8mb4_unicode_ci		Hi	Немає		
<input type="checkbox"/>	3 password_hash	varchar(255)	utf8mb4_unicode_ci		Hi	Немає		
<input type="checkbox"/>	4 name	varchar(100)	utf8mb4_unicode_ci		Так	NULL		
<input type="checkbox"/>	5 role	enum('buyer', 'admin', 'packer')	utf8mb4_unicode_ci		Так	buyer		
<input type="checkbox"/>	6 created_at	timestamp			Hi	current_timestamp()		

Рисунок 3.1 – Створена таблиця користувачів за допомогою запити

Приклад створення таблиці товарів:

```
CREATE TABLE products (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    description TEXT,  
    price DECIMAL(10,2) NOT NULL,  
    image_url VARCHAR(255),  
    category_id INT,  
    FOREIGN KEY (category_id) REFERENCES categories(id)  
);
```

Результат запиту створення таблиці товарів зображено на рисунку 3.2.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1 id 🔑	int(11)			Ні	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає		
<input type="checkbox"/>	3 description	text	utf8mb4_unicode_ci		Так	NULL		
<input type="checkbox"/>	4 brand	varchar(100)	utf8mb4_unicode_ci		Так	NULL		
<input type="checkbox"/>	5 color	varchar(50)	utf8mb4_unicode_ci		Так	NULL		
<input type="checkbox"/>	6 price	decimal(10,2)			Ні	Немає		
<input type="checkbox"/>	7 image_url	varchar(255)	utf8mb4_unicode_ci		Так	NULL		
<input type="checkbox"/>	8 stock	int(11)			Так	0		
<input type="checkbox"/>	9 category_id 🔑	int(11)			Так	NULL		

Рисунок 3.2 – Створена таблиця товарів за допомогою запиту

Аналогічно створювалися запити для решти таблиць системи, з урахуванням специфіки кожної сутності та встановленням необхідних зв'язків між таблицями через зовнішні ключі. Усі запити для створення таблиць бази даних розміщено у додатку А.

Після створення структури таблиць здійснюється первинне наповнення бази даних через SQL-запити типу INSERT INTO, які дозволяють заповнювати таблиці початковими даними для тестування і демонстрації роботи системи.

Приклад заповнення таблиці товарів:

```
INSERT INTO products (name, description, price, image_url,  
category_id) VALUES
```

```
( 'Гель-лак XYZ', 'Гель-лак високої якості з насиченим кольором та стійким покриттям.', 150.00, '/assets/images/gel_xyz.jpg', 1),
( 'Манікюрні ножиці ABC', 'Професійні манікюрні ножиці з гострими лезами.', 85.50, '/assets/images/scissors_abc.jpg', 2),
( 'Пилка для нігтів DEF', 'Двостороння пилка з абразивом середньої жорсткості.', 35.00, '/assets/images/file_def.jpg', 3);
```

Результат виконання запиту зображено на рисунку 3.3.

id	name	description	brand	color	price	image_url	stock	category_id
1	Гель-лак №101	Стійкий гель-лак для манікюру	Kodi	Рожевий	150.00	/nail/assets/6.jpg	10	1
2	Кусачки	Кусачки для манікюру та педикюру	Невідомий	Срібний	350.00	/nail/assets/1.jpg	30	2
3	Камінці для дизайну	Набір блискіток для нігтів	Невідомий	Асорті	50.00	/nail/assets/7.jpg	25	3

Рисунок 3.3 – Заповнення таблиці товарів

Подібним чином заповнювалися всі інші таблиці системи відповідними тестовими даними, що дозволило створити повноцінну демонстраційну версію онлайн-системи. Запити для заповнення таблиць подано у додатку Б.

Використання описаного підходу дозволяє створити надійну та масштабовану базу даних, яка забезпечує ефективне зберігання та управління всією необхідною інформацією для функціонування онлайн-системи для продажу манікюрних товарів. Структура бази даних готова для подальшого розвитку та підтримки роботи системи, а також може бути легко адаптована під нові вимоги бізнесу.

3.3 Реалізація основних функцій інформаційної системи

В рамках розробки системи було створено комплексний механізм аутентифікації та реєстрації користувачів. Процес реєстрації включає валідацію унікальності email-адреси та безпечне зберігання паролів у вигляді хеш-значень за допомогою функції `password_hash()`.

Система входу базується на серверній валідації даних через захищені SQL-запити.

При успішній аутентифікації в PHP-сесії зберігаються ідентифікатор користувача та його роль, що забезпечує контроль доступу до захищених ресурсів.

Програмний код для реалізації логіки входу:

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['login'])) {
    $stmt = $conn->prepare("SELECT id,password_hash,role FROM
users WHERE email=?");
    $stmt->bind_param("s",$_POST['email']);
    $stmt->execute();
    $u = $stmt->get_result()->fetch_assoc();
    if ($u &&
password_verify($_POST['password'],$u['password_hash'])) {
        $_SESSION['user_id']=$u['id'];
        $_SESSION['role']=$u['role'];
        header("Location: profile_page.php");
        exit();
    } else {
        $login_error = "Невірний email або пароль.";
    }
}
// profile.php - обробка реєстрації
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['register'])) {
    // ... INSERT INTO
users(email,password_hash,name,role='buyer')
}
```

Розмежування між операціями входу та реєстрації здійснюється через перевірку відповідних ключів у масиві \$_POST. Функція password_verify() забезпечує порівняння введеного пароля з хешованим значенням у базі даних.

При успішній аутентифікації виконується редирект на профільну сторінку з обов'язковим викликом exit() для запобігання подальшому виконанню коду. У випадку невдалої спроби входу формується відповідне повідомлення про помилку.

Блок реєстрації працює за аналогічним принципом, спочатку перевіряється унікальність email-адреси, після чого пароль хешується та зберігається в базі даних. Після успішної реєстрації користувач може бути перенаправлений на сторінку входу (рис. 3.4) або отримати підтвердження про створення облікового запису.

									Арк.
									49
Змн.	Арк.	№ докум.	Підпис	Дата					

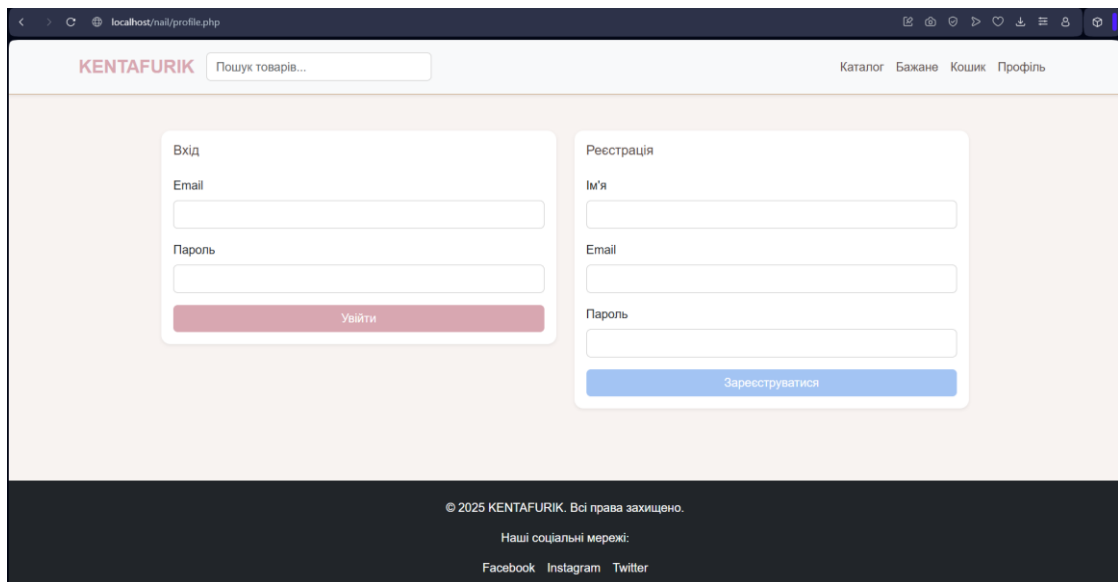


Рисунок 3.4 – Сторінка входу та реєстрації

Така архітектура забезпечує надійний захист від основних загроз безпеки, включаючи SQL-ін'єкції та компрометацію паролів, використовуючи стандартні PHP-механізми сесій, підготовлені SQL-запити та вбудовані функції хешування. Повний лістинг програмного коду сторінки розміщено у додатку В.

Користувач має доступ до перегляду каталогу товарів, де асортимент виводиться у вигляді карток з назвою, фото та ціною.

Система каталогу товарів реалізована з використанням ефективних SQL-запитів для сортування та фільтрації даних. Застосування конструкцій ORDER BY та WHERE забезпечує швидку обробку користувацьких запитів щодо вибору категорій та параметрів сортування. Детальна інформація про кожен товар відображається на окремій сторінці через файл product.php, який витягує з бази даних повні відомості про обраний продукт за унікальним ідентифікатором.

Відображення списку товарів здійснюється шляхом ітерації через результат SQL-запиту SELECT * FROM products. Кожен виклик методу \$result->fetch_assoc() повертає асоціативний масив з даними товару, де ключі відповідають назвам колонок бази даних (id, name, price, image_url). Цикл автоматично завершується при досягненні кінця результуючого набору.

									Арк.
									50
Змн.	Арк.	№ докум.	Підпис	Дата	КР.КН 25.593.10.000 ПЗ				

Для візуального представлення товарів використовується компонент Bootstrap Card з класом h-100, що забезпечує однакову висоту всіх карток у ряду незалежно від обсягу контенту. Зображення товарів інтегровані через тег з класом card-img-top, що автоматично адаптує їх під ширину картки (рис. 3.5).

Ось приклад реалізації відображення товарів:

```
<?php while($row = $result->fetch_assoc()): ?>
    <div class="card h-100">
        
        <div class="card-body d-flex flex-column">
            <h5><?= $row['name'] ?></h5>
            <p><?= number_format($row['price'],2) ?> грн</p>
            <a href="product.php?id=<?= $row['id'] ?>" class="btn btn-
primary mb-2">Детальніше</a>
            <a href="cart.php?add=<?= $row['id'] ?>" class="btn btn-
success mb-2">В кошик</a>
            <a href="products.php?add_wish=<?= $row['id'] ?>" class="btn
btn-outline-primary mt-auto">В бажане</a>
        </div>
    </div>
<?php endwhile; ?>
```

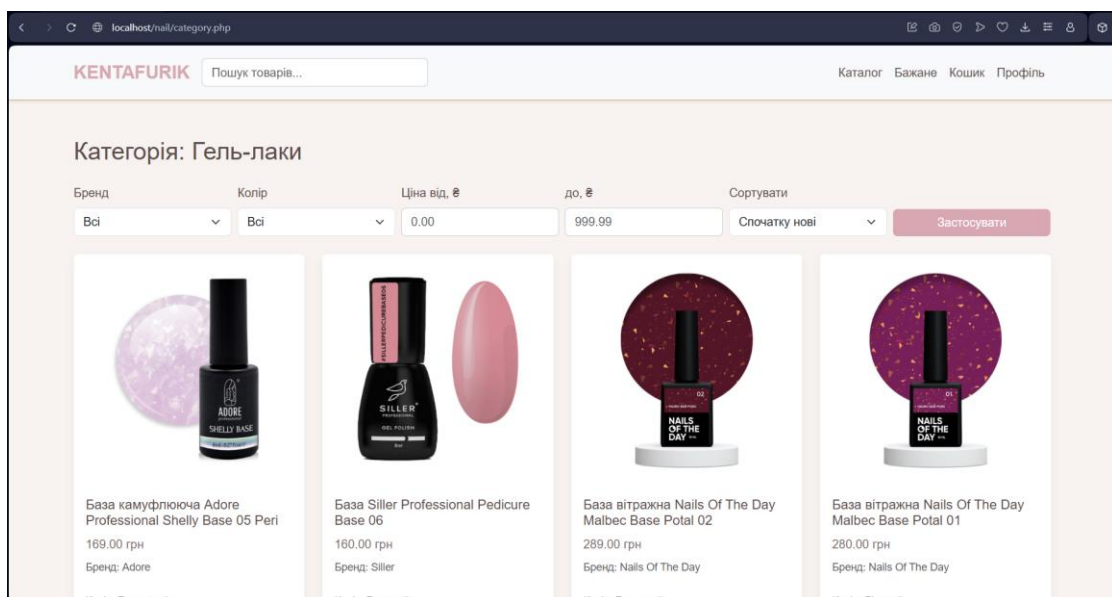


Рисунок 3.5 – Сторінка товарів обраної категорії

Функціональність карток включає три основні дії, це перегляд детальної інформації через посилання на `product.php?id=...`, додавання товару до кошика через `cart.php?add=ID`, та додавання до списку бажаного. GET-параметри в URL забезпечують передачу ідентифікатора товару для подальшої обробки відповідними скриптами.

Адаптивність інтерфейсу досягається завдяки Bootstrap-класам та Flexbox-верстці, що автоматично підлаштовує елементи під різні розміри екранів. Flex-контейнер в body картки (рис.3. 6).

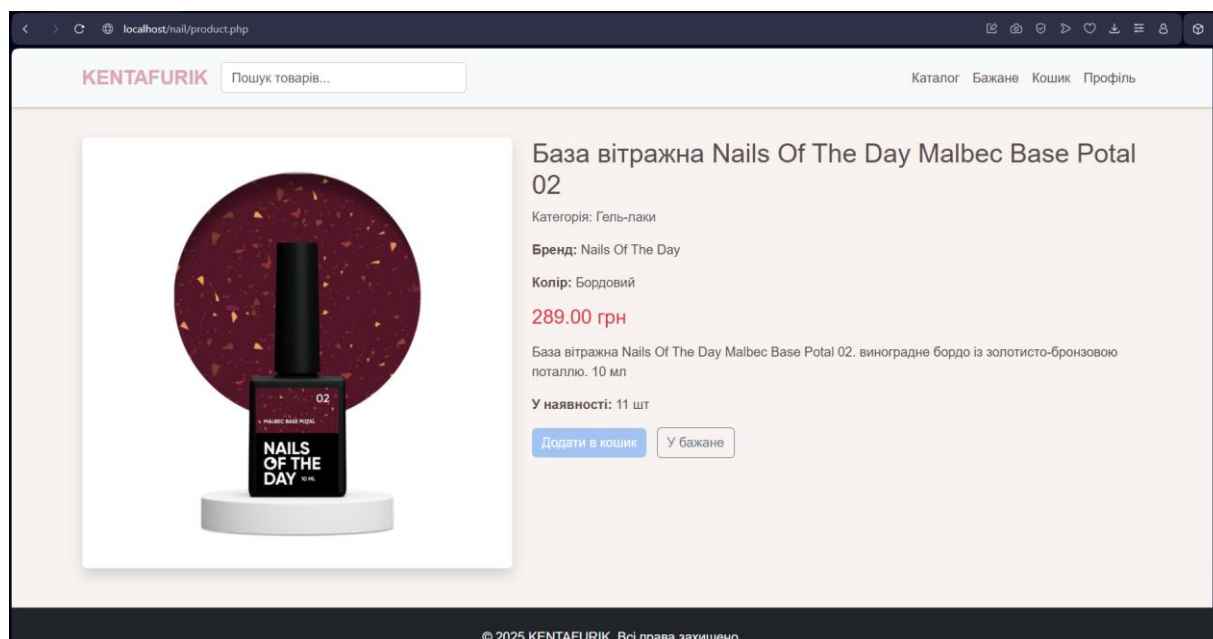


Рисунок 3.6 – Сторінка товарів обраної категорії

Таким чином забезпечує оптимальне розташування кнопок у нижній частині незалежно від довжини опису товару, створюючи візуально привабливий та функціональний інтерфейс. Повний лістинг програмного коду сторінки розміщено у додатку В.

Система кошика розроблена на основі сесійного масиву `$_SESSION['cart']`, який зберігає співвідношення ідентифікаторів товарів до їх кількості. Механізм додавання товарів реалізований через GET-параметри, що забезпечує швидке оновлення вмісту кошика з будь-якої сторінки сайту.

					КР.КН 25.593.10.000 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Основна логіка обробки зосереджена у файлі `cart.php`, який відповідає за відображення поточного стану кошика та управління його вмістом.

Структура даних кошика базується на асоціативному масиві, де ключами є ідентифікатори продуктів, а значеннями – їх кількість:

```
// Ідентифікатор товару => кількість
$_SESSION['cart'] = [
    12 => 2, // продукт #12 – 2 одиниці
    45 => 1, // продукт #45 – 1 одиниця
    // ...
];
```

Інтерфейс додавання товарів реалізований через посилання з GET-параметрами на всіх сторінках каталогу:

```
<a href="cart.php?add=<?=$product_id ?>" class="btn btn-success">
    В кошик
</a>
```

Обробка додавання товарів відбувається на початку `cart.php`:

```
if (isset($_GET['add']) && is_numeric($_GET['add'])) {
    $pid = (int)$_GET['add'];
    // Якщо в сесії вже є цей товар – просто збільшуємо лічильник
    if (isset($_SESSION['cart'][$pid])) {
        $_SESSION['cart'][$pid]++;
    } else {
        // Інакше – додаємо новий запис із кількістю 1
        $_SESSION['cart'][$pid] = 1;
    }
    // Після цього робимо редирект назад на кошик,
    // щоб уникнути повторного додавання при оновленні сторінки.
    header("Location: cart.php");
    exit();
}
```

Цей підхід забезпечує поступове збільшення кількості для існуючих товарів або створення нового запису для нових позицій. Результат подано на рисунку (рис. 3.7).

					КР.КН 25.593.10.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

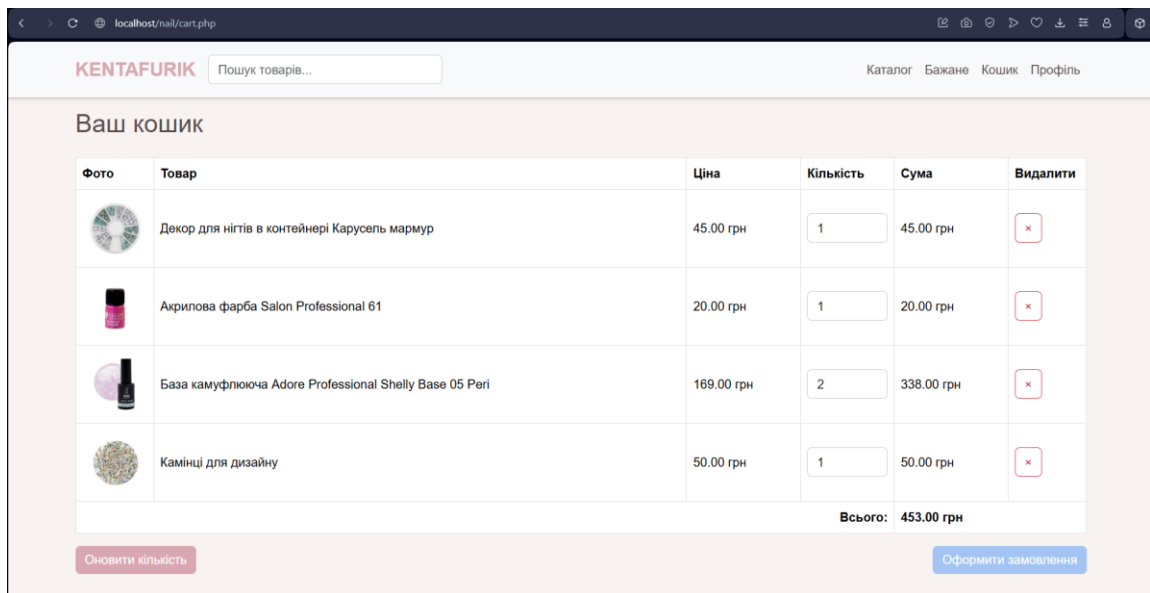


Рисунок 3.7 – Кошик з товарами

Функціонал оновлення кількості реалізований через POST-форму з масивом полів quantities:

```

if (isset($_POST['update_cart'])) {
    foreach ($_POST['quantities'] as $pid => $qty) {
        $pid = (int)$pid;
        $qty = max(1, (int)$qty); // мінімум 1
        $_SESSION['cart'][$pid] = $qty;
    }
    header("Location: cart.php");
    exit();}

```

Оптимізація роботи з базою даних досягається завдяки одноразовому запиту всіх необхідних товарів через конструкцію IN:

```

$sids = array_keys($_SESSION['cart']);
$placeholders = implode(',', array_fill(0, count($sids), '?'));
$stmt = $conn->prepare(
    "SELECT id,name,price,image_url FROM products WHERE id IN
($placeholders) " );
$stmt->bind_param(str_repeat('i', count($sids)), ...$sids);
$stmt->execute();
$result = $stmt->get_result();

```

Такий підхід мінімізує кількість звернень до бази даних до одного запиту замість множинних запитів для кожного товару окремо, що суттєво покращує продуктивність системи. Додатково передбачена можливість повного видалення позицій з кошика та розрахунку загальної вартості замовлення з урахуванням кількості кожного товару. Повний лістинг програмного коду сторінки розміщено у додатку Г.

Модуль списку бажаного реалізований через окрему таблицю wishlist з прив'язкою до ідентифікатора користувача, що забезпечує персоналізоване зберігання обраних товарів.

Основна функціональність включає додавання та видалення позицій через GET-параметри з додатковими повідомленнями про статус операцій у вигляді автоматично зникаючих повідомлень.

Логіка видалення товарів з списку бажаного обробляється на початку файлу wishlist.php:

```
if (isset($_GET['remove']) && is_numeric($_GET['remove'])) {
    $wid = (int)$_GET['remove'];
    $stmt = $conn->prepare("DELETE FROM wishlist WHERE id = ? AND
user_id = ?");
    $stmt->bind_param("ii", $wid, $user_id);
    $stmt->execute();
    // Після видалення перенаправляємо назад, щоб очистити
параметр remove і оновити список
    header("Location:  wishlist.php?msg=" . urlencode("Товар
видалено"));
    exit();
}
```

Цей підхід гарантує безпечне видалення записів через валідацію параметрів та підготовлені SQL-вирази, що запобігає несанкціонованому доступу до чужих даних. Умова WHERE id = ? AND user_id = ? забезпечує подвійну перевірку прав доступу.

					КР.КН 25.593.10.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

Відображення списку бажаного здійснюється через оптимізований JOIN-запит, який об'єднує дані з таблиць wishlist та products:

```
$stmt = $conn->prepare("
    SELECT w.id as wid, w.product_id as pid, p.name, p.price,
    p.image_url
    FROM wishlist w
    JOIN products p ON w.product_id = p.id
    WHERE w.user_id = ?
");
$stmt->bind_param("i", $user_id);
$stmt->execute();
$wish = $stmt->get_result();
```

Результат подано на рисунку 3.8.

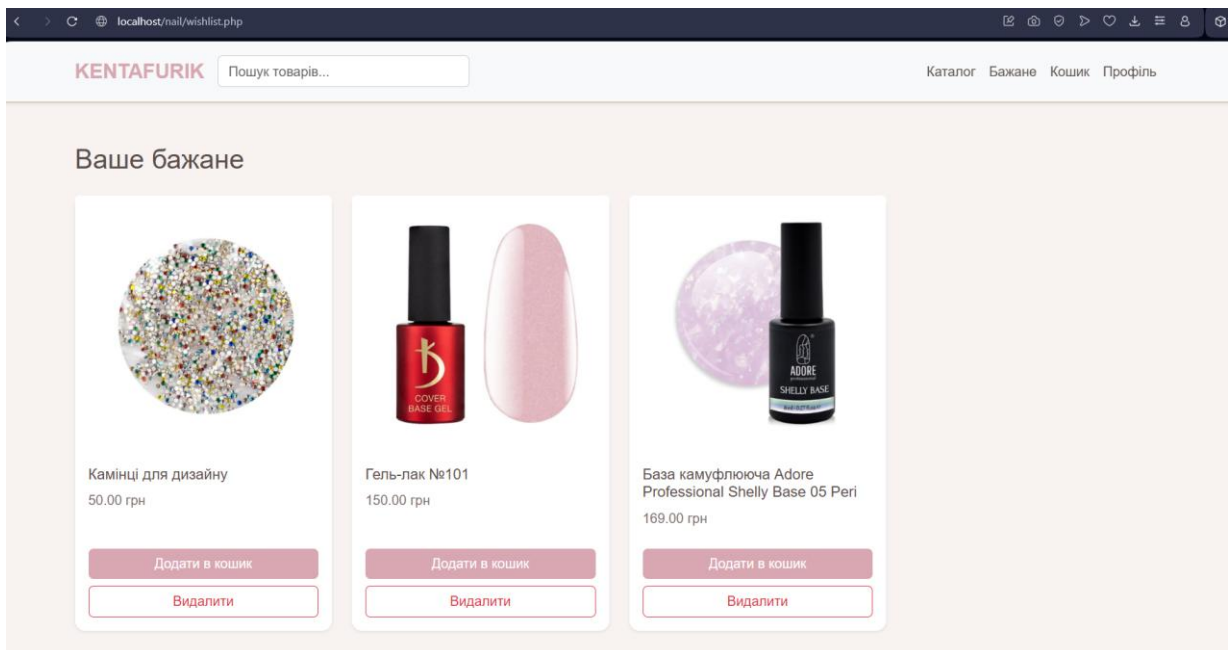


Рисунок 3.8 – Список бажаного

Такий запит мінімізує кількість звернень до бази даних, отримуючи всю необхідну інформацію про товари та їх прив'язку до списку бажаного за одну операцію.

Фільтрація за user_id гарантує конфіденційність даних користувачів.

					КР.КН 25.593.10.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтерфейс списку бажаного побудований на Bootstrap-компонентах з інтегрованими функціями управління:

```
<?php while ($item = $wish->fetch_assoc()): ?>
    <div class="card h-100">
        
        <div class="card-body d-flex flex-column">
            <h5 class="card-title"><?= htmlspecialchars($item['name'])
?></h5>
            <p class="card-text"><?= number_format($item['price'],2) ?>
грн</p>
            <a href="cart.php?add=<?= $item['pid'] ?>" class="btn btn-
primary mb-2">В кошик</a>
            <a href="wishlist.php?remove=<?= $item['wid'] ?>"
class="btn btn-outline-danger mt-auto"
onclick="return confirm('Видалити з бажаного?')">
                Видалити
            </a> </div> </div> <?php endwhile; ?>
```

Кожна картка товару містить можливості переміщення до кошика та видалення з бажаного (рис. 3. 9).

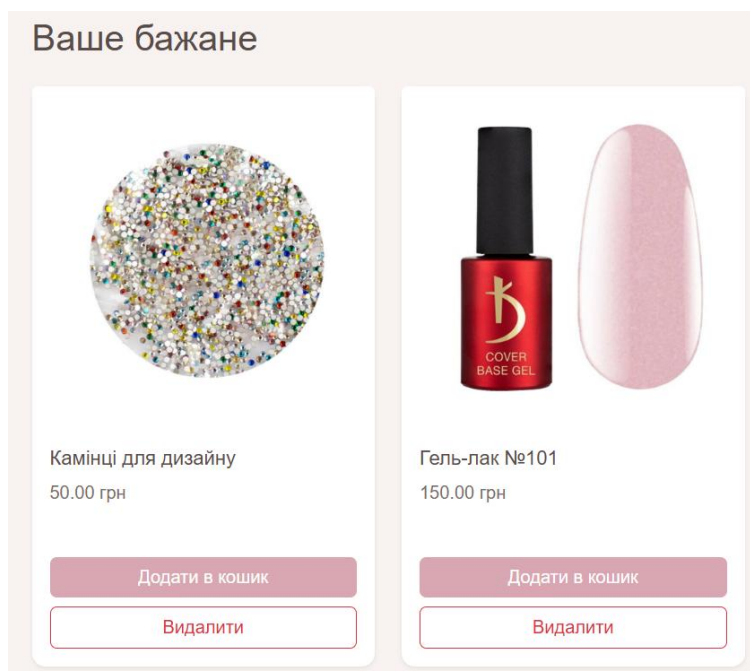


Рисунок 3. 9 – Сторінка з переліком бажаних товарів

					КР.КН 25.593.10.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

JavaScript-підтвердження через `confirm()` запобігає випадковому видаленню позицій.

Система повідомлень реалізована через URL-параметри з автоматичним видаленням алертів:

```
<?php if (isset($_GET['msg'])): ?>
    <div id="alert" class="alert alert-success">
        <?= htmlspecialchars($_GET['msg']) ?>
    </div>
    <script>
        setTimeout(() => document.getElementById('alert').remove(),
3000);
    </script>
<?php endif; ?>
```

Ця реалізація забезпечує повнофункціональний модуль списку бажаного з мінімальним навантаженням на базу даних, захищеними операціями та зручним користувацьким інтерфейсом.

Всі дії супроводжуються відповідним фідбеком через тимчасові повідомлення, що покращує загальний користувацький досвід. Повний лістинг програмного коду сторінки розміщено у додатку Д.

Система оформлення замовлень реалізована через модуль `checkout.php`, який забезпечує комплексну обробку товарів з кошика за допомогою транзакційного підходу.

Процес включає створення основного запису замовлення, додавання деталізованих позицій та очищення кошика користувача. Інтерфейс надає можливості вибору способу доставки та методу оплати з динамічною адаптацією полів введення (рис. 3.10).

					КР.КН 25.593.10.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

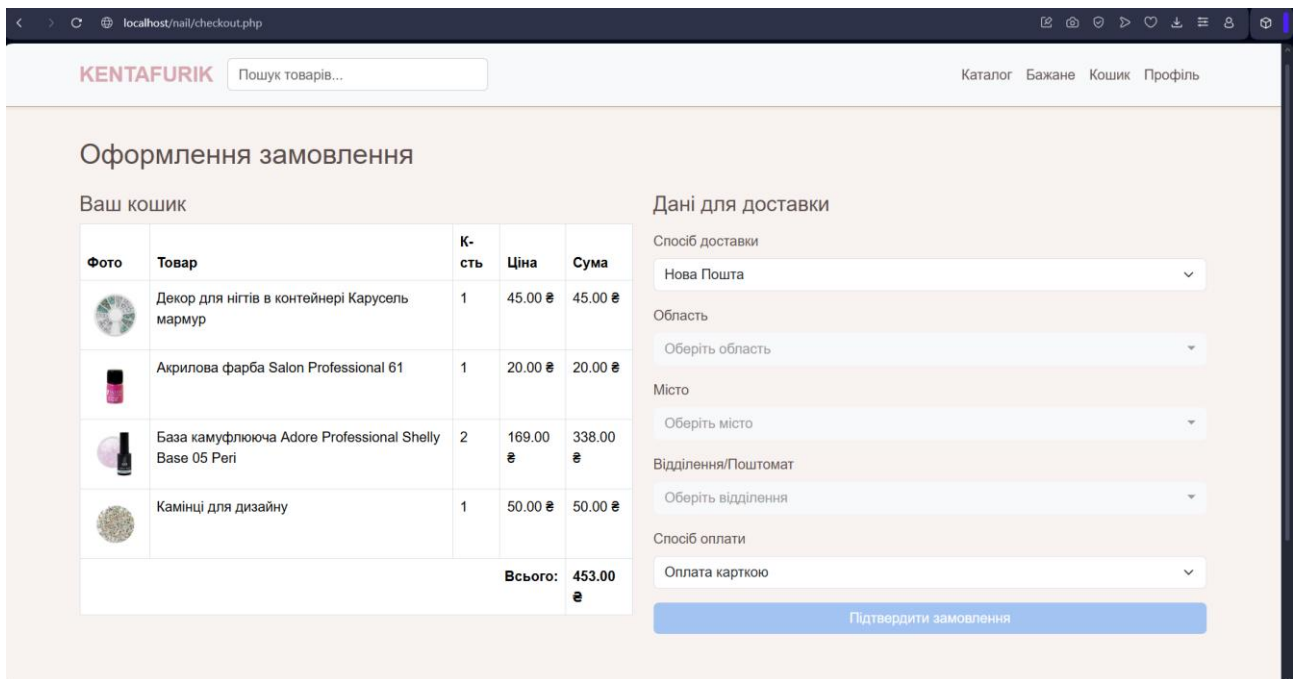


Рисунок 3.10 – Сторінка оформлення замовлення

Початковий етап обробки включає отримання актуальної інформації про товари з розрахунком загальної вартості:

```

$total = 0;
$items = [];
while ($r = $cart->fetch_assoc()) {
    $line = $r['price'] * $r['quantity'];
    $total += $line;
    $items[] = [
        'product_id' => $r['product_id'],
        'quantity' => $r['quantity'],
        'price' => $r['price']
    ];
}

```

Такий підхід гарантує точність розрахунків на основі поточних цін з бази даних, виключаючи можливості маніпуляцій з клієнтської сторони.

					КР.КН 25.593.10.000 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

Користувацький інтерфейс включає адаптивні елементи для вибору параметрів доставки:

```
<select id="delivery_method" name="delivery_method"
class="form-select" required>
  <option value="Нова Пошта">Нова Пошта</option>
  <option value="УкрПошта">УкрПошта</option>
  <option value="Самовивіз">Самовивіз</option>
</select>

<div id="np-fields" style="display:none;">
  <!-- динамічні селекти для НП -->
</div>

<div id="text-address" class="mb-3">
  <textarea name="delivery_address_text" class="form-control"
    rows="3" placeholder="Введіть адресу" required></textarea>
</div>

<select name="payment_method" class="form-select" required>
  <option value="Готівка">Готівка</option>
  <option value="Карта">Карта онлайн</option>
  <option value="Приват24">Приват24</option>
</select>
```

JavaScript-обробка події change керує відображенням відповідних блоків інтерфейсу залежно від обраного способу доставки, динамічно показуючи або приховуючи поля для вибору відділення або введення текстової адреси.

Критичною особливістю системи є використання транзакцій для забезпечення цілісності даних:

```
$conn->begin_transaction();
try {
  // 3.1 Вставка в orders
  $insOrder = $conn->prepare("
```

					КР.КН 25.593.10.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

INSERT INTO orders
    (user_id, status, total, delivery_method, delivery_address,
payment_method)
    VALUES (?, 'new', ?, ?, ?, ?) ";
$insOrder->bind_param("idsss",
    $user_id,
    $total,
    $delivery_method,
    $delivery_address,
    $payment_method );
$insOrder->execute();
$order_id = $conn->insert_id;
// 3.2 Вставка рядків у order_items
$insItem = $conn->prepare("    INSERT INTO order_items
    (order_id, product_id, quantity, price)
    VALUES (?, ?, ?, ?) ");
foreach ($items as $it) {
    $insItem->bind_param("iiid",
        $order_id,
        $it['product_id'],
        $it['quantity'],
        $it['price'] );
    $insItem->execute(); }
// 3.3 Очищення кошика
$delCart = $conn->prepare("DELETE FROM cart_items WHERE
user_id = ?");
$delCart->bind_param("i", $user_id);
$delCart->execute();
$conn->commit();
header("Location: order_confirm.php?id=$order_id"); exit;
} catch (\Exception $e) { $conn->rollback();
// Виведіть користувачу помилку оформлення
echo "<div class='alert alert-danger'>Помилка оформлення
замовлення, спробуйте пізніше.</div>";}

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

Використання транзакцій гарантує, що замовлення не залишиться в неконсистентному стані: або воно цілком записане разом зі своїми рядками, або жоден із кроків не буде застосований.

Транзакційний механізм забезпечує атомарність операцій: у випадку помилки на будь-якому етапі всі зміни відкочуються, запобігаючи створенню неконсистентних записів. Успішне завершення транзакції супроводжується перенаправленням на сторінку підтвердження з ідентифікатором створеного замовлення.

Така архітектура гарантує надійність процесу оформлення та коректне ведення обліку замовлень в системі. Повний лістинг програмного коду сторінки розміщено у додатку Е.

Адміністративна панель системи (рис.3.11) реалізована як універсальний інтерфейс управління даними через єдиний файл `admin_dashboard.php`.

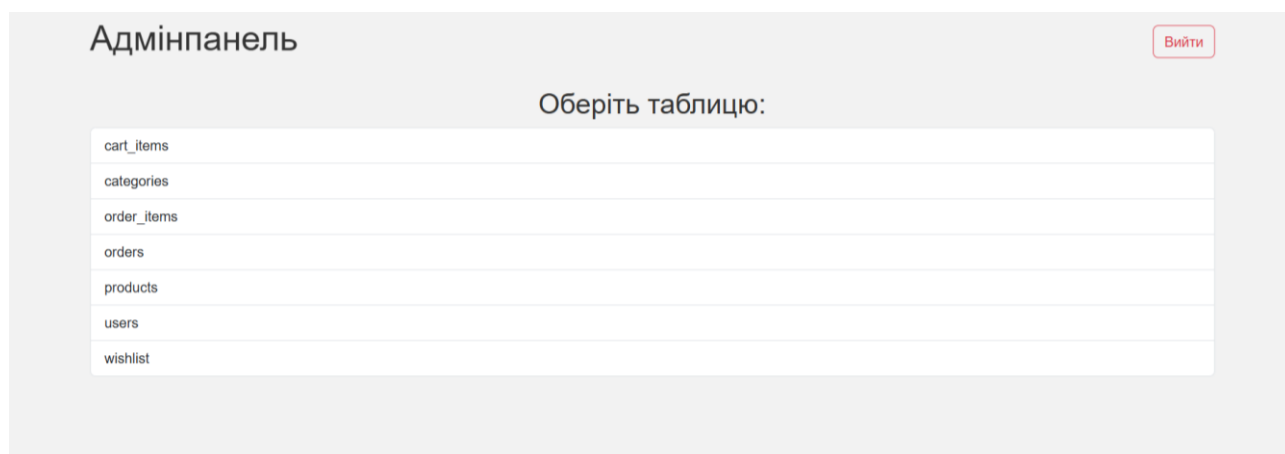


Рисунок 3.11 – Сторінка адміністратора з вибором таблиці

Модуль забезпечує динамічну роботу з будь-якими таблицями бази даних без необхідності створення окремого коду для кожної сутності, підтримуючи повний цикл CRUD-операцій з автоматичним генеруванням форм та спеціалізованим відображенням медіафайлів (рис. 3.12).

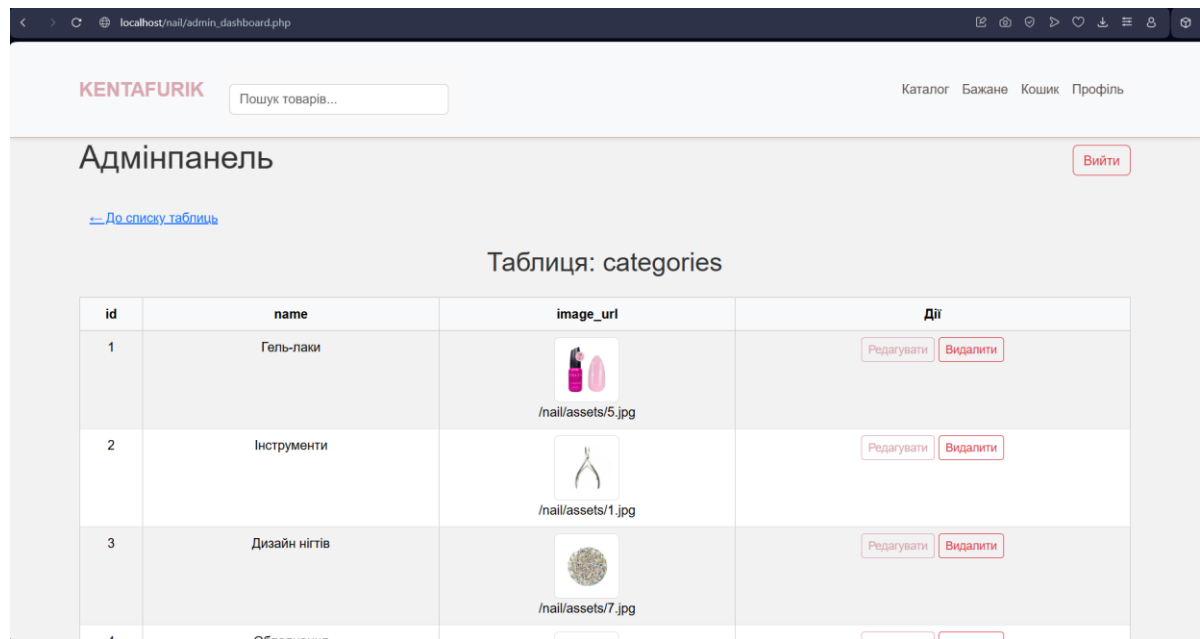


Рисунок 3.12 – Сторінка адміністратора з вибраною таблицею

Архітектура системи базується на GET-параметрі `tbl` для вибору таблиці та змінній `$action` для визначення типу операції. Механізм видалення записів реалізований через підготовлені запити:

```
if ($action === 'delete') {
    $stmt = $conn->prepare("DELETE FROM `{$tbl}` WHERE id = ?");
    $stmt->bind_param("i", $_GET['id']);
    $stmt->execute();
}
```

Універсальна обробка форм для додавання та редагування записів здійснюється через динамічне формування SQL-запитів:

```
if ($action === 'add' || $action === 'edit') {
    $fields = []; $placeholders = []; $types = '';
    foreach ($_POST as $col => $val) {
        if ($col === 'action' || $col === 'id') continue;
        $fields[] = "`$col` = ?";
        $types .= is_numeric($val) ? 'i' : 's';
        $values[] = $val;
    }
    if ($action === 'add') {
```

```

        $sql = "INSERT INTO `\$tbl` SET " . implode(',', $fields);
    } else {
        $sql = "UPDATE `\$tbl` SET " . implode(',', $fields) . "
WHERE id = ?";
        $types .= 'i'; $values[] = $_POST['id'];
    }
    $stmt = $conn->prepare($sql);
    $stmt->bind_param($types, ...$values);
    $stmt->execute();
}

```

Цей підхід автоматично обробляє всі поля форми, генеруючи відповідні плейсхолдери та типи даних для безпечного виконання запитів.

Відображення табличних даних включає спеціалізовану обробку різних типів підключення:

```

$res = $conn->query("SELECT * FROM `\$tbl`");
while ($row = $res->fetch_assoc()) {
    echo '<tr>';
    foreach ($row as $col => $val) {
        if ($col === 'image_url') {
            echo "<td><img src='{\$val}' style='max-
width:100px'></td>";
        } else {
            echo "<td>" . htmlspecialchars($val) . "</td>";
        }
    }
    // Кнопки редагування та видалення
    echo "<td>
    <a
href='?tbl={\$tbl}&action=edit&id={\$row['id']}'>Редагувати</a>
    <a href='?tbl={\$tbl}&action=delete&id={\$row['id']}'
    onclick=\"return confirm('Видалити?')\">Видалити</a>
    </td>";
    echo '</tr>';}

```

					КР.КН 25.593.10.000 ПЗ	Арк. 64
Змн.	Арк.	№ докум.	Підпис	Дата		

Особливістю системи є автоматичне розпізнавання полів `image_url` з відображенням мініатюр замість текстових значень.

Форми редагування та додавання генеруються автоматично на основі структури таблиці:

```
$cols = $conn->query("DESCRIBE `{$tbl}`")-
>fetch_all(MYSQLI_ASSOC);
echo '<form method="POST">';
echo "<input type='hidden' name='action' value='{$action}'>";
if ($action === 'edit') {
    echo "<input type='hidden' name='id' value='{$_GET['id']}'>";
}
foreach ($cols as $col) {
    if ($col['Field'] === 'id') continue;
    $value = $record[$col['Field']] ?? '';
    echo "<label>{$col['Field']}</label>";
    echo "<input name='{$col['Field']}' value='" .
htmlspecialchars($value) . "'>";
}
echo "<button>" . ($action === 'edit' ? 'Оновити' : 'Додати')
. "</button>";
echo '</form>';
```

Використання команди `DESCRIBE` дозволяє автоматично створювати поля форми відповідно до структури таблиці, підставляючи поточні значення при редагуванні. Повний лістинг програмного коду сторінки розміщено у додатку Є.

Така архітектура дозволяє адміністратору керувати всіма аспектами системи через єдиний інтерфейс, включаючи товари, користувачів, замовлення та інші сутності. Додавання нових таблиць вимагає лише передачі назви таблиці в URL-параметрі без модифікації основного коду. Система також включає захисні механізми у вигляді JavaScript-підтверджень для критичних операцій та екранування даних для запобігання XSS-атак.

					КР.КН 25.593.10.000 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Створена інформаційна система для онлайн-продажу манікюрних товарів представляє собою комплексне рішення, що успішно інтегрує передові веб-технології для забезпечення оптимального користувацького досвіду. Архітектура системи базується на перевірених і надійних технологічних рішеннях, що гарантують стабільну роботу та високу продуктивність.

Вибір технологічного стеку PHP у поєднанні з реляційною базою даних MySQL створює міцний фундамент для обробки великих обсягів даних та забезпечення швидкого відгуку системи. Frontend-частина, побудована на базі Bootstrap framework та JavaScript з використанням сучасного Fetch API, забезпечує адаптивний дизайн, що автоматично підлаштовується під різні пристрої та розміри екранів.

Особлива увага приділена забезпеченню безпеки системи. Реалізація підготовлених SQL-запитів (prepared statements) ефективно запобігає SQL-ін'єкціям, тоді як криптографічне хешування паролів із використанням сучасних алгоритмів гарантує конфіденційність облікових даних користувачів. Система авторизації та контролю доступу через механізм сесій забезпечує розмежування прав доступу між різними категоріями користувачів.

Процес оформлення та обробки замовлень виконуються в рамках транзакційних блоків, що гарантує атомарність операцій та запобігає втраті або пошкодженню даних у випадку системних збоїв. Це забезпечує високий рівень надійності при обробці платіжних операцій та управлінні товарними запасами.

Розроблена панель адміністрування дозволяє ефективно керувати всіма аспектами роботи інтернет-магазину: від управління каталогом товарів та обробки замовлень до аналізу продажів та взаємодії з клієнтами. Інтуїтивно зрозумілий інтерфейс адміністраторської панелі мінімізує час на навчання персоналу та підвищує продуктивність роботи.

Особлива увага приділена створенню інтуїтивного та зручного користувацького інтерфейсу. Швидка навігація, зручна система пошуку та

					КР.КН 25.593.10.000 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

фільтрації товарів, спрощений процес оформлення замовлень та можливість відстеження статусу доставки створюють позитивний досвід покупок для клієнтів.

3.4. Тестування інформаційної системи

Тестування інформаційної системи є важливим етапом розробки, що забезпечує її коректну роботу, стабільність, безпеку та зручність використання.

Перший етап включав детальне функціональне тестування форм реєстрації та авторизації користувачів. Тестування охоплювало реєстрацію нових користувачів з валідними даними, авторизацію зареєстрованих користувачів з коректними обліковими даними, автоматичне перенаправлення авторизованого користувача на сторінку профілю. Крім того, перевірялись негативні сценарії, включаючи авторизацію та реєстрацію з неправильним паролем.

В процесі було перевірено правильність роботи форми з коректними, некоректними та порожніми даними. Система демонструвала стабільну поведінку, видаючи відповідні повідомлення при помилках введення, включаючи валідацію на стороні клієнта та сервера (рис. 3.13, 3.14).

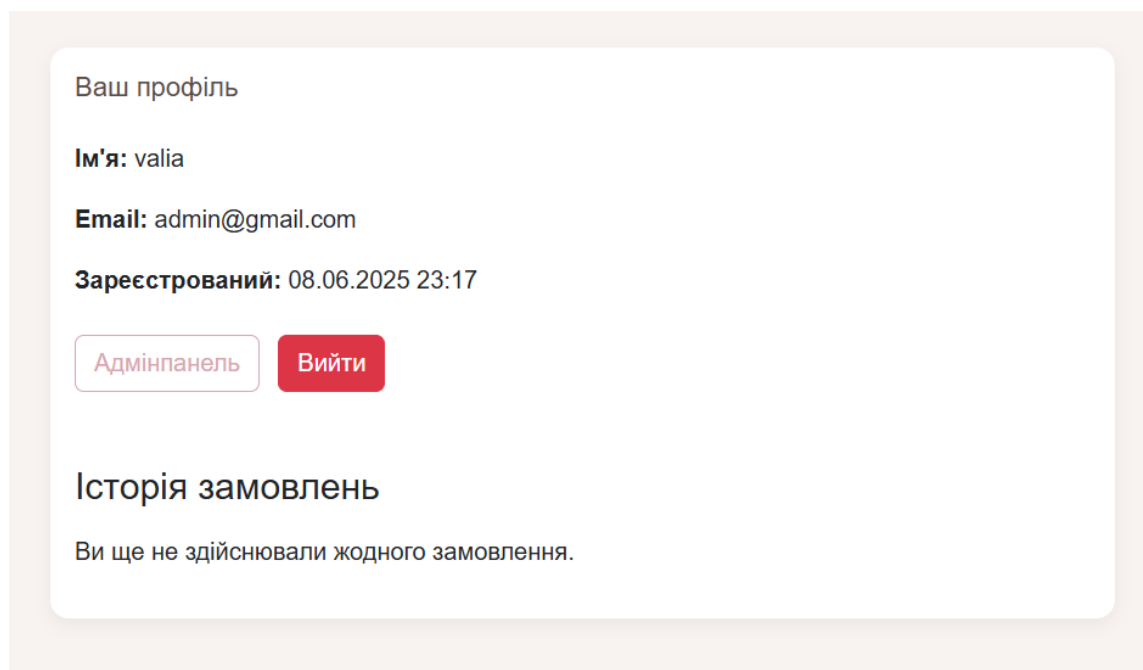


Рисунок 3.13 – Сторінка авторизації з коректними даними

Вхід

Невірний email або пароль.

Email

Пароль

Увійти

Рисунок 3.14 – Сторінка авторизації з помилковими даними

Наступним етапом стало тестування процесу додавання товарів у кошик та список бажаного. Тестування включало додавання товарів різних категорій до кошика, зміну кількості товарів у кошику, видалення окремих товарів та очищення всього кошика, збереження вмісту кошика між сеансами для авторизованих користувачів та розрахунок загальної вартості товарів (рис. 3.15 – 3.17).

Ваш кошик

Товар додано до кошика



Фото	Товар	Ціна	Кількість	Сума	Видалити
	Лампа	685.00 грн	<input type="text" value="1"/>	685.00 грн	<input type="button" value="x"/>
	Набір пензлів	140.00 грн	<input type="text" value="1"/>	140.00 грн	<input type="button" value="x"/>
Всього:				825.00 грн	

Рисунок 3.15 – Кошик із доданими товарами

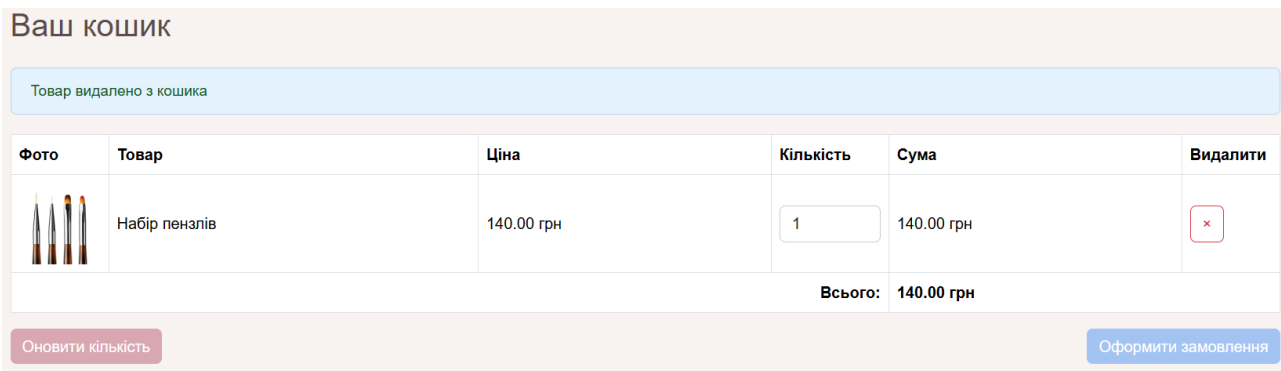


Рисунок 3.16– Видалення товару з кошика

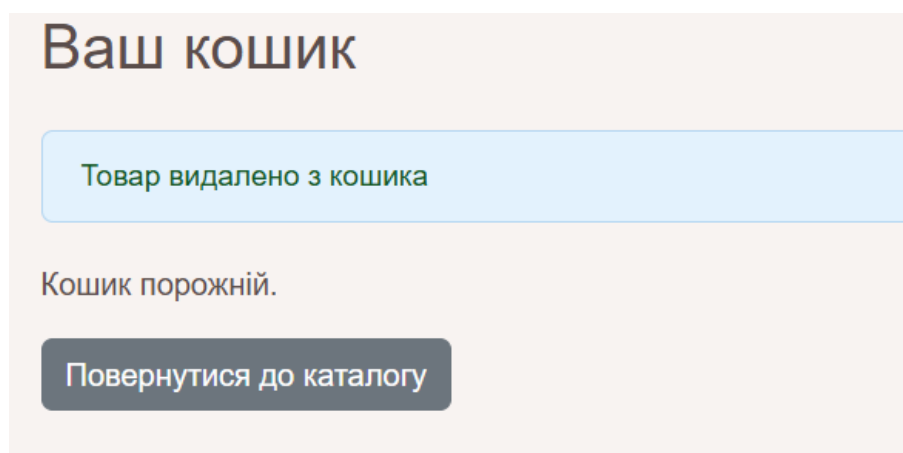


Рисунок 3.17 – Очищення кошика

Функціональність списку бажаного охоплювала додавання та видалення товарів зі списку бажаного, перенесення товарів зі списку бажаного до кошика.

Перевірялося коректне відображення товарів після їх додавання, динамічне оновлення інтерфейсу користувача та синхронізація даних з базою даних.

Система показала чітку та правильну роботу у всіх сценаріях тестування (рис. 3.18 – 3.19).

Ваше бажане



Кусачки
350.00 грн

Додати в кошик

Видалити



Блискітки You Are Cute Magic
Sparkles 06

80.00 грн

Додати в кошик

Видалити

Рисунок 3.18 – Список бажаного

Ваш кошик

Товар додано до кошика

Фото	Товар	Ціна
	Кусачки	350.00 грн

Оновити кількість

Рисунок 3.19 – Перенесення товару у кошик

Змн.	Арк.	№ докум.	Підпис	Дата

Під час тестування оформлення замовлень були проведені перевірки коректності процесу від початку до кінця. Тестування включало п'ять основних етапів: вибір товарів та опцій з перевіркою коректності відображення вибраних товарів, їх кількості та вартості; введення контактної інформації; вибір способу доставки, вибір методу оплати з інтеграцією з платіжними системами; підтвердження замовлення з фінальною перевіркою всіх даних та створенням замовлення в системі (рис. 3.20 – 3.22).

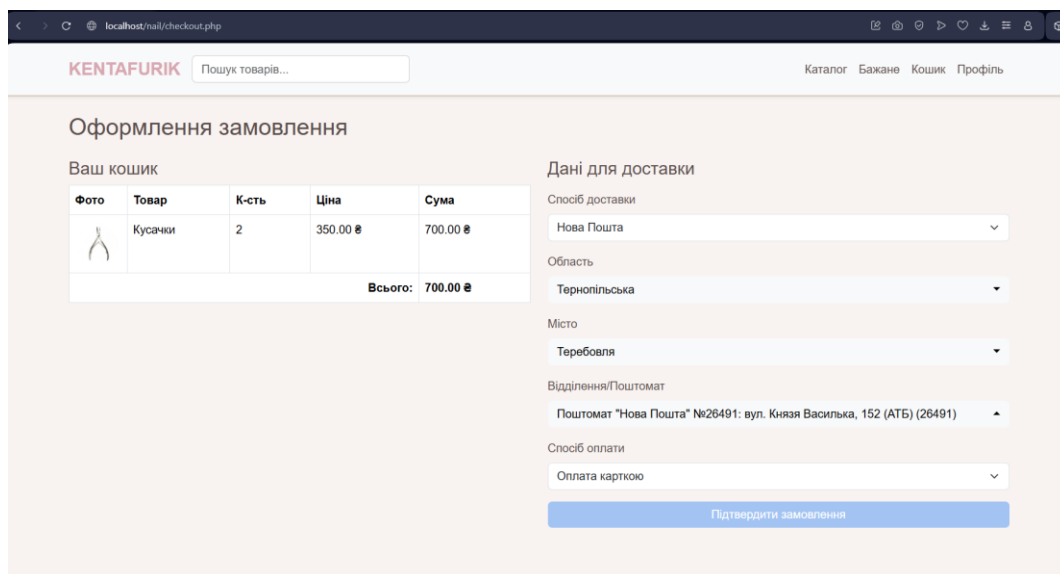


Рисунок 3.20 – Сторінка оформлення замовлення

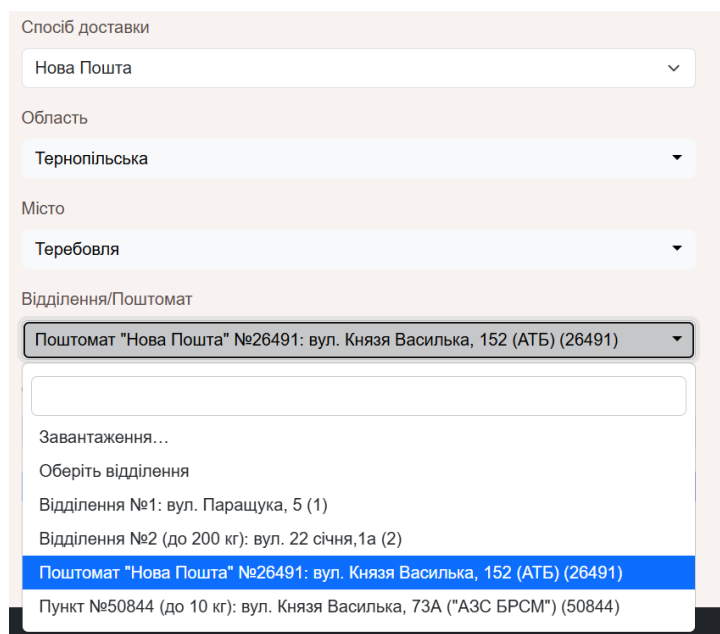


Рисунок 3.21 – Вибір доставки замовлення

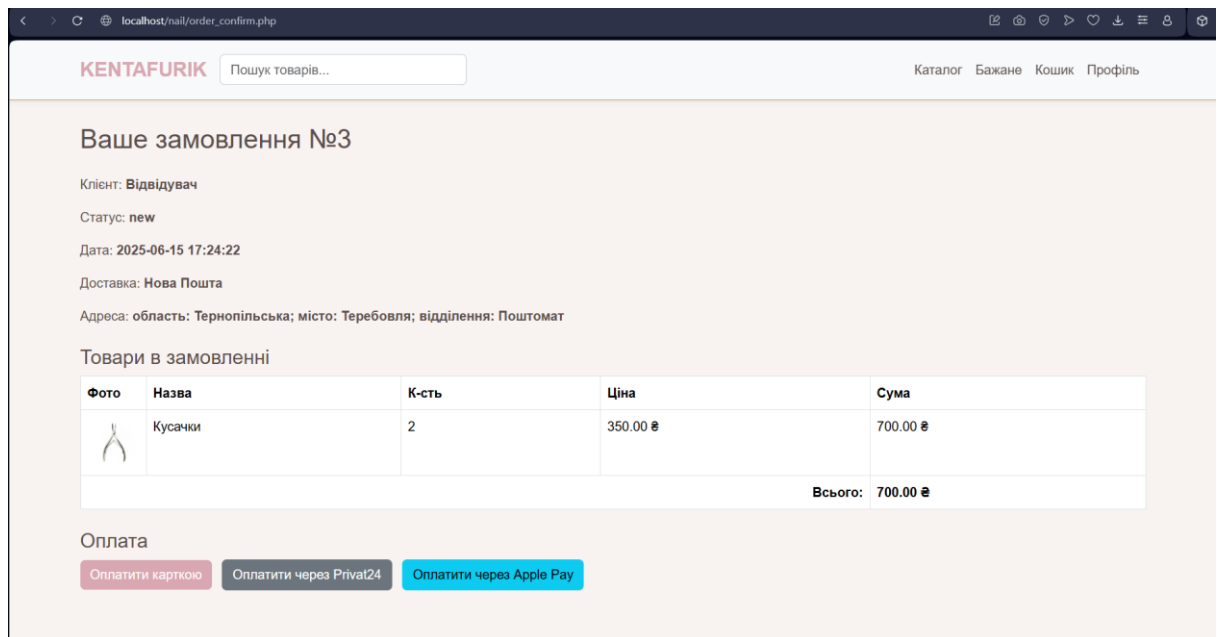


Рисунок 3.22 – Оплата замовлення

Перевірено реакцію системи на помилки введення адреси доставки, неправильні дані платіжних методів. Система успішно виконала всі операції без помилок, забезпечуючи належну валідацію на кожному етапі (рис. 3.23).

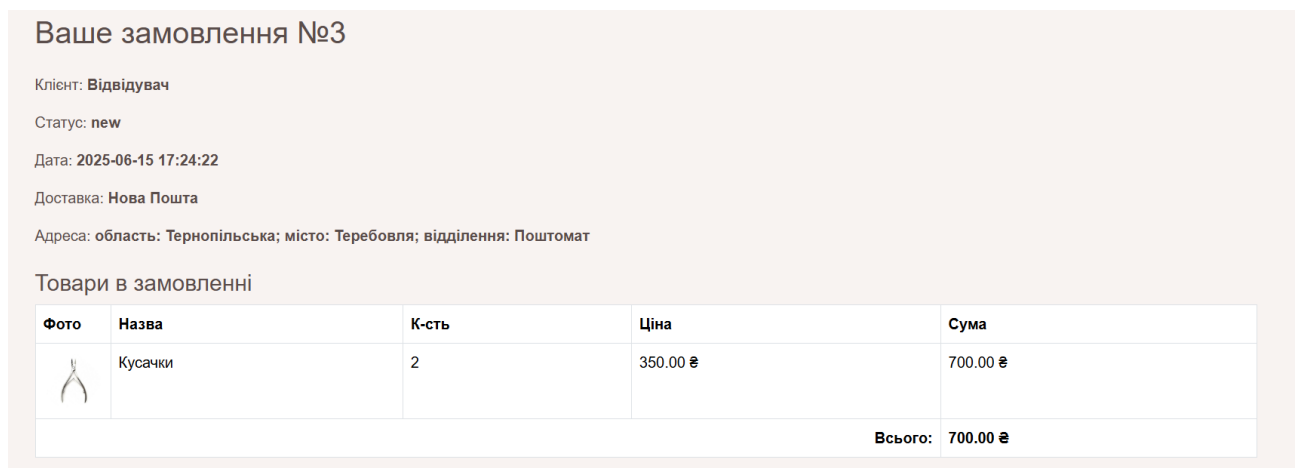


Рисунок 3.23 – Сторінка підтвердження замовлення.

Тестування адміністративної панелі передбачало всебічну перевірку можливостей адміністрування системи.

					КР.КН 25.593.10.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

Управління користувачами подано на рисунку з можливістю створення, редагування та видалення облікових записів (рис. 3.24).

The screenshot shows the admin panel for KENTAFURIK. At the top, there is a search bar and navigation links for 'Каталог', 'Бажане', 'Кошик', and 'Профіль'. The main heading is 'Адмінпанель' with a 'Вийти' button. A link '← До списку таблиць' is also present. The central part is titled 'Таблиця: users' and contains a table with columns: id, email, password_hash, name, role, created_at, and Дії. Below the table, there is a button 'Редагувати запис #3'.

id	email	password_hash	name	role	created_at	Дії
2	admin@gmail.com	\$2y\$10\$A/PIjdOdElee8zGq50BruUr4fre673GrMNI3w8E3skU0li3o5ZFm	valia	admin	2025-06-08 23:17:04	Редагувати Видалити
3	vid@gmail.com	\$2y\$10\$D5Dx/dEYVRkGPix5DnxINOLrREdOv50NITQYXxufP0X6w04vjG.Li	Відвідувач	buyer	2025-06-09 22:05:34	Редагувати Видалити
4	com@gmail.com	\$2y\$10\$KkAF0GyS14zAb2tIQRXP5O9kEZZ0W7u/CR.IT0GsEV/41FLmjli66	Комплектувальник	packer	2025-06-15 14:56:01	Редагувати Видалити

Рисунок 3.24 – Адміністративна панель із таблицею користувачів

Управління товарами з додаванням нових товарів, редагуванням описів, зміною цін (рис. 3.25-3.26)

The screenshot shows the 'Додати запис' (Add record) form in the KENTAFURIK admin panel. It includes a search bar and navigation links at the top. The form fields are: name (Наклейка для нігтів), description (Наклейка для нігтів mART Nail Sticker F634 Тварини), brand (mART), color (Чорний), and price (15).

Рисунок 3.25 – Додавання товарів

Додати запис

name

image_url

Рисунок 3.26 – Додавання категорій

Управління замовленнями з переглядом, обробкою та зміною статусів замовлень (рис. 3.27).

Замовлення №3 від 15.06.2025 17:24 Новий

Клієнт: Відвідувач (vid@gmail.com)

Доставка: Нова Пошта

Адреса:
область: Тернопільська; місто: Теремовля; відділення: Поштомат

Оплата: Карта

Товари в замовленні


Фото	Назва	К-сть	Ціна	Сума
	Кусачки	2	350.00 ₴	700.00 ₴
Всього:				700.00 ₴

Рисунок 3.27 – Зміна статусу замовлення

Особлива увага приділялась перевірці прав доступу для різних ролей адміністраторів. Всі функції працювали коректно, забезпечуючи належний рівень безпеки та захисту даних. Система адміністрування продемонструвала зручний інтерфейс та ефективні інструменти для управління всіма аспектами електронної комерції.

Результати тестування показали, без перешкод є можливість виконати базові завдання. Виявлено три незначні проблеми з навігацією, які були оперативно виправлені для покращення користувацького досвіду.

Проведено кросплатформенне тестування для забезпечення коректної роботи системи на різних пристроях та браузерах.

Тестові середовища включали браузери Chrome, Firefox, Safari, Edge та Opera, операційні системи Windows 10, та Linux Ubuntu, мобільні пристрої iOS та Android різних виробників.

Система продемонструвала високу сумісність з усіма тестовими середовищами, забезпечуючи однакову функціональність та візуальне відображення.

Адаптивний дизайн коректно масштабувався для всіх розмірів екранів, а функціональність залишалась повністю доступною на всіх платформах (рис. 3.28-3.29).

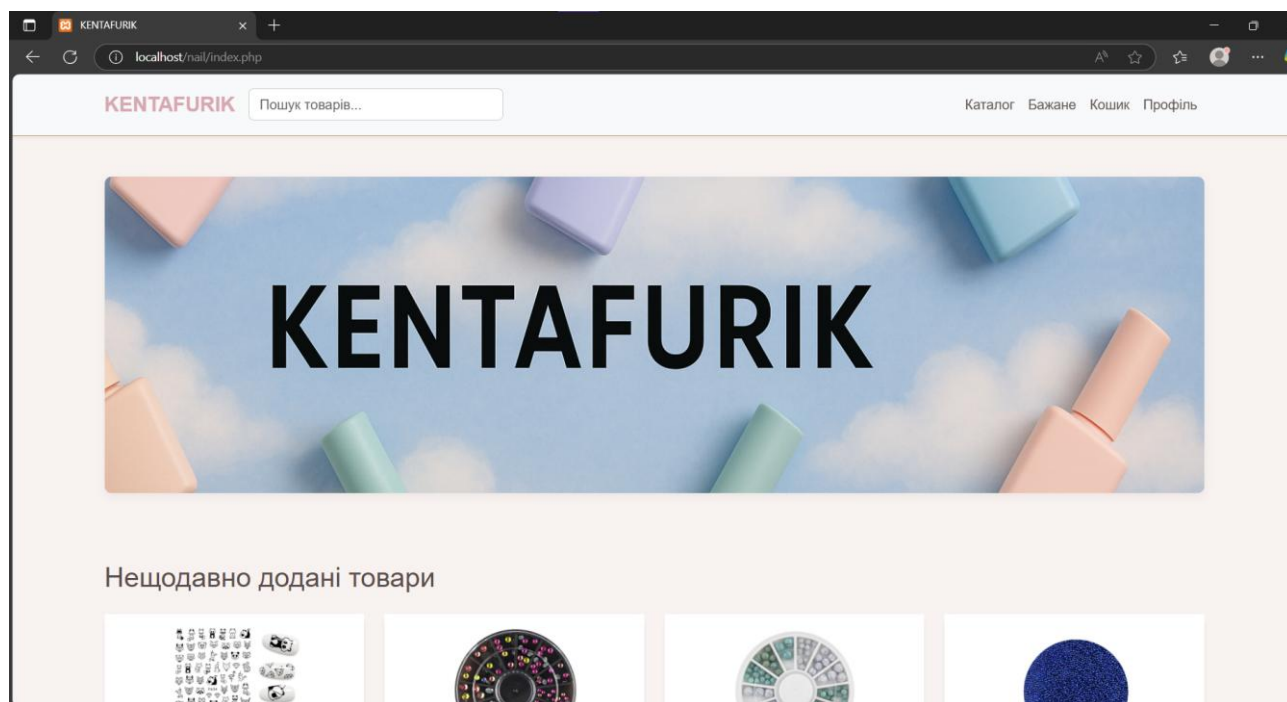


Рисунок 3.28 – Тестування в різних браузерах

					КР.КН 25.593.10.000 ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

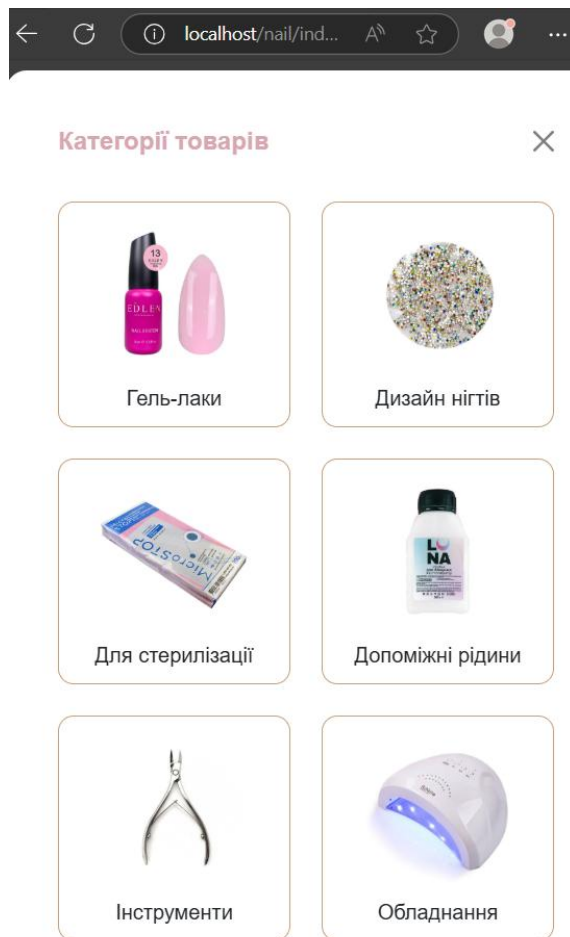


Рисунок 3.29 – Тестування на мобільному пристрої

У результаті проведеного тестування інформаційна система показала високий рівень стабільності, зручності та безпеки, відповідаючи всім поставленим вимогам. Система продемонструвала функціональну повноту з коректною роботою всіх запланованих функцій, достатню продуктивність з швидким часом відгуку, відмінну зручність використання з інтуїтивним інтерфейсом, а також кросплатформенну сумісність з коректною роботою на всіх тестових середовищах.

Виконання тестування забезпечило ефективний контроль якості та швидке виявлення можливих помилок. Система готова до повноцінної експлуатації та подальшого розвитку з можливістю масштабування відповідно до зростаючих потреб користувачів та бізнес-вимог.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

4.1 Аналіз ринку

Сучасний ринок товарів для манікюру демонструє стійку позитивну динаміку розвитку з щорічним приростом 6-8% згідно з аналітичними дослідженнями галузі. Така тенденція зумовлена декількома ключовими факторами: зростанням популярності догляду за нігтями, розширенням мережі професійних салонів краси, підвищенням стандартів особистої гігієни та естетики серед населення.

Пандемія COVID-19 кардинально трансформувала споживчі звички, прискоривши перехід до цифрових каналів продажу. Споживачі оцінили переваги онлайн-шопінгу, що призвело до формування стійкої поведінкової моделі навіть після зняття карантинних обмежень. Аналіз поведінки споживачів показує, що клієнти віддають перевагу платформам з інтуїтивним інтерфейсом, детальною продуктовою інформацією, гнучкими умовами доставки та зручними способами оплати.

Конкурентний ландшафт представлений непоганими системами, серед яких виділяються платформи «Furman» та «Naomi24». Аналіз їх діяльності виявив певні недоліки – перенасичений інформаційний контент, що ускладнює навігацію, обмежені опції логістики та платежів, застарілий дизайн інтерфейсу, недостатня персоналізація користувацького досвіду. Ці слабкі місця створюють простір для нових платформ з покращеним функціоналом та сучасним підходом до електронної комерції.

Сегментація ринку показує особливо високий попит на професійні гель-лаки, інноваційні інструменти та технологічне обладнання, включаючи UV/LED лампи, апаратні засоби для манікюру та системи вентиляції робочих місць.

Розробка спеціалізованої платформи з акцентом на якість обслуговування, технологічність рішень та естетику презентації створює значний конкурентний потенціал.

					КР.КН 25.593.10.000 ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2 Економічний розрахунок проектних витрат

Для визначення економічної доцільності створення онлайн-системи продажу товарів для манікюру необхідно провести розрахунок витрат, пов'язаних із її розробкою. Розрахунок охоплює основні категорії витрат, включаючи оплату праці, внески до фондів, витрати на електроенергію, програмне забезпечення, технічні засоби та інші супутні витрати.

Загальна сума витрат на розробку (Срозр) визначається за формулою:

$$Срозр = Сзп + Свн + Сел + Спр + Сінш$$

де: – Сзп – витрати на заробітну плату; – Свн – витрати на внески до фондів (ЄСВ); – Сел – витрати на електроенергію; – Спр – витрати на програмне забезпечення та технічні засоби; – Сінш – інші супутні витрати.

Розрахунок витрат на заробітну плату (Сзп):

- Веб-розробники: 2 фахівці × 180 год × 160 грн/год = 57 600 грн
- Дизайнер: 1 фахівець × 100 год × 140 грн/год = 14 000 грн
- Тестувальник: 1 фахівець × 80 год × 130 грн/год = 10 400 грн

Загальні витрати на заробітну плату: Сзп = 57 600 + 14 000 + 10 400 = 82 000 грн

Єдиний соціальний внесок (ЄСВ) становить 22%: Свн = 82 000 × 0,22 = 18 040 грн

Витрати на електроенергію визначаються із середньою потужністю одного робочого місця – 0,044 кВт (ноутбук 20 Вт + монітор 24 Вт), за тарифом 6 грн/кВт·год:

- Веб-розробники: 360 год × 0,044 кВт × 6 грн = 95,04 грн
- Дизайнер: 100 год × 0,044 кВт × 6 грн = 26,40 грн
- Тестувальник: 80 год × 0,044 кВт × 6 грн = 21,12 грн
- Сел = 95,04 + 26,40 + 21,12 = 142,56 грн

Витрати на програмне забезпечення та технічні засоби:

- Ліцензійне ПЗ (середовища розробки): 12 000 грн

					КР.КН 25.593.10.000 ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

– Оренда серверу та домену на 1 рік: 9 000 грн

– Спр = 12 000 + 9 000 = 21 000 грн

Інші супутні витрати:

– Канцелярські товари: 800 грн

– Інтернет (3 місяці): 510 грн

– Дрібні офісні витрати: 1 000 грн

Сінш = 800 + 510 + 1 000 = 2 310 грн

Загальні витрати на розробку: Срозр = 82 000 + 18 040 + 142,56 + 21 000 + 2 310 = 123 492,56 грн

Щомісячні витрати після запуску:

– Зарплата модератора: 20 робочих днів × 4 год/день × 100 грн/год = 8 000 грн

– Вартість оренди серверу: 750 грн/місяць

Сукупні місячні витрати: 8 750 грн

Економічний ефект автоматизації:

При ручній обробці замовлень витрачалося близько 20 хв на замовлення, після автоматизації – 5 хв. Середня вартість години праці менеджера – 100 грн, кількість замовлень за місяць – 300:

$E = (0,333 - 0,0833) \times 100 \text{ грн} \times 300 = 7\,490 \text{ грн/місяць}$

Щорічний економічний ефект становитиме: $E = 7\,490 \times 12 = 89\,880 \text{ грн}$

Термін окупності проєкту: $\text{Ток} = 123\,492,56 / 89\,880 \approx 1,37 \text{ року (16,4 місяця)}$

Коефіцієнт економічної ефективності: $\text{Кеф} = 89\,880 / 123\,492,56 \approx 0,73$

Отже, розробка інформаційної системи економічно доцільна завдяки високій ефективності та порівняно короткому терміну окупності.

					КР.КН 25.593.10.000 ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3 Обґрунтування доцільності проекту

Розробка онлайн-системи продажу манікюрної індустрії відповідає стратегічним трендам цифровізації економіки та зміни споживчих прераференцій. Онлайн-платформа забезпечує суттєві конкурентні переваги порівняно з традиційними каналами збуту. Операційна ефективність досягається через автоматизацію ключових бізнес-процесів: обробки замовлень, управління інвентарем, клієнтського сервісу та логістичних операцій. Це дозволяє значно скоротити потребу в персоналі, оптимізувати складські витрати та підвищити швидкість обслуговування клієнтів. Система працює цілодобово без вихідних, що максимізує доступність для клієнтів та потенціал до збільшення доходів.

Технологічні можливості платформи створюють унікальні інструменти для підвищення якості клієнтського досвіду. Персоналізовані рекомендації на основі історії покупок, система лояльності з накопичувальними знижками, детальна аналітика поведінки користувачів дозволяють формувати довгострокові клієнтські відносини та збільшувати середній чек замовлення.

Стратегічна гнучкість системи забезпечує можливості для швидкого масштабування асортименту, інтеграції з новими постачальниками, розширення географічного покриття та адаптації до змін ринкових умов. Цифрова природа платформи дозволяє швидко тестувати нові бізнес-моделі, промоційні механізми та канали маркетингу без значних капітальних інвестицій. Аналітичні можливості системи забезпечують менеджмент даними в реальному часі для прийняття обґрунтованих бізнес-рішень. Моніторинг продажів, аналіз попиту, оптимізація цінової політики та прогнозування трендів створюють основу для стратегічного планування та конкурентного позиціонування.

Соціальний вимір проекту полягає у підвищенні доступності якісних товарів для манікюру. Розвиток електронної комерції сприяє створенню нових робочих місць у сфері інформаційних технологій. Таким чином, даний проект характеризується високим ступенем економічної доцільності, стратегічної перспективності та соціальної значущості, що робить його інвестиційно привабливим та ринково конкурентоспроможним рішенням.

					КР.КН 25.593.10.000 ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В процесі виконання кваліфікаційної роботи було розроблено повноцінну web-базовану інформаційну систему для онлайн-продажу товарів для манікюру. Актуальність даної теми зумовлена зростанням попиту на електронну комерцію та необхідністю забезпечення зручного інструменту для придбання професійних засобів як для майстрів манікюру, так і для побутових споживачів. У процесі роботи виконано дослідження предметної області, вивчено особливості аналогічних рішень, визначено функціональні та нефункціональні вимоги до системи.

На основі проведеного аналізу було спроектовано архітектуру системи з тривірневою структурою: презентаційний рівень (інтерфейс), логічний рівень (серверна частина) та рівень даних (база даних MySQL). Структуру бази даних розроблено із врахуванням нормалізації, що забезпечує логічну цілісність даних та уникнення дублювання. Створено механізми реєстрації, авторизації, обробки замовлень, керування товарами та перегляду історії покупок. Адміністративна панель системи розроблена з акцентом на простоту використання та функціональну повноту.

Інтерфейс системи спроектовано з урахуванням принципів зручності, простоти навігації та адаптивності до мобільних пристроїв. Структура меню логічно організована, що дозволяє користувачеві швидко знаходити необхідні функції. Застосовано сучасні технології – HTML, CSS, JavaScript та PHP, які забезпечили інтерактивність і динамічну взаємодію з користувачем.

Результатом проєкту є працездатна система, готова до впровадження в реальних умовах. Її використання дозволить автоматизувати процеси онлайн-продажу, підвищити якість обслуговування клієнтів та забезпечити ефективне адміністрування інтернет-магазину. Проєкт має практичну цінність та може бути основою для подальшого розширення функціоналу.

					КР.КН 25.593.10.000 ПЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. PHP: Безпека PHP. Офіційна документація. URL: <https://www.php.net/manual/uk/security.intro.php> (дата звернення: 10.02.2025).
2. MySQL: Офіційна документація. URL: <https://dev.mysql.com/doc/refman/8.0/en/> (дата звернення: 25.02.2025).
3. Bootstrap 5: Вступ та компоненти. Офіційна документація. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата звернення: 05.03.2025).
4. MDN Web Docs: Fetch API. URL: https://developer.mozilla.org/uk/docs/Web/API/Fetch_API (дата звернення: 13.04.2025).
5. Nova Poshta Developers: API-документація. URL: <https://developers.novaposhta.ua/> (дата звернення: 14.04.2025).
6. DOU.UA: Як створити інтернет-магазин на PHP та MySQL. URL: <https://dou.ua/lenta/articles/php-mysql-shop/> (дата звернення: 19.04.2025).
7. Nielsen Norman Group: E-commerce UX: як дизайн впливає на конверсію. URL: <https://www.nngroup.com/articles/ecommerce-ux/> (дата звернення: 05.05.2025).
8. OWASP. SQL Injection Prevention Cheat Sheet. OWASP Foundation. URL: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html (дата звернення: 16.05.2025).
9. PHP: Безпечні сесії. Офіційна документація PHP. URL: <https://www.php.net/manual/uk/session.security.php> (дата звернення: 26.05.2025).
10. Google Developers. Lighthouse: інструмент для поліпшення продуктивності веб-сайтів. Google Developers. URL: <https://developers.google.com/web/tools/lighthouse> (дата звернення: 01.06.2025).
11. Smashing Magazine. Best Practices for E-Commerce Performance Optimization. Smashing Magazine. URL: <https://www.smashingmagazine.com/2018/01/performance-optimization-e-commerce/> (дата звернення: 09.06.2025).

					КР.КН 25.593.10.000 ПЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		


```

) ENGINE=InnoDB
  DEFAULT CHARSET=utf8mb4
  COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `cart_items` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `user_id` INT(11) DEFAULT NULL,
  `product_id` INT(11) DEFAULT NULL,
  `quantity` INT(11) NOT NULL DEFAULT 1,
  `added_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  KEY `idx_cart_user` (`user_id`),
  KEY `idx_cart_product` (`product_id`),
  CONSTRAINT `fk_cart_user`
  FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT `fk_cart_product`
  FOREIGN KEY (`product_id`) REFERENCES `products` (`id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE
) ENGINE=InnoDB
  DEFAULT CHARSET=utf8mb4
  COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `wishlist` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `user_id` INT(11) DEFAULT NULL,
  `product_id` INT(11) DEFAULT NULL,
  `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  KEY `idx_wish_user` (`user_id`),
  KEY `idx_wish_product` (`product_id`),
  CONSTRAINT `fk_wish_user`
  FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT `fk_wish_product`
  FOREIGN KEY (`product_id`) REFERENCES `products` (`id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE
) ENGINE=InnoDB
  DEFAULT CHARSET=utf8mb4
  COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `orders` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `user_id` INT(11) DEFAULT NULL,
  `status`
  ENUM('new','processing','shipped','delivered','cancelled') NOT NULL
  DEFAULT 'new',

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		

```

`total` DECIMAL(10,2) DEFAULT NULL,
`delivery_method` VARCHAR(100) DEFAULT NULL,
`delivery_address` TEXT,
`payment_method` VARCHAR(50) DEFAULT NULL,
`created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (`id`),
KEY `idx_orders_user` (`user_id`),
CONSTRAINT `fk_orders_user`
FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
ON DELETE SET NULL
ON UPDATE CASCADE
) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `order_items` (
`id` INT(11) NOT NULL AUTO_INCREMENT,
`order_id` INT(11) DEFAULT NULL,
`product_id` INT(11) DEFAULT NULL,
`quantity` INT(11) DEFAULT NULL,
`price` DECIMAL(10,2) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `idx_orderitems_order` (`order_id`),
KEY `idx_orderitems_product` (`product_id`),
CONSTRAINT `fk_orderitems_order`
FOREIGN KEY (`order_id`) REFERENCES `orders` (`id`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_orderitems_product`
FOREIGN KEY (`product_id`) REFERENCES `products` (`id`)
ON DELETE SET NULL
ON UPDATE CASCADE
) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Б

SQL-запити для заповнення таблиць записами

```
INSERT INTO `categories` (`name`, `image_url`) VALUES
  ('Лаки', 'assets/images/cat_lak.jpg'),
  ('Інструменти', 'assets/images/cat_tools.jpg'),
  ('Декор', 'assets/images/cat_decor.jpg');
```

```
INSERT INTO `users` (`email`, `password_hash`, `name`, `role`)
VALUES
  ('admin@site.ua',
  '$2y$10$abcdefghijklmnopqrstuv1234567890abcdefghijklmnopghi', 'Адмін',
  'admin'),
  ('buyer1@site.ua',
  '$2y$10$abcdefghijklmnopqrstuv1234567890abcdefghijklmnopghi', 'Іван
  Покупець', 'buyer'),
  ('packer@site.ua', '$2y$10$abcdefghijklmnopqrstuv1234567890abcdefghijklmnopghi
  ', 'Петро Пакувальник', 'packer');
```

```
INSERT INTO `products`
  (`name`, `description`, `price`, `image_url`, `stock`,
  `category_id`, `brand`, `color`)
VALUES
  ('Червоний лак №1', 'Насичений червоний лак', 99.00,
  'assets/images/naill.jpg', 10, 1, 'BrandA', 'Червоний'),
  ('Пилка для нігтів', 'Надійна пилка з двома
  гранями', 49.50, 'assets/images/file.jpg', 25, 2, 'BrandB',
  'Сірий'),
  ('Набір стразів', 'Різнокольорові стрази для
  декору', 20.00, 'assets/images/stras.jpg', 100, 3, 'BrandC',
  'Різнокольоровий');
```

```
INSERT INTO `cart_items` (`user_id`, `product_id`, `quantity`)
VALUES
  (2, 1, 2);
```

```
INSERT INTO `wishlist` (`user_id`, `product_id`) VALUES
  (2, 2);
```

```
INSERT INTO `orders`
  (`user_id`, `status`, `total`, `delivery_method`,
  `delivery_address`, `payment_method`)
VALUES
  (2, 'new', 198.00, 'Нова Пошта', 'область: Київська; місто:
  Київ; відділення: №10', 'Карта');
```

```
INSERT INTO `order_items` (`order_id`, `product_id`, `quantity`,
  `price`) VALUES (1, 1, 2, 99.00);
```

					КР.КН 25.593.10.000 ПЗ	Арк.
						86
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток В

Програмний код сторінки входу та реєстрації

```
<?php
// profile.php

session_start();
include 'includes/db.php';

if (isset($_SESSION['user_id'])) {
    header("Location: profile_page.php");
    exit();
}

$login_error      = '';
$register_error    = '';
$register_success  = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
    isset($_POST['login'])) {
    $stmt = $conn->prepare("SELECT id,password_hash,role FROM users
WHERE email = ?");
    $stmt->bind_param("s", $_POST['email']);
    $stmt->execute();
    $u = $stmt->get_result()->fetch_assoc();
    if ($u && password_verify($_POST['password'],
    $u['password_hash'])) {
        $_SESSION['user_id'] = $u['id'];
        $_SESSION['role']    = $u['role'];
        header("Location: profile_page.php");
        exit();
    }
    $login_error = "Невірний email або пароль.";
}

if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
    isset($_POST['register'])) {
    $stmt = $conn->prepare("SELECT id FROM users WHERE email = ?");
    $stmt->bind_param("s", $_POST['email']);
    $stmt->execute();
    if ($stmt->get_result()->num_rows) {
        $register_error = "Цей email уже використовується.";
    } else {
        $hash = password_hash($_POST['password'],
PASSWORD_DEFAULT);
        $ins = $conn->prepare("
INSERT INTO users (email,password_hash,name,role)
VALUES (?, ?, ?, 'buyer')
");
    }
}
```

									Арк.
									87
Змн.	Арк.	№ докум.	Підпис	Дата					

```

        $ins->bind_param("sss", $_POST['email'], $hash,
$_POST['name']);
        if ($ins->execute()) {
            $register_success = "Реєстрація пройшла успішно.";
        } else {
            $register_error = "Сталася помилка, спробуйте
пізніше.";
        }
    }
}

include 'includes/header.php';
?>
<div class="container py-5">
    <div class="row justify-content-center g-4">
        <!-- Форма входу -->
        <div class="col-md-5">
            <div class="card shadow-sm">
                <div class="card-body">
                    <h3 class="card-title mb-4">Вхід</h3>
                    <?php if($login_error): ?>
                        <div class="alert alert-danger"><?=
htmlspecialchars($login_error) ?></div>
                    <?php endif; ?>
                    <form method="POST">
                        <input type="hidden" name="login" value="1">
                        <div class="mb-3">
                            <label class="form-label">Email</label>
                            <input type="email" name="email" class="form-control"
required>
                        </div>
                        <div class="mb-3">
                            <label class="form-label">Пароль</label>
                            <input type="password" name="password" class="form-
control" required>
                        </div>
                        <button type="submit" class="btn btn-primary w-
100">Увійти</button>
                    </form>
                </div>
            </div>
        </div>
        <!-- Форма реєстрації -->
        <div class="col-md-5">
            <div class="card shadow-sm">
                <div class="card-body">
                    <h3 class="card-title mb-4">Реєстрація</h3>
                    <?php if($register_error): ?>
                        <div class="alert alert-danger"><?=
htmlspecialchars($register_error) ?></div>

```

					КР.КН 25.593.10.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		88

```

        <?php elseif($register_success): ?>
            <div class="alert alert-success"><?=
htmlspecialchars($register_success) ?></div>
        <?php endif; ?>
        <form method="POST">
            <input type="hidden" name="register" value="1">
            <div class="mb-3">
                <label class="form-label">Имя</label>
                <input type="text" name="name" class="form-control"
required>
            </div>
            <div class="mb-3">
                <label class="form-label">Email</label>
                <input type="email" name="email" class="form-control"
required>
            </div>
            <div class="mb-3">
                <label class="form-label">Пароль</label>
                <input type="password" name="password" class="form-
control" required>
            </div>
            <button type="submit" class="btn btn-success w-
100">Зареєструватися</button>
        </form>
    </div>
</div>
</div>
</div>
</div>
<?php include 'includes/footer.php'; ?>

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						89
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Г

Програмний код кошику товарів

```
<?php
// cart.php

session_start();
include 'includes/db.php';

if (!isset($_SESSION['user_id'])) {
    header("Location: profile.php");
    exit();
}
$user_id = $_SESSION['user_id'];

if (isset($_GET['add']) && is_numeric($_GET['add'])) {
    $pid = (int)$_GET['add'];
    $chk = $conn->prepare("
        SELECT id, quantity
        FROM cart_items
        WHERE user_id = ? AND product_id = ?
    ");
    $chk->bind_param("ii", $user_id, $pid);
    $chk->execute();
    $res = $chk->get_result();

    if ($row = $res->fetch_assoc()) {
        $newQty = $row['quantity'] + 1;
        $upd = $conn->prepare("
            UPDATE cart_items
            SET quantity = ?
            WHERE id = ?
        ");
        $upd->bind_param("ii", $newQty, $row['id']);
        $upd->execute();
    } else {
        $ins = $conn->prepare("
            INSERT INTO cart_items (user_id, product_id, quantity)
            VALUES (?, ?, 1)
        ");
        $ins->bind_param("ii", $user_id, $pid);
        $ins->execute();
    }

    header("Location: cart.php?msg=" . urlencode("Товар додано до кошика"));
    exit();
}
```

					КР.КН 25.593.10.000 ПЗ	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (isset($_GET['remove']) && is_numeric($_GET['remove'])) {
    $cid = (int)$_GET['remove'];
    $del = $conn->prepare("
        DELETE FROM cart_items
        WHERE id = ? AND user_id = ?
    ");
    $del->bind_param("ii", $cid, $user_id);
    $del->execute();

    header("Location: cart.php?msg=" . urlencode("Товар видалено з
кошика"));
    exit();
}

if (isset($_POST['update_cart'])) {
    foreach ($_POST['quantities'] as $cid => $qty) {
        $cid = (int)$cid;
        $qty = max(1, (int)$qty);
        $upd = $conn->prepare("
            UPDATE cart_items
            SET quantity = ?
            WHERE id = ? AND user_id = ?
        ");
        $upd->bind_param("iii", $qty, $cid, $user_id);
        $upd->execute();
    }

    header("Location: cart.php?msg=" . urlencode("Кількість
оновлена"));
    exit();
}

include 'includes/header.php';

$msg = $_GET['msg'] ?? '';

$stmt = $conn->prepare("
    SELECT
        ci.id AS cid,
        p.id AS pid,
        p.name,
        p.price,
        p.image_url,
        ci.quantity
    FROM cart_items ci
    JOIN products p ON ci.product_id = p.id
    WHERE ci.user_id = ?
");
$stmt->bind_param("i", $user_id);
$stmt->execute();

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$cart = $stmt->get_result();

$total = 0;
$items = [];
while ($row = $cart->fetch_assoc()) {
    $row['line_total'] = $row['price'] * $row['quantity'];
    $total += $row['line_total'];
    $items[] = $row;
}
?>

<div class="container py-5">
    <h2 class="mb-4">Ваш кошик</h2>

    <?php if ($msg): ?>
        <div id="alert" class="alert alert-success"><?=  

htmlspecialchars($msg) ?></div>
        <script>
            setTimeout(() => document.getElementById('alert')?.remove(),
3000);
        </script>
    <?php endif; ?>

    <?php if (empty($items)): ?>
        <p>Кошик порожній.</p>
        <a href="index.php" class="btn btn-secondary">Повернутися до  

каталогу</a>
    <?php else: ?>
        <form method="POST" action="cart.php">
            <table class="table table-bordered align-middle">
                <thead>
                    <tr>
                        <th>Фото</th>
                        <th>Товар</th>
                        <th>Ціна</th>
                        <th>Кількість</th>
                        <th>Сума</th>
                        <th>Видалити</th>
                    </tr>
                </thead>
                <tbody>
                    <?php foreach ($items as $it): ?>
                        <tr>
                            <td style="width:100px;">
                                ">
                                </td>
                            <td><?=  

htmlspecialchars($it['name']) ?></td>
                            <td><?=  

number_format($it['price'],2) ?> грн</td>

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Д

Програмний код сторінки бажаного

```
<?php
// wishlist.php

session_start();
include 'includes/db.php';

if (!isset($_SESSION['user_id'])) {
    header("Location: profile.php");
    exit();
}
$user_id = $_SESSION['user_id'];

$success_msg = '';

if (isset($_GET['add']) && is_numeric($_GET['add'])) {
    $pid = (int)$_GET['add'];
    $check = $conn->prepare("
        SELECT id
        FROM wishlist
        WHERE user_id = ? AND product_id = ?
    ");
    $check->bind_param("ii", $user_id, $pid);
    $check->execute();
    if ($check->get_result()->num_rows === 0) {
        $ins = $conn->prepare("
            INSERT INTO wishlist (user_id, product_id)
            VALUES (?, ?)
        ");
        $ins->bind_param("ii", $user_id, $pid);
        $ins->execute();
        $success_msg = "Товар додано до бажаного";
    } else {
        $success_msg = "Товар уже є у вашому бажаному";
    }
    header("Location: wishlist.php?msg=" .
    urlencode($success_msg));
    exit();
}

if (isset($_GET['remove']) && is_numeric($_GET['remove'])) {
    $wid = (int)$_GET['remove'];
    $del = $conn->prepare("
        DELETE FROM wishlist
        WHERE id = ? AND user_id = ?
    ");
    $del->bind_param("ii", $wid, $user_id);
```

									Арк.
									94
Змн.	Арк.	№ докум.	Підпис	Дата					


```

        Додати в кошик
    </a>
    <a href="wishlist.php?remove=<?= $item['wid'] ?>"
        class="btn btn-outline-danger w-100"
        onclick="return confirm('Видалити цей товар з
вашого бажаного?')">
        Видалити
    </a>
</div>
</div>
</div>
</div>
<?php endwhile; ?>
</div>
<?php else: ?>
    <p>Ваш список бажаного порожній.</p>
    <a href="index.php" class="btn btn-secondary">Повернутися до
каталогу</a>
    <?php endif; ?>
</div>

<script>
document.addEventListener('DOMContentLoaded', () => {
    const alert = document.getElementById('alert');
    if (alert) setTimeout(() => alert.remove(), 3000);
});
</script>

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

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Е

Програмний код сторінки оформлення замовлення

```
<?php
// checkout.php

session_start();
include 'includes/db.php';

if (!isset($_SESSION['user_id'])) {
    header("Location: profile.php");
    exit;
}
$user_id = $_SESSION['user_id'];

$stmtCart = $conn->prepare("
    SELECT ci.product_id, ci.quantity, p.name, p.price, p.image_url
    FROM cart_items ci
    JOIN products p ON ci.product_id = p.id
    WHERE ci.user_id = ?
");
$stmtCart->bind_param("i", $user_id);
$stmtCart->execute();
$cart = $stmtCart->get_result();

if ($cart->num_rows === 0) {
    header("Location: cart.php?msg=" . urlencode("Ваш кошик
    порожній"));
    exit;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    $stmt = $conn->prepare("
        SELECT ci.product_id, ci.quantity, p.name, p.price
        FROM cart_items ci
        JOIN products p ON ci.product_id = p.id
        WHERE ci.user_id = ?
    ");
    $stmt->bind_param("i", $user_id);
    $stmt->execute();
    $cartProc = $stmt->get_result();
    $total = 0;
    $items = [];
    while ($r = $cartProc->fetch_assoc()) {
        $line = $r['price'] * $r['quantity'];
        $total += $line;
        $items[] = $r;
    }
}
```

									Арк.
									97
Змн.	Арк.	№ докум.	Підпис	Дата					

```

$delivery_method = $_POST['delivery_method'];
if ($delivery_method === 'Нова Пошта') {
    $delivery_address = sprintf(
        "область: %s; місто: %s; відділення: %s",
        $_POST['areaDesc'], $_POST['cityDesc'],
$_POST['warehouseDesc']
    );
} else {
    $delivery_address = trim($_POST['delivery_address_text']);
}
$payment_method = $_POST['payment_method'];

$conn->begin_transaction();
try {
    // orders
    $insO = $conn->prepare("
        INSERT INTO orders
        (user_id, status, total, delivery_method,
delivery_address, payment_method)
        VALUES (?, 'new', ?, ?, ?, ?)
    ");
    $insO->bind_param("idsss", $user_id, $total,
$delivery_method, $delivery_address, $payment_method);
    $insO->execute();
    $order_id = $conn->insert_id;
    $insI = $conn->prepare("
        INSERT INTO order_items
        (order_id, product_id, quantity, price)
        VALUES (?, ?, ?, ?)
    ");
    foreach ($items as $it) {
        $insI->bind_param("iiid",
            $order_id,
            $it['product_id'],
            $it['quantity'],
            $it['price']
        );
        $insI->execute();
    }

    $del = $conn->prepare("DELETE FROM cart_items WHERE user_id
= ?");
    $del->bind_param("i", $user_id);
    $del->execute();

    $conn->commit();

    header("Location: order_confirm.php?id=$order_id");
    exit;
} catch (Exception $e) {
    $conn->rollback();
}

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $error = "Сталася помилка оформлення. Спробуйте пізніше.";
    }
}

include 'includes/header.php';
?>
<link
    rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap-select@1.14.0-beta3/dist/css/bootstrap-select.min.css"
/>

<div class="container py-5">
    <h2 class="mb-4">Оформлення замовлення</h2>

    <?php if (!empty($error)): ?>
        <div class="alert alert-danger"><?= htmlspecialchars($error)
?></div>
    <?php endif; ?>

    <div class="row">
        <!-- Ліва колонка: Підсумок кошика -->
        <div class="col-md-6 mb-4">
            <h4>Ваш кошик</h4>
            <table class="table table-bordered">
                <thead>
                    <tr>
                        <th>Фото</th><th>Товар</th><th>К-
сть</th><th>Ціна</th><th>Сума</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                    $cart->data_seek(0);
                    $sum = 0;
                    while ($r = $cart->fetch_assoc()):
                        $line = $r['price'] * $r['quantity'];
                        $sum += $line;
                    ?>
                    <tr>
                        <td style="width:80px;">
                            
                        </td>
                        <td><?= htmlspecialchars($r['name']) ?></td>
                        <td><?= $r['quantity'] ?></td>
                        <td><?= number_format($r['price'],2) ?> ₴</td>
                        <td><?= number_format($line,2) ?> ₴</td>
                    </tr>
                    <?php endwhile; ?>
                <tr>

```

					КР.КН 25.593.10.000 ПЗ	Арк. 99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <td colspan="4" class="text-
end"><strong>Всього:</strong></td>
        <td><strong><?= number_format($sum,2) ?>
&</strong></td>
    </tr>
</tbody>
</table>
</div>
<div class="col-md-6">
    <form method="post" action="checkout.php">
        <h4 class="mb-3">Дані для доставки</h4>

        <div class="mb-3">
            <label class="form-label">Спосіб доставки</label>
            <select id="delivery_method" name="delivery_method"
class="form-select" required>
                <option value="Нова Пошта">Нова Пошта</option>
                <option value="УкрПошта">УкрПошта</option>
                <option value="Самовивіз">Самовивіз</option>
            </select>
        </div>

        <div id="np-fields" style="display:none;">
            <div class="mb-3">
                <label class="form-label">Область</label>
                <select id="np-area"
                    class="selectpicker"
                    data-live-search="true"
                    data-width="100%"
                    title="Оберіть область">
                </select>
            </div>
            <div class="mb-3">
                <label class="form-label">Місто</label>
                <select id="np-city"
                    class="selectpicker"
                    data-live-search="true"
                    data-width="100%"
                    title="Оберіть місто">
                </select>
            </div>
            <div class="mb-3">
                <label class="form-label">Відділення/Поштомот</label>
                <select id="np-warehouse"
                    class="selectpicker"
                    data-live-search="true"
                    data-width="100%"
                    title="Оберіть відділення">
                </select>
            </div>
            <input type="hidden" name="areaDesc" id="areaDesc">

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						100
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <input type="hidden" name="cityDesc" id="cityDesc">
        <input type="hidden" name="warehouseDesc"
id="warehouseDesc">
    </div>

    <div id="text-address" class="mb-3">
        <label class="form-label">Адреса доставки</label>
        <textarea name="delivery_address_text"
            class="form-control"
            rows="3"
            placeholder="Вкажіть адресу"></textarea>
    </div>

    <div class="mb-3">
        <label class="form-label">Спосіб оплати</label>
        <select name="payment_method" class="form-select"
required>
            <option value="Готівка">Готівка при отриманні</option>
            <option value="Карта">Оплата карткою</option>
            <option value="Приват24">Оплата через Приват24</option>
        </select>
    </div>

    <button type="submit" class="btn btn-success w-100">
        Підтвердити замовлення
    </button>
</form>
</div>
</div>
</div>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap
.bundle.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap-select@1.14.0-
beta3/dist/js/bootstrap-select.min.js"></script>

<script>
$(function() {
    $('#selectpicker').selectpicker();

    const method    = $('#delivery_method'),
        npFields    = $('#np-fields'),
        textAddr    = $('#text-address'),
        areaSel     = $('#np-area'),
        citySel     = $('#np-city'),
        whSel       = $('#np-warehouse'),
        areaDesc    = $('#areaDesc'),
        cityDesc    = $('#cityDesc'),
        whDesc      = $('#warehouseDesc');

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						101
Змн.	Арк.	№ докум.	Підпис	Дата		

```

function toggleNP(){
  if (method.val() === 'Нова Пошта') {
    npFields.show();
    textAddr.hide();
    areaSel.prop('required', true).prop('disabled', false);
    citySel.prop('required', true).prop('disabled', false);
    whSel.prop('required', true).prop('disabled', false);
    $('textarea[name="delivery_address_text"]')
      .prop('required', false)
      .prop('disabled', true);
    loadAreas();
  } else {
    npFields.hide();
    textAddr.show();
    areaSel.prop('required', false).prop('disabled', true);
    citySel.prop('required', false).prop('disabled', true);
    whSel.prop('required', false).prop('disabled', true);
    $('textarea[name="delivery_address_text"]')
      .prop('required', true)
      .prop('disabled', false);
  }
  areaSel.selectpicker('refresh');
  citySel.selectpicker('refresh');
  whSel.selectpicker('refresh');
}

method.on('change', toggleNP);
toggleNP(); // виклик при завантаженні

function loadAreas(){
  fetch('np_api.php', {
    method: 'POST',
    headers: {'Content-Type':'application/json'},
    body: JSON.stringify({action:'getAreas'})
  })
  .then(r=>r.json())
  .then(data=>{
    areaSel.empty().append('<option value="">Оберіть
область</option>');
    data.forEach(a=>{
      areaSel.append(
        `<option value="${a.Ref}" data-
desc="${a.Description}">${a.Description}</option>`
      );
    });
    areaSel.selectpicker('refresh');
    citySel.empty().selectpicker('refresh');
    whSel.empty().selectpicker('refresh');
  });
}

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						102
Змн.	Арк.	№ докум.	Підпис	Дата		

```

areaSel.on('changed.bs.select', function(){
    const ar = $(this).val();
    citySel.empty().append('<option>Завантаження...</option>').select
picker('refresh');
    fetch('np_api.php', {
        method: 'POST',
        headers: {'Content-Type':'application/json'},
        body: JSON.stringify({action:'getCities', areaRef: ar})
    })
    .then(r=>r.json())
    .then(data=>{
        citySel.empty().append('<option value="">Оберіть
місто</option>');
        data.forEach(c=>{
            citySel.append(
                `<option value="${c.Ref}" data-
desc="${c.Description}">${c.Description}</option>`
            );
        });
        citySel.selectpicker('refresh');
        whSel.empty().selectpicker('refresh');
    });
});

citySel.on('changed.bs.select', function(){
    const cr = $(this).val();
    whSel.empty().append('<option>Завантаження...</option>').selectpi
cker('refresh');
    fetch('np_api.php', {
        method: 'POST',
        headers: {'Content-Type':'application/json'},
        body: JSON.stringify({action:'getWarehouses', cityRef: cr})
    })
    .then(r=>r.json())
    .then(data=>{
        whSel.empty().append('<option value="">Оберіть
відділення</option>');
        data.forEach(w=>{
            const lbl = `${w.Description} (${w.Number})`;
            whSel.append(
                `<option value="${w.Description}" data-
desc="${lbl}">${lbl}</option>`
            );
        });
        whSel.selectpicker('refresh');
    });
});

$('form').on('submit', function(){
    if (method.val() === 'Нова Пошта') {

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						103
Змн.	Арк.	№ докум.	Підпис	Дата		

```
        areaDesc.val(areaSel.find('option:selected').data('desc') ||
    '');
        cityDesc.val(citySel.find('option:selected').data('desc') ||
    '');
        whDesc.val(whSel.find('option:selected').data('desc') || '');
    }
    });
});
</script>

<?php include 'includes/footer.php'; ?>
```

					КР.КН 25.593.10.000 ПЗ	Арк.
						104
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Є

Програмний код сторінки адміністратора

```
<?php
// admin_dashboard.php

session_start();
include 'includes/db.php';

if (empty($_SESSION['role']) || $_SESSION['role'] !== 'admin') {
    header("Location: /nail/profile.php");
    exit();
}

$table = $_GET['table'] ?? null;
$action = $_REQUEST['action'] ?? null;
$id = isset($_REQUEST['id']) ? (int)$_REQUEST['id'] : null;

function bind_type($colType) {
    if (strpos($colType, 'int') !== false) return 'i';
    if (strpos($colType, 'double') !== false ||
        strpos($colType, 'float') !== false) return 'd';
    return 's';
}

if ($table && $action === 'delete' && $id) {
    $conn->query("DELETE FROM `table` WHERE id = $id");
    header("Location: admin_dashboard.php?table=" .
urlencode($table));
    exit();
}

if ($table && $_SERVER['REQUEST_METHOD'] === 'POST' && $action ===
'add') {
    $cols = $conn->query("DESCRIBE `table`");
    $fields = [];
    $placeholders = [];
    $values = [];
    $types = '';

    while ($c = $cols->fetch_assoc()) {
        if ($c['Extra'] === 'auto_increment') continue;
        $fields[] = "`{$c['Field']}`";
        $placeholders[] = '?';
        $types .= bind_type($c['Type']);
        $values[] = $_POST[$c['Field']] ?? null;
    }
}
```

					КР.КН 25.593.10.000 ПЗ	Арк.
						105
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    $sql = "INSERT INTO ` $tbl ` ( " . implode(',', $fields) . " )
            VALUES ( " . implode(',', $placeholders) . " )";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param($types, ...$values);
    $stmt->execute();

    header("Location: admin_dashboard.php?table=" .
urlencode($tbl));
    exit();
}

$record = null;
if ($tbl && $_SERVER['REQUEST_METHOD'] === 'POST' && $action ===
'edit' && $id) {
    $cols = $conn->query("DESCRIBE ` $tbl `");
    $sets = [];
    $values = [];
    $types = '';

    while ($c = $cols->fetch_assoc()) {
        if ($c['Extra'] === 'auto_increment') continue;
        $sets[] = "`{$c['Field']}` = ?";
        $types .= bind_type($c['Type']);
        $values[] = $_POST[$c['Field']] ?? null;
    }

    $types .= 'i';
    $values[] = $id;

    $sql = "UPDATE ` $tbl ` SET " . implode(',', $sets) . " WHERE id
= ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param($types, ...$values);
    $stmt->execute();

    header("Location: admin_dashboard.php?table=" .
urlencode($tbl));
    exit();
}

if ($tbl && $action === 'edit' && $id) {
    $record = $conn->query("SELECT * FROM ` $tbl ` WHERE id = $id")
->fetch_assoc();
}

include 'includes/header.php';
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						106
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<meta name="viewport" content="width=device-width,initial-
scale=1">
<title>Адмінпанель – KENTAFURIK</title>
<link href="/nail/assets/css/admin_style.css" rel="stylesheet">
</head>
<body>
<div class="container my-5">
<div class="d-flex justify-content-between align-items-center
mb-4">
<h1>Адмінпанель</h1>
<a href="logout.php" class="btn btn-outline-danger">Вийти</a>
</div>

<?php if (!$tbl):
    $tables = [];
    $res = $conn->query("SHOW TABLES");
    while ($r = $res->fetch_array()) {
        $tables[] = $r[0];
    }
?>
<h2>Оберіть таблицю:</h2>
<ul class="list-group mb-5">
<?php foreach ($tables as $t): ?>
<li class="list-group-item">
<a href="?table=?= urlencode($t) ??"><?=
htmlspecialchars($t) ?></a>
</li>
<?php endforeach; ?>
</ul>
<?php exit; ?>
<?php endif; ?>

<a href="admin_dashboard.php" class="btn btn-link mb-3">&larr;
До списку таблиць</a>
<h2 class="mb-4">Таблиця: <?= htmlspecialchars($tbl) ?></h2>

<?php
$res = $conn->query("SELECT * FROM `{$tbl}`");
if ($res && $res->num_rows):
?>
<table class="table table-striped table-bordered mb-4">
<thead class="table-light">
<tr>
<?php foreach ($res->fetch_fields() as $f): ?>
<th><?= htmlspecialchars($f->name) ?></th>
<?php endforeach; ?>
<th>Дії</th>
</tr>
</thead>
<tbody>
<?php

```

					КР.КН 25.593.10.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		107

```

$res->data_seek(0);
while ($row = $res->fetch_assoc()):
?>
    <tr>
        <?php foreach ($row as $col => $val): ?>
            <td>
                <?php
                    $ext = strtolower(pathinfo($val,
PATHINFO_EXTENSION));
                    if ((stripos($col,'image')!==false ||
in_array($ext, ['jpg','jpeg','png','gif'])) && $val):
                        echo "<img src=\"\" . htmlspecialchars($val) .
\"\" class=\"img-thumbnail\" style=\"max-width:80px;\"><br>";
                    endif;
                    echo htmlspecialchars($val);
                ?>
            </td>
            <?php endforeach; ?>
            <td>
                <a
                    href="?table=<?= urlencode($tbl)
?>&action=edit&id=<?= $row['id'] ?>"
                    class="btn btn-sm btn-outline-primary"
                >Редагувати</a>
                <a
                    href="?table=<?= urlencode($tbl)
?>&action=delete&id=<?= $row['id'] ?>"
                    class="btn btn-sm btn-outline-danger"
                    onclick="return confirm('Ви впевнені?')"
                >Видалити</a>
            </td>
        </tr>
    <?php endwhile; ?>
</tbody>
</table>
<?php else: ?>
    <p>Немає записів у таблиці.</p>
<?php endif; ?>

<?php $isEdit = ($action === 'edit' && $record); ?>
<div class="card p-4 mb-5">
    <h3 class="mb-3"><?= $isEdit ? "Редагувати запис
#{$record['id']}" : "Додати запис" ?></h3>
    <form method="POST">
        <input type="hidden" name="action" value="<?= $isEdit ?
'edit' : 'add' ?>">
        <?php if ($isEdit): ?>
            <input type="hidden" name="id" value="<?= $record['id']
?>">
        <?php endif; ?>

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						108
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<?php
$cols = $conn->query("DESCRIBE ` $tbl `");
while ($c = $cols->fetch_assoc()):
    if ($c['Extra'] === 'auto_increment') continue;
    $f = $c['Field'];
    $v = $isEdit ? $record[$f] : '';
?>
    <div class="mb-3">
        <label class="form-label"><?= htmlspecialchars($f)
?></label>
        <input
            type="text"
            name="<?= htmlspecialchars($f) ?>"
            class="form-control"
            value="<?= htmlspecialchars($v) ?>"
            required
        >
    </div>
<?php endwhile; ?>

    <button class="btn btn-<?= $isEdit ? 'primary' : 'success'
?>">
        <?= $isEdit ? 'Оновити' : 'Додати' ?>
    </button>
</form>
</div>
</div>
</body>
</html>

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						109
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Ж

Програмний код стилю сайту

```
:root {
  --color-bg: #f8f3f1;
  --color-primary: #d8a7b1;
  --color-secondary: #a3c4f3;
  --color-accent: #c6a07a;
  --color-text: #5a4d4b;
  --color-muted: #7a6d6b;
}

body {
  font-family: 'Quicksand', sans-serif;
  background-color: var(--color-bg);
  color: var(--color-text);
  margin: 0;
  padding: 0;
}

a {
  text-decoration: none;
  color: inherit;
}

.navbar {
  padding: 0.8rem 0;
  background-color: #fff;
  border-bottom: 1px solid var(--color-accent);
  font-weight: 500;
}

.navbar-brand {
  font-weight: 700;
  font-size: 1.5rem;
  color: var(--color-primary) !important;
}

.nav-link {
  padding: 0.5rem 1rem;
  color: var(--color-text) !important;
  transition: color 0.3s;
}

.nav-link:hover {
  color: var(--color-primary) !important;
}

.nav-item .nav-link {
  position: relative;
}

.nav-item .nav-link::after {
  content: '';
```

					КР.КН 25.593.10.000 ПЗ	Арк.
						110
Змн.	Арк.	№ докум.	Підпис	Дата		

```

position: absolute;
bottom: 5px;
left: 0;
width: 0;
height: 2px;
background-color: var(--color-primary);
transition: width 0.3s;
}
.nav-item .nav-link:hover::after {
width: 100%;
}

.modal-content {
border-radius: 16px;
border: none;
box-shadow: 0 8px 24px rgba(0, 0, 0, 0.05);
}
.modal-header {
border-bottom: none;
}
.modal-title {
font-size: 1.5rem;
font-weight: 600;
color: var(--color-primary);
}

.category-card {
border: 1px solid var(--color-accent);
border-radius: 12px;
padding: 10px;
background: #fff;
transition: background 0.3s, box-shadow 0.3s;
height: 100%;
display: flex;
flex-direction: column;
justify-content: center;
}
.category-card:hover {
background: #fdeef1;
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
}
.category-card img {
max-width: 100%;
max-height: 120px;
object-fit: contain;
display: block;
margin: 0 auto 10px;
border-radius: 8px;
}

.banner-image {

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						111
Змн.	Арк.	№ докум.	Підпис	Дата		

```

width: 100%;
max-height: 50vh;
object-fit: cover;
border-radius: 8px;
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
margin-bottom: 2rem;
}

.card {
background: #fff;
border: none;
border-radius: 12px;
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
transition: transform 0.3s, box-shadow 0.3s;
}

.card:hover {
transform: translateY(-5px);
box-shadow: 0 8px 24px rgba(0, 0, 0, 0.1);
}

.card-img-wrapper {
height: 180px;
overflow: hidden;
display: flex;
align-items: center;
justify-content: center;
background: #fff;
}

.card-img-wrapper .card-img-top {
max-width: 100%;
max-height: 100%;
object-fit: contain;
transition: transform 0.3s;
}

.card-img-wrapper .card-img-top:hover {
transform: scale(1.05);
}

.card-title {
font-size: 1.1rem;
color: var(--color-text);
}

.card-text {
color: var(--color-muted);
}

.product-img {
transition: transform 1.9s ease;
}

.product-img:hover {
transform: rotate(360deg);
}

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						112
Змн.	Арк.	№ докум.	Підпис	Дата		

```

.btn-primary {
  background-color: var(--color-primary);
  border: none;
  color: #fff;
  transition: background 0.3s;
}
.btn-primary:hover {
  background-color: #b0788c;
}
.btn-success {
  background-color: var(--color-secondary);
  border: none;
  color: #fff;
  transition: background 0.3s;
}
.btn-success:hover {
  background-color: #7fa1da;
}
.btn-outline-primary {
  color: var(--color-primary);
  border: 1px solid var(--color-primary);
  transition: background 0.3s, color 0.3s;
}
.btn-outline-primary:hover {
  background-color: var(--color-primary);
  color: #fff;
}

.alert {
  margin-bottom: 20px;
  padding: 12px 20px;
  border-radius: 6px;
  font-size: 0.95rem;
}
.alert-success {
  color: #155724;
  background-color: #e3f2fd;
  border: 1px solid #b3d6f2;
}
.alert-danger,
.alert-error {
  color: #842029;
  background-color: #fde2e2;
  border: 1px solid #f5c2c7;
}

input.form-control,
textarea.form-control {
  border: 1px solid #ccc;
  border-radius: 6px;
  background: #fff;
}

```

					КР.КН 25.593.10.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		113

```

}

@media (max-width: 768px) {
  .category-card img { max-height: 100px; }
  .modal-title { font-size: 1.2rem; }
  .nav-link { padding: 0.4rem 0.8rem; }
}
#suggestions .list-group-item:hover {
  background-color: var(--color-accent);
  color: #fff;
}
#suggestions .suggestion-img {
  width: 40px;
  height: 40px;
  object-fit: cover;
  border-radius: 4px;
}
#suggestions {
  position: absolute;
  top: 100%;
  left: 0;
  right: 0;
  z-index: 1050;
  display: none;
  background: #fff;
  border: 1px solid #ddd;
  border-top: none;
  max-height: 220px;
  overflow-y: auto;
}

#suggestions .list-group-item {
  cursor: pointer;
  padding: 0.5rem 0.75rem;
}

#suggestions .list-group-item:hover {
  background-color: var(--color-accent);
  color: #fff;
}

```

					КР.КН 25.593.10.000 ПЗ	Арк.
						114
Змн.	Арк.	№ докум.	Підпис	Дата		