

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

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

на тему:

**«Система електронної комерції для реалізації
авторських виробів»**

Виконав студент групи ЗПЗ-22-1
ТРОФІМОВА Марія Олександрівна

Керівник роботи:
ГОДЛЕВСЬКИЙ Юрій Олександрович

Рецензент:
ПОНТКІВСЬКИЙ Владислав Ігорович

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

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії програмного
забезпечення

Тетяна Вакалюк

«27» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ТРОФІМОВА Марія Олександрівна**

Керівник роботи: **ГОДЛЕВСЬКИЙ Юрій Олександрович**

Тема роботи: **«Система електронної комерції для реалізації авторських виробів»**
затверджена Наказом закладу вищої освіти від «27» лютого 2026 р., №93/с

Вихідні дані для роботи: **процеси електронної комерції, управління товарами, замовленнями та інформаційним наповненням у сфері реалізації авторських-виробів; предмет дослідження — методи проєктування веборієнтованих систем електронної комерції, архітектура клієнт-серверних застосунків, реляційні бази даних (MySQL) та сучасні програмні засоби розробки вебсистем.**

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Годлевський Ю.О.	01.03.2026р.	01.03.2026р
2	Годлевський Ю.О.	01.04.2026р	01.04.2026р
3	Годлевський Ю.О.	01.05.2026р	01.05.2026р

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра на тему «Система електронної комерції для реалізації авторських виробів» складається з вебсистеми інтернет-магазину handmade-свічок з адміністративною панеллю та пояснювальної записки. Пояснювальна записка містить 86 сторінок, 29 ілюстрацій, 17 таблиць, 8 додатків та 31 використане джерело.

Об'єктом дослідження є процеси автоматизації електронної комерції у сфері реалізації handmade-продукції. Предметом дослідження є методи та засоби проєктування й розробки веборієнтованих систем електронної комерції.

Метою роботи є розробка вебсистеми інтернет-магазину handmade-свічок з адміністративною панеллю, що забезпечує автоматизацію управління товарами, замовленнями, користувачами та інформаційним наповненням сайту.

У процесі розробки використано React, Node.js, Express.js, MySQL, Sequelize та технологію JWT-автентифікації. Реалізовано клієнтську частину інтернет-магазину, систему реєстрації та авторизації користувачів, особистий кабінет покупця, систему керування товарами, замовленнями, відгуками, категоріями та інформаційним контентом. Окрім цього, впроваджено механізм динамічного розрахунку вартості товару залежно від обраних характеристик, систему керування користувачами та адаптивну адміністративну панель.

Результати роботи можуть бути використані для автоматизації діяльності малого бізнесу, що займається виготовленням та продажем handmade-свічок і супутньої декоративної продукції.

КЛЮЧОВІ СЛОВА: АДМІНІСТРАТИВНА ПАНЕЛЬ, HANDMADE-БІЗНЕС, ІНТЕРНЕТ-МАГАЗИН, REST API, ВЕБСИСТЕМА.

					ІПЗ.КР.Б – 121 – 16 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Трофімова М.О.				«Система електронної комерції для реалізації авторських виробів»	Літ.	Арк.	Аркуші
Керівник	Годлевський Ю.О.						3	86
Рецензент	Піонтківський В. І.					ФІКТ, Гр. ЗІПЗ-22-1		
Зав. каф.	Вакалюк Т. А.							

ABSTRACT

The bachelor's qualification thesis entitled "E-Commerce System for the Sale of Handmade Products" consists of a web-based online store for handmade candles with an administrative panel and an explanatory report. The explanatory report comprises 86 pages, 29 figures, 17 tables, 8 appendices, and 31 references.

The object of the research is the automation processes of e-commerce in the field of handmade product sales. The subject of the research is the methods and tools for designing and developing web-oriented e-commerce systems.

The purpose of the thesis is to develop a web-based online store system for handmade candles with an administrative panel that automates the management of products, orders, users, and website content.

During the development process, React, Node.js, Express.js, MySQL, Sequelize, and JWT authentication technology were used. The implemented system includes the client-side of the online store, user registration and authentication, a customer account, and management modules for products, orders, reviews, categories, and website content. In addition, a dynamic pricing mechanism based on selected product characteristics, a user management system, and a responsive administrative panel were implemented.

The results of this work can be used to automate the activities of small businesses engaged in the production and sale of handmade candles and related decorative products.

KEYWORDS: ADMINISTRATIVE PANEL, HANDMADE BUSINESS, ONLINE STORE, REST API, WEB SYSTEM.

					<i>ІПЗ.КР.Б – 121 – 16 – ПЗ</i>	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Зміст

ВСТУП	7
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ТА ОБҐРУНТУВАННЯ ПРОЄКТУ	10
1.1. Постановка задачі	10
1.2. Аналіз предметної області	12
1.3. Перелік задач, що підлягають автоматизації	16
1.4. Аналіз існуючих рішень для реалізації вебсистем електронної комерції	19
1.5. Обґрунтування доцільності власної розробки	22
Висновки до розділу 1	26
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБСИСТЕМИ ІНТЕРНЕТ-МАГАЗИНУ HANDMADE-СВІЧОК	27
2.1. Визначення вимог та варіантів використання системи	27
2.2. Проєктування бази даних системи	30
2.3. Проєктування алгоритмів роботи системи	35
2.4. Реалізація програмного комплексу	39
2.5. Структура програмного комплексу	42
Висновки до розділу 2	46
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБСИСТЕМОЮ ІНТЕРНЕТ-МАГАЗИНУ HANDMADE-СВІЧОК	47
3.1. Встановлення та налаштування системи	47
3.2. Структура та основні елементи інтерфейсу	50
3.3. Тестування програмного комплексу	64
3.4. Перевірка безпеки вебсистеми	68
Висновки до розділу 3	72
ВИСНОВКИ	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	75
ДОДАТКИ	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – прикладний програмний інтерфейс (Application Programming Interface).

JWT – JSON Web Token — механізм автентифікації та передачі захищених даних.

REST API – архітектурний стиль взаємодії між клієнтом і сервером через HTTP-запити (Representational State Transfer).

БД – база даних.

СУБД – система управління базами даних.

ПЗ – програмне забезпечення.

HTTP – протокол передачі гіпертекстових повідомлень (HyperText Transfer Protocol).

HTTPS – захищений протокол передачі гіпертекстових повідомлень (HyperText Transfer Protocol Secure).

ORM – об'єктно-реляційне відображення (Object-Relational Mapping).

CRUD – базові операції з даними: створення, читання, оновлення, видалення (Create, Read, Update, Delete).

MVC – архітектурний шаблон «модель — вигляд — контролер» (Model–View–Controller).

JSON – текстовий формат обміну даними (JavaScript Object Notation).

UI – користувацький інтерфейс (User Interface).

UX – досвід взаємодії користувача із системою (User Experience).

UML – уніфікована мова моделювання (Unified Modeling Language).

CORS – механізм спільного використання ресурсів між різними джерелами (Cross-Origin Resource Sharing).

SQL – мова структурованих запитів (Structured Query Language).

XSS – міжсайтовий скриптинг (Cross-Site Scripting).

SaaS – програмне забезпечення як послуга (Software as a Service).

CMS – система керування вмістом (Content Management System).

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						6
Змн.	Арк.	№ докum.	Підпис	Дата		

ВСТУП

Сучасний розвиток ринку товарів ручної роботи в Україні супроводжується активним переходом майстрів handmade-виробництва у цифровий простір. Зростання доступності інтернету, поширення смартфонів та зміна купівельних звичок сприяли тому, що значна частина споживачів сьогодні надає перевагу онлайн-шопінгу навіть при виборі авторських виробів, зокрема ароматичних і декоративних свічок. Для власників малого handmade-бізнесу це означає необхідність забезпечення присутності в інтернеті не лише у форматі сторінок у соціальних мережах, а й у вигляді повноцінного інтернет-магазину з можливістю перегляду каталогу, оформлення замовлень та відстеження їхнього стану.

Разом із тим специфіка handmade-бізнесу суттєво відрізняється від традиційної роздрібної торгівлі. Власник такого підприємства, як правило, одночасно виконує функції виробника, менеджера замовлень, контент-менеджера та продавця. За таких умов автоматизація рутинних процесів набуває критичного значення: кожна година, витрачена на ручне опрацювання замовлень чи оновлення асортименту, це час, вилучений із безпосереднього виробництва. Саме тому інструменти управління магазином повинні бути максимально простими, зрозумілими та доступними з будь-якого пристрою, зокрема зі смартфона.

Аналіз доступних на ринку рішень для електронної комерції свідчить про те, що універсальні SaaS-платформи та CMS-системи не повністю задовольняють потреби handmade-магазину. Вони або надають надлишковий функціонал, що ускладнює щоденну роботу, або не підтримують специфічну бізнес-логіку, притаманну продажу авторських виробів із великою кількістю індивідуальних характеристик і динамічним ціноутворенням. Це обумовлює актуальність розробки спеціалізованої вебсистеми, орієнтованої безпосередньо на потреби handmade-магазину.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єктом дослідження є процеси автоматизації електронної комерції у сфері реалізації handmade-продукції, зокрема управління каталогом товарів, оброблення замовлень, взаємодія з покупцями та адміністрування інформаційного наповнення вебресурсу.

Предметом дослідження є методи та засоби проєктування й розробки веборієнтованих систем електронної комерції з використанням сучасного стеку вебтехнологій — React, Node.js, Express.js, MySQL та Sequelize.

Методи дослідження. Для досягнення поставленої мети використано методи системного аналізу предметної області, об'єктно-орієнтованого проєктування, моделювання за допомогою UML, методи проєктування реляційних баз даних, технології клієнт-серверної взаємодії, а також методи функціонального тестування програмного забезпечення.

Метою роботи є розробка вебсистеми інтернет-магазину handmade-свічок з адміністративною панеллю, яка забезпечує автоматизацію управління товарами, замовленнями, користувачами та інформаційним наповненням сайту.

Для досягнення поставленої мети необхідно вирішити такі завдання: провести аналіз предметної галузі та визначити особливості функціонування handmade-магазину, що потребують автоматизації, дослідити існуючі рішення для реалізації вебсистем електронної комерції; спроектувати структуру бази даних та архітектуру програмного комплексу, реалізувати клієнтську частину інтернет-магазину з підтримкою кастомізації товарів і динамічного ціноутворення, розробити адміністративну панель для централізованого управління системою, провести функціональне тестування та перевірку безпеки вебсистеми.

Практичне значення роботи полягає у створенні готової до застосування вебсистеми, яка може бути використана для автоматизації діяльності малого бізнесу з виготовлення та продажу handmade-свічок і супутньої декоративної продукції.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Структура роботи. Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі проведено системний аналіз предметної галузі та обґрунтовано доцільність власної розробки. У другому розділі описано проєктування та реалізацію вебсистеми. У третьому розділі наведено опис інтерфейсу системи, результати тестування та перевірки безпеки.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ТА ОБҐРУНТУВАННЯ ПРОЄКТУ

1.1. Постановка задачі

Розробка вебсистеми електронної комерції для handmade-магазину є комплексним завданням, що охоплює проєктування клієнтської частини, серверної логіки, бази даних та адміністративного інтерфейсу. На відміну від типових інтернет-магазинів із фіксованим асортиментом, магазин handmade-свічок потребує підтримки процесів, пов'язаних з індивідуальним виготовленням продукції, вибором характеристик товару та подальшим супроводом замовлення.

Основною особливістю предметної галузі є поєднання двох типів товарів: готових виробів і продукції, що виготовляється під замовлення. Для товарів у наявності важливим є відображення актуального залишку та можливість швидкого оформлення замовлення. Для виробів під замовлення необхідно передбачити вибір кольору, аромату та декоративних елементів із переліку доступних варіантів. Поєднання цих сценаріїв в одній системі потребує продуманої структури бази даних і зручного клієнтського інтерфейсу.

Окремим завданням є реалізація динамічного ціноутворення. Оскільки окремі характеристики виробу можуть впливати на його собівартість, кінцева ціна товару не завжди є фіксованою. Вона має розраховуватися автоматично залежно від параметрів, обраних покупцем. Такий підхід дає змогу відразу показувати покупцеві актуальну вартість конкретної конфігурації товару та зменшує потребу в додаткових уточненнях щодо ціни.

Клієнтська частина вебсистеми повинна забезпечувати перегляд каталогу продукції з розподілом за категоріями та підкатегоріями, відкриття сторінки окремого товару, перегляд опису й фотографій, вибір характеристик, додавання товарів до кошика та оформлення замовлення. При оформленні замовлення покупець зазначає контактні дані, адресу доставки та, за потреби, коментар. Оформлення замовлення має бути доступним не лише для зареєстрованих

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

користувачів, а й для відвідувачів без облікового запису, оскільки примусова реєстрація перед першою покупкою може ускладнювати взаємодію з магазином.

Для зареєстрованих користувачів передбачається особистий кабінет. У ньому покупець може переглядати власні дані, редагувати контактну інформацію та відстежувати історію замовлень із поточними статусами. Такий функціонал спрощує повторні покупки та дає користувачеві доступ до інформації про виконання замовлення без необхідності додаткового звернення до продавця.

Адміністративна панель є другою важливою частиною системи. Вона повинна забезпечувати власнику магазину можливість керувати товарами, категоріями, підкатегоріями, тематичними колекціями, матеріалами, замовленнями, користувачами, відгуками та інформаційним наповненням сайту. Для товарів необхідно передбачити створення, редагування, завантаження кількох зображень, призначення головного фото та зміну статусу активності без видалення запису з бази даних.

Окрему роль в адміністративній частині відіграє керування матеріалами та характеристиками продукції. Адміністратор повинен мати можливість змінювати перелік доступних кольорів, ароматів і декоративних елементів, а також задавати цінову надбавку для кожного з них. Якщо певний матеріал тимчасово недоступний, його можна приховати з переліку опцій для покупця без видалення з бази даних. Це дозволяє підтримувати актуальність інформації в каталозі без внесення змін до програмного коду.

Робота із замовленнями передбачає перегляд їхнього складу, контактних даних покупця, обраних характеристик товарів і загальної вартості. Адміністратор повинен мати можливість змінювати статус замовлення відповідно до етапу його виконання: від нового замовлення до обробки, готовності, відправки, завершення або скасування. Також адміністративна панель має підтримувати модерацію відгуків, керування банерами головної сторінки, статтями блогу та розділом поширених запитань.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

До нефункціональних вимог належить адаптивність клієнтської та адміністративної частин системи. Інтерфейс має коректно відображатися на пристроях із різними розмірами екрана, оскільки покупці часто переглядають товари зі смартфонів, а власник магазину може керувати замовленнями та асортиментом не лише з комп'ютера. Для реалізації клієнтської частини обрано React, компонентна модель якого підходить для побудови адаптивних інтерфейсів [8, 30].

З погляду безпеки вебсистема повинна забезпечувати захист персональних даних користувачів і розмежування прав доступу між покупцем та адміністратором. Для цього передбачено зберігання паролів у хешованому вигляді, використання JWT-автентифікації та перевірку доступу до захищених адміністративних маршрутів на серверному рівні.

Для реалізації поставлених завдань необхідно спроектувати реляційну базу даних із визначеними зв'язками між сутностями, розробити серверну частину на платформі Node.js з REST API та реалізувати клієнтську частину засобами React. Взаємодія між клієнтською і серверною частинами здійснюється через REST API, що забезпечує незалежність компонентів і спрощує подальший розвиток системи [24].

Постановка задачі передбачає створення вебсистеми електронної комерції, яка поєднує зручний клієнтський інтерфейс, адміністративну панель, підтримку індивідуальних характеристик handmade-товарів, динамічний розрахунок вартості та базові механізми захисту даних. Результатом роботи має стати вебсистема, придатна для практичного використання в діяльності малого крафтового виробництва.

1.2. Аналіз предметної області

Предметною галуззю дослідження є діяльність інтернет-магазину авторських свічок ручної роботи, який реалізує власну продукцію через мережу Інтернет. Для визначення вимог до програмного комплексу необхідно

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

розглянути специфіку такого бізнесу, його учасників та основні операційні процеси, що потребують автоматизації.

Handmade-бізнес у сфері виготовлення свічок є типовим представником малого авторського виробництва. Асортимент подібних магазинів включає декоративні та ароматичні свічки різних форм і розмірів — формові (торси, колони, куби), контейнерні (у скляних та гіпсових кашпо), свічки з вощини, а також супутні товари: аромасаше, гіпсові підставки та інші декоративні вироби. Кожен продукт є унікальним за своєю природою, оскільки при його виготовленні враховуються побажання конкретного покупця — обраний аромат, колір воску або зовнішнє оформлення з декоративними елементами. Неповторність виробів є однією з основних переваг handmade-продукції для споживача, проте одночасно вона суттєво ускладнює організацію онлайн-продажів порівняно зі стандартними інтернет-магазинами.

Основними учасниками бізнес-процесу є власник магазину та покупці. Власник одночасно виконує функції виробника, менеджера замовлень, контент-менеджера та продавця. Багатофункціональність власника є однією з основних причин необхідності автоматизації. Без спеціалізованого програмного інструментарію обробка кожного замовлення вимагає від власника значних витрат часу на уточнення параметрів, виставлення рахунку, відстеження оплати та інформування покупця про готовність і відправку замовлення. Вебсистема повинна максимально спростити цей ланцюжок та перекласти частину операцій на самого покупця — зокрема, вибір характеристик і формування замовлення.

Центральним елементом предметної галузі є каталог товарів. Він повинен бути організований таким чином, щоб покупець міг швидко знайти потрібний тип виробу, не переглядаючи весь асортимент. Для цього каталог структурується за категоріями — формові свічки, контейнерні свічки, вощинні свічки, аромасаше, підставки. Кожна категорія може містити підкатегорії для детальнішого розподілу продукції. Наприклад, формові свічки поділяються на торси, колони та сезонні вироби, а під окремою рубрикою виділяються тематичні

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

колекції до певних свят або подій. Така ієрархічна структура суттєво спрощує навігацію та скорочує час пошуку потрібного товару [1, 7].

Однією з особливостей предметної галузі, що відрізняє handmade-магазин від звичайного онлайн-ритейлу, є підтримка індивідуального налаштування характеристик кожного виробу. Під час оформлення замовлення покупець може обирати колір воску або забарвлення свічки з переліку доступних варіантів, а також декоративні елементи. При цьому кожна характеристика може впливати на кінцеву вартість виробу: певні аромати або рідкісні декоративні елементи мають вищу собівартість, що відображається у вигляді цінової надбавки. Система повинна автоматично враховувати ці надбавки при розрахунку підсумкової вартості обраної конфігурації.

Важливим аспектом предметної галузі є поділ асортименту на два типи продукції. Перший тип — це вироби, що виготовляються виключно під замовлення. Такі товари не мають фіксованого складського залишку, а строк виготовлення може становити від кількох днів до кількох тижнів залежно від складності та завантаженості виробника. Другий тип — це готові вироби, що вже перебувають у наявності. Для них актуальним є контроль залишку та оперативне оформлення замовлення без тривалого очікування. Ці два типи мають різну логіку відображення в каталозі та різний сценарій оформлення замовлення, що вимагає окремого опрацювання при програмній реалізації.

Процес обробки замовлень є найбільш операційно насиченим у діяльності handmade-магазину. Після того як покупець оформив замовлення через сайт, власник магазину повинен його побачити, проаналізувати склад і обрані параметри, підтвердити прийняття до виробництва, відстежити процес виготовлення, зафіксувати факт готовності, здійснити відправку та повідомити покупця. Усі ці стани повинні відображатися через систему статусів замовлення, яка є невід'ємною частиною програмного комплексу. Покупець, у свою чергу, повинен мати можливість бачити поточний статус свого замовлення в особистому кабінеті, не звертаючись щоразу до власника за уточненнями [9].

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Управління матеріалами та доступністю характеристик є окремим напрямком предметної галузі. Наявність певних ароматів, кольорів або декоративних елементів у виробника може змінюватися: деякі позиції закупаються обмеженими партіями і можуть тимчасово бути відсутні. Система повинна дозволяти адміністратору оперативно приховувати недоступні матеріали з переліку опцій для покупців, не видаляючи їх із бази даних. Це гарантує, що покупець не зможе обрати характеристику, яку неможливо виготовити, і знижує кількість ситуацій, коли власник магазину змушений відмовляти в замовленні або домовлятися про заміну параметрів.

Управління інформаційним наповненням сайту є ще однією складовою предметної галузі. Банери головної сторінки слугують для анонсу нових колекцій та акцій. Блог дозволяє публікувати корисні матеріали про догляд за свічками, ідеї для подарунків або інформацію про нові аромати. Розділ поширених запитань допомагає покупцям самостійно отримати відповіді на типові питання щодо замовлення, оплати та доставки без необхідності звертатися до продавця. Усі ці інструменти є невід'ємною частиною сучасного інтернет-магазину і повинні управлятися безпосередньо через адміністративну панель без залучення технічних спеціалістів.

Аналіз предметної галузі також вимагає врахування поведінки цільової аудиторії. Покупці handmade-свічок здебільшого шукають унікальний подарунок або предмет декору з можливістю персоналізації. Вони цінують якісне візуальне представлення товарів, детальний опис складу та ароматичного профілю, а також прозорість щодо строків виготовлення. Значна частина покупок здійснюється через мобільні пристрої, що робить адаптивність інтерфейсу базовою вимогою до системи. Зручна навігація, простота оформлення замовлення та коректне відображення на всіх типах пристроїв безпосередньо впливають на конверсію відвідувачів у покупців.

Таким чином, аналіз предметної галузі дозволив виявити ключові особливості handmade-бізнесу у сфері реалізації авторських свічок, визначити

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

основні бізнес-процеси та сформулювати специфічні вимоги до програмного комплексу. Отримані результати склали основу для подальшого формування вимог до системи, вибору технологічного стеку та проєктування архітектури програмного комплексу.

1.3. Перелік задач, що підлягають автоматизації

Визначення переліку задач, що підлягають автоматизації, є необхідним кроком між аналізом предметної галузі та формуванням технічних вимог до системи. На цьому етапі операційні процеси магазину розглядаються з точки зору їхньої придатності до програмної реалізації та доцільності автоматизації з огляду на потреби власника малого бізнесу.

Управління каталогом товарів є одним із найбільш трудомістких щоденних процесів в роботі handmade-магазину. Власник повинен своєчасно додавати нові позиції до асортименту, вносити зміни до описів, коригувати ціни залежно від зміни вартості матеріалів, оновлювати фотографії та контролювати актуальність представлення кожного товару на сайті. У разі тимчасової відсутності певної позиції необхідна можливість приховати товар без його видалення з бази даних, зберігши таким чином усю супутню інформацію. Автоматизація цього процесу передбачає реалізацію зручного інтерфейсу для створення та редагування товарних карток із підтримкою завантаження кількох зображень, призначення товарів до відповідних категорій і підкатегорій, а також оперативної зміни статусу відображення на сайті.

Управління характеристиками товарів і матеріалами виокремлено в окремий напрямок автоматизації через важливість цього процесу для підтримки коректного відображення опцій кастомізації. Перелік доступних кольорів, ароматів і декоративних елементів не є статичним, він змінюється залежно від наявності матеріалів, сезону та поновлення асортименту постачальників. Система повинна дозволяти адміністратору в будь-який момент додати новий матеріал із визначенням його типу та можливої цінової надбавки, змінити

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

існуючий запис або позначити матеріал як недоступний. Якщо певний аромат тимчасово відсутній, він не повинен відображатись у переліку.

Автоматизація процесу оформлення та обробки замовлень є центральним завданням системи. На клієнтській стороні цей процес охоплює послідовність дій від додавання товарів із обраними характеристиками до кошика до заповнення форми із контактними даними та підтвердження замовлення. Система повинна зберігати вміст кошика між сеансами браузера, щоб покупець не втрачав підібраний асортимент при закритті вкладки або переході на іншу сторінку. Після відправлення форми всі дані замовлення: склад, обрані характеристики кожного товару, загальна вартість, контактні дані покупця та адреса доставки, автоматично зберігаються в базі даних, а нове замовлення стає доступним адміністратору. На стороні адміністратора автоматизація передбачає зручний перегляд повного переліку замовлень із фільтрацією за статусом, можливість розгортання деталей кожного замовлення та зміну статусу виконання. Система статусів відображає повний життєвий цикл замовлення і дозволяє власнику контролювати процес виробництва та доставки.

Автоматизація управління матеріалами доповнюється завданням контролю доступності опцій у клієнтській частині. Коли адміністратор позначає певний матеріал як недоступний, зміна негайно відображається у переліку варіантів вибору для покупців, жодне ручне втручання у програмний код для цього не вимагається. Це принципово важливо для підтримки актуальності інформації на сайті без залучення розробників.

Управління обліковими записами користувачів передбачає автоматизацію реєстрації, авторизації та контролю доступу. Реєстрація нових покупців відбувається самостійно через форму на сайті. Система автоматично перевіряє унікальність електронної пошти, хешує пароль та створює обліковий запис. Після авторизації покупець отримує персоналізований доступ до особистого кабінету. Адміністратор, у свою чергу, може переглядати список зареєстрованих

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

користувачів та за необхідності блокувати доступ до системи для конкретного облікового запису, що є важливим інструментом захисту від зловживань.

Автоматизація управління відгуками вирішує завдання збору та модерації зворотного зв'язку від покупців. Система повинна дозволяти покупцям залишати оцінки та текстові коментарі до товарів, а адміністратору — переглядати нові відгуки та приймати рішення щодо їхньої публікації. Наявність системи модерації запобігає автоматичній появі небажаного контенту на сторінках товарів та дозволяє підтримувати актуальну і достовірну колекцію відгуків, що підвищує довіру потенційних покупців.

Управління інформаційним наповненням сайту банерами, блогом та розділом поширених запитань, автоматизується шляхом надання адміністратору зручного інтерфейсу для редагування цього контенту без необхідності змінювати програмний код. Власник магазину може самостійно оновлювати банер головної сторінки перед запуском нової колекції, додавати статті блогу та коригувати перелік відповідей на поширені запитання відповідно до нових умов роботи.

Нарешті, система формує базову статистику в рамках дашборду адміністративної панелі. Власник магазину отримує зведену інформацію про загальну кількість товарів у каталозі, замовлень за весь час, зареєстрованих користувачів та відгуків. Ці дані відображаються на головній сторінці адміністративної панелі та дають оперативне уявлення про стан бізнесу без необхідності самостійно підраховувати показники. Автоматизація формування таких зведень заощаджує час та мінімізує можливість помилок при ручному підрахунку.

Додатково важливим завданням автоматизації є забезпечення цілісності та узгодженості даних між усіма модулями вебсистеми. Будь-які зміни, внесені адміністратором, повинні автоматично відображатися в клієнтській частині без необхідності додаткового налаштування або втручання в програмний код. Це стосується актуалізації товарів, характеристик, інформаційного наповнення сайту, а також статусів замовлень і доступності окремих функціональних

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

можливостей. Такий підхід забезпечує коректну взаємодію всіх компонентів системи, зменшує ймовірність виникнення помилок, пов'язаних із ручним оновленням інформації, та спрощує щоденне адміністрування інтернет-магазину.

Таким чином, визначений перелік задач охоплює всі ключові операційні процеси handmade-магазину та формує повне уявлення про функціональний обсяг розроблюваної системи. Автоматизація перелічених процесів дозволить суттєво підвищити ефективність роботи власника магазину, скоротити витрати часу на рутинні операції та покращити якість обслуговування покупців.

1.4. Аналіз існуючих рішень для реалізації вебсистем електронної комерції

Для формування вимог до розроблюваної вебсистеми та обґрунтування доцільності власної розробки було проведено аналіз існуючих рішень, які практично використовуються в українському та міжнародному сегментах ринку для реалізації handmade-продукції. Об'єктами аналізу обрано три конкретні вебсайти: ресурс Lava.aromacandles на платформі Horoshop, ресурс 5candles на платформі OpenCart та ресурс «Бажані» на платформі Shopify. Такий вибір охоплює два основні класи рішень — SaaS-платформи та CMS-системи з відкритим вихідним кодом та дозволяє порівняти їхні можливості стосовно вимог предметної галузі.

Платформа Horoshop є вітчизняним SaaS-рішенням для електронної комерції, що орієнтоване передусім на потреби українського ринку. Сайт Lava.aromacandles, реалізований на цій платформі, демонструє типовий набір можливостей, які надає Horoshop своїм клієнтам. Платформа надає готову адміністративну панель із модулями управління товарами та категоріями, вбудовані інтеграції з популярними українськими платіжними системами, зокрема LiqPay та Приватбанком, та провідними службами доставки, зокрема Новою Поштою та Укрпоштою. Серед переваг платформи є відносна простота запуску: власник може розпочати продажі без глибоких технічних знань, оскільки вся інфраструктура обслуговується безпосередньо сервісом. Вбудовані

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

інструменти пошукової оптимізації, базова аналітика відвідуваності та продажів, а також підтримка адаптивних шаблонів дизайну роблять Horoshop зручним для швидкого старту інтернет-магазину. Водночас аналіз функціоналу платформи в контексті потреб handmade-магазину виявляє суттєві обмеження [27]. Механізм налаштування варіантів товару, реалізований у Horoshop, призначений для стандартних атрибутів на кшталт розміру чи кольору з фіксованим переліком, однак не передбачає можливості динамічного розрахунку ціни залежно від комбінації обраних характеристик кожної позиції. Для реалізації цього функціоналу в рамках платформи потрібні або нестандартні налаштування, або залучення стороннього розробника. Функціонал розділу відгуків, блогу та FAQ присутній, але реалізований у рамках стандартного шаблону і має обмежені можливості кастомізації без доплати за розширені тарифи.

Платформа OpenCart є системою управління вмістом з відкритим вихідним кодом, що надає повний доступ до програмного коду та структури бази даних. Сайт 5candles, реалізований на базі OpenCart, ілюструє можливості цієї системи в контексті handmade-торгівлі [29]. OpenCart підтримує розгалужену систему категорій, можливість призначення кількох зображень для одного товару, систему атрибутів та опцій продукту, налаштовувані статуси замовлень, модулі для роботи з відгуками та базовий функціонал блогу через сторонні розширення. Значною перевагою є відкритість вихідного коду: будь-яку ділянку системи можна змінити або доповнити відповідно до специфічних вимог бізнесу. Офіційний маркетплейс розширень пропонує значну кількість модулів — від платіжних систем до складних інструментів аналітики. Проте саме ця гнучкість є одночасно і суттєвим недоліком для малого handmade-бізнесу. Розгортання OpenCart вимагає наявності власного хостингу, налаштування сервера, встановлення оновлень безпеки та регулярного технічного обслуговування системи. Для власника бізнесу, який не є технічним спеціалістом, це перекладає на нього відповідальність за підтримку серверної інфраструктури або вимагає залучення стороннього адміністратора. Реалізація механізму динамічного

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

ціноутворення на базі OpenCart також потребує або розробки спеціального розширення, або суттєвих модифікацій базового коду, що є нетривіальним завданням.

Платформа Shopify є одним із найпоширеніших міжнародних SaaS-рішень для електронної комерції. Сайт «Бажані», реалізований на цій платформі, демонструє широкі можливості сервісу в контексті handmade-продукції [28]. Shopify забезпечує потужну систему варіантів товару, що підтримує до 100 різних комбінацій характеристик для однієї позиції, детальну аналітику продажів та поведінки покупців, розвинену екосистему застосунків у офіційному маркетплейсі, а також вбудовані інструменти для email-маркетингу та інтеграції з соціальними мережами. Технічна надійність платформи є незаперечною: Shopify гарантує стабільну роботу та самостійно вирішує питання безпеки та масштабування інфраструктури. Водночас використання Shopify пов'язане з рядом практичних обмежень. Платформа потребує регулярних витрат на підписку, що починаються від базового тарифу і можуть суттєво зростати при підключенні необхідних розширень. Реалізація специфічної бізнес-логіки, зокрема, динамічного розрахунку ціни виробу залежно від нестандартних комбінацій матеріалів, потребує або платного спеціалізованого застосунку, або розробки власного. Крім того, адміністративний інтерфейс Shopify значною мірою орієнтований на типові сценарії масової роздрібної торгівлі і може виявитися надлишково складним для власника невеликого handmade-магазину, якому потрібна проста і швидка панель управління [17].

Проведений аналіз показав, що кожне з розглянутих рішень має власні переваги та обмеження. SaaS-платформи забезпечують швидке розгортання інтернет-магазину та мінімальні вимоги до технічного супроводу, проте обмежують можливості зміни бізнес-логіки. OpenCart надає значно ширші можливості для модифікації функціоналу, однак потребує самостійного адміністрування та технічної підтримки. Для узагальнення результатів аналізу

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

та наочного порівняння характеристик розглянутих платформ із власною розробкою сформовано таблицю 1.1.

Таблиця 1.1.

Порівняльна характеристика аналогів

Платформа	Тип рішення	Доступ до програмного коду	Можливість зміни бізнес-логіки	Необхідність технічного супроводу
Horoshop	SaaS	Ні	Обмежена	Низька
OpenCart	CMS/Open Source	Так	Висока	Висока
Shopify	SaaS	Частково	Частково обмежена	Низька
Власна розробка	Custom	Так	Повна	Потребує самостійної підтримки

Як видно з таблиці 1.1, власна розробка є оптимальним варіантом, що поєднує повний доступ до програмного коду, необмежені можливості зміни бізнес-логіки, вбудований механізм динамічного ціноутворення та відсутність регулярних витрат на підписку.

1.5. Обґрунтування доцільності власної розробки

Результати порівняльного аналізу, проведеного в попередньому підрозділі, підтверджують, що жодне з розглянутих рішень не задовольняє в повній мірі специфічні вимоги handmade-магазину без суттєвих додаткових витрат або технічних компромісів. Це є основним аргументом на користь власної розробки, проте не єдиним. Доцільність спеціалізованого програмного комплексу обумовлена сукупністю взаємопов'язаних чинників, кожен із яких окремо є достатнім для обґрунтування такого рішення, а в поєднанні вони формують переконливу аргументацію на його користь.

Центральним аргументом виступає необхідність підтримки динамічного ціноутворення на основі обраних характеристик товару. Як було встановлено при аналізі предметної галузі, кінцева вартість handmade-виробу залежить від комбінації параметрів, які обирає покупець, і кожен із них може мати власну цінову надбавку. Реалізувати цей механізм засобами готових платформ або неможливо без суттєвих модифікацій, або можливо лише через платні сторонні модулі з обмеженими можливостями налаштування. На платформі Horoshop варіанти товару прив'язані до фіксованих артикулів і не підтримують динамічний підрахунок вартості на основі довільних комбінацій характеристик. У Shopify цей функціонал реалізується через окремі платні застосунки, вартість підписки на які нерідко перевищує базовий тариф самої платформи. OpenCart потребує розробки спеціального розширення, що фактично означає замовлення індивідуальної розробки поверх готової системи. Власний програмний комплекс дозволяє реалізувати механізм динамічного ціноутворення точно у тому вигляді, який відповідає бізнес-логіці конкретного магазину, без жодних технічних обхідних шляхів.

Другим важливим чинником є повний контроль над архітектурою системи та можливість її розвитку відповідно до потреб бізнесу, які змінюються з часом. Handmade-бізнес є динамічним: асортимент постійно оновлюється, з'являються нові категорії продукції та тематичні колекції, змінюються умови роботи з покупцями. Сьогодні магазин може потребувати лише базового функціоналу, але вже через рік може виникнути потреба у системі знижок для постійних покупців, у модулі керування оптовими замовленнями або в інтеграції з конкретною службою доставки. Власна кодова база дозволяє впроваджувати будь-які зміни без обмежень, що накладаються закритою архітектурою SaaS-платформ. При використанні готових рішень кожне таке доопрацювання або неможливе в принципі, або потребує очікування, поки постачальник платформи реалізує відповідну функцію у своєму продукті.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

Третім аргументом є економічна доцільність у довгостроковій перспективі. SaaS-платформи вимагають регулярних витрат на підписку, розмір яких зростає зі збільшенням обсягу функціоналу та кількості замовлень. Для малого handmade-бізнесу з обмеженим бюджетом це може становити відчутне фінансове навантаження, особливо на початковому етапі, коли доходи ще нестабільні. Ситуацію додатково ускладнює те, що необхідний функціонал, як правило, не покривається базовим тарифом і потребує підключення платних розширень, кожне з яких також стягує щомісячну плату. Власна розробка потребує одноразових витрат на розробку, після чого система може тривалий час експлуатуватися без жодних регулярних платежів за програмне забезпечення. Витрати на хостинг для Node.js-застосунку та MySQL-бази є суттєво нижчими, ніж сукупна вартість підписки на SaaS-платформу з необхідними розширеннями.

Четвертим чинником є вимога до простоти та відповідності адміністративного інтерфейсу реальним потребам користувача. Власник handmade-магазину не є технічним спеціалістом і потребує інструменту управління, який дозволяє виконувати всі необхідні операції швидко та без тривалого навчання. Адміністративні панелі OpenCart і Shopify є функціонально потужними, але саме тому вони переобтяжені налаштуваннями, більша частина яких ніколи не знадобиться власнику невеликого авторського магазину. Людина, яка щодня виготовляє свічки вручну, не повинна витрачати час на орієнтацію у складній системі з десятками розділів і сотнями опцій лише для того, щоб змінити статус замовлення або завантажити нову фотографію. Власна розробка дозволяє спроектувати адміністративний інтерфейс виключно під реальний набір операцій конкретного магазину, де кожна функція знаходиться рівно там, де її очікує користувач, і де немає нічого зайвого [1].

П'ятим аргументом є незалежність від сторонніх сервісів та пов'язаних із ними ризиків. При використанні SaaS-платформ власник магазину цілком залежить від стабільності роботи, цінової політики та стратегічних рішень постачальника. Якщо платформа змінить умови обслуговування, суттєво

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

підвищить тарифи або припинить роботу, перенесення магазину на інше рішення може виявитися тривалим і витратним процесом. Особливо гостро ця проблема стоїть при використанні закритих SaaS-платформ, де дані зберігаються у форматі, прив'язаному до конкретного сервісу, і їхній експорт може бути ускладнений або обмежений умовами договору. Власна розробка з відкритим доступом до бази даних і вихідного коду усуває цю залежність повністю: система може бути перенесена на будь-який хостинг, що підтримує Node.js та MySQL, без будь-яких обмежень з боку третіх сторін.

Нарешті, суттєвим чинником є можливість точного контролю над безпекою системи та персональними даними покупців. При використанні SaaS-платформ власник магазину фактично передає відповідальність за зберігання даних клієнтів сторонньому постачальнику, покладаючись на його політику безпеки. Власна розробка дає змогу самостійно визначати, де і як зберігаються дані, реалізовувати необхідні механізми захисту та контролювати доступ до інформаційної системи. У розробленому програмному комплексі захист реалізовано через хешування паролів засобами bcrypt, JWT-автентифікацію та серверну перевірку прав доступу, що відповідає сучасним стандартам безпеки вебдодатків [21, 22, 23].

З урахуванням усіх наведених чинників власна розробка вебсистеми електронної комерції для handmade-магазину є обґрунтованим рішенням, що забезпечує необхідну функціональну відповідність, економічну доцільність і технічну незалежність. Для реалізації програмного комплексу обрано сучасний стек технологій: React для клієнтської частини, Node.js з Express для серверної частини, MySQL як систему керування базами даних та ORM Sequelize для взаємодії з базою даних. Цей стек є добре документованим, широко підтримуваним спільнотою та забезпечує необхідну гнучкість для реалізації всього запланованого функціоналу [3, 4, 8].

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 1

У першому розділі проведено аналіз предметної області та визначено особливості функціонування інтернет-магазину handmade-продукції. Досліджено основні бізнес-процеси, що потребують автоматизації, зокрема управління каталогом товарів, характеристиками продукції, замовленнями, користувачами та інформаційним наповненням сайту. Також виконано аналіз існуючих рішень для реалізації вебсистем електронної комерції та визначено їх переваги й обмеження з точки зору потреб малого handmade-бізнесу.

За результатами проведеного аналізу обґрунтовано доцільність розробки власної вебсистеми, орієнтованої на підтримку індивідуальних характеристик товарів, автоматизацію процесу оформлення замовлень та спрощення управління діяльністю handmade-магазину. Отримані результати стали основою для вибору архітектури програмного комплексу, визначення вимог до системи та подальшого проєктування структури бази даних, алгоритмів роботи та програмних модулів вебсистеми.

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

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБСИСТЕМИ ІНТЕРНЕТ-МАГАЗИНУ HANDMADE-СВІЧОК

2.1. Визначення вимог та варіантів використання системи

Формування вимог до вебсистеми є ключовим етапом проєктування, що визначає функціональні межі програмного комплексу та є основою для всіх подальших проєктних рішень. Вимоги поділяються на функціональні, що визначають конкретні можливості системи, та нефункціональні, що описують якісні характеристики програмного продукту [2].

Під час формування вимог було враховано специфіку предметної області, пов'язану з реалізацією виробів ручної роботи. На відміну від типових інтернет-магазинів, система повинна підтримувати індивідуальне налаштування окремих товарів, автоматичний перерахунок їхньої вартості залежно від обраних характеристик, а також забезпечувати зручне адміністрування асортименту, інформаційних сторінок і замовлень.

Функціональні вимоги до клієнтської частини системи охоплюють кілька ключових напрямків. Перший із них — це відображення каталогу продукції. Система повинна надавати відвідувачам можливість переглядати повний каталог товарів із розподілом за категоріями та підкатегоріями, здійснювати перехід між розділами без перезавантаження сторінки та отримувати детальну інформацію про кожен виріб, зокрема повний опис, склад, строки виготовлення та кілька фотографій. Відображення товарів повинно враховувати їхній статус активності, деактивовані позиції не повинні з'являтися в каталозі незалежно від способу формування запиту.

Другим напрямком функціональних вимог є кастомізація товарів. Для товарів, що підтримують індивідуальне налаштування, система повинна надавати покупцеві можливість обирати доступні кольори, аромати та декоративні елементи. При кожній зміні обраного параметра вартість виробу повинна автоматично перераховуватися без перезавантаження сторінки. У

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

переліку доступних опцій повинні відображатися лише матеріали зі статусом «активний», недоступні позиції не повинні бути видимі для покупця.

Третій напрямок функціональних вимог стосується роботи з кошиком. Покупець повинен мати можливість додавати товари до кошика із збереженням обраних характеристик та кількості, змінювати кількість доданих позицій, видаляти окремі товари з кошика, а також бачити актуальну загальну вартість замовлення після кожної зміни. Вміст кошика повинен зберігатися між сесіями браузера.

Четвертий напрямок — це оформлення замовлення. При переході до оформлення система відображає форму для введення контактних даних покупця: імені, номера телефону, адреси електронної пошти, адреси доставки. Підтримується додавання коментаря до замовлення. Перед збереженням система виконує валідацію введених даних. Після успішного оформлення вміст кошика автоматично очищується, а замовлення фіксується в базі даних із початковим статусом.

П'ятий напрямок функціональних вимог охоплює систему авторизації та особистий кабінет. Покупець може зареєструватися, вказавши ім'я, електронну пошту та пароль. Авторизація виконується за адресою електронної пошти та паролем. Зареєстрований покупець отримує доступ до особистого кабінету, де відображається його персональна інформація та повна історія замовлень із поточними статусами. Персональні дані доступні для редагування.

Шостий напрямок — інформаційні сторінки. Система відображає статті блогу, відповіді на поширені запитання та контактну інформацію. Ці сторінки формуються на основі даних, що зберігаються у базі та оновлюються адміністратором.

Функціональні вимоги до адміністративної частини охоплюють повний спектр операцій управління системою. Адміністратор повинен мати можливість створювати, редагувати та деактивувати товари; завантажувати кілька зображень для кожного товару та призначати головне; управляти категоріями та

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

підкатегоріями каталогу; керувати переліком матеріалів із зазначенням їхнього типу та цінової надбавки. Окрім цього, адміністратор переглядає список усіх замовлень, отримує деталі кожного замовлення та змінює його статус, управляє обліковими записами покупців, переглядаючи їхні дані та змінюючи статус активності, модерує відгуки покупців, оновлює банери головної сторінки, статті блогу та записи FAQ.

Нефункціональні вимоги охоплюють якісні характеристики системи. Інтерфейс як клієнтської частини, так і адміністративної панелі повинен коректно відображатися та повноцінно функціонувати на пристроях із різними розмірами екрана. Система повинна забезпечувати захист персональних даних користувачів через хешування паролів і JWT-автентифікацію. Взаємодія між клієнтською та серверною частинами реалізується виключно через REST API, що забезпечує незалежність компонентів [24, 25]. Архітектура системи повинна передбачати можливість подальшого розширення функціоналу без зміни базової структури.

На підставі визначених вимог сформовано варіанти використання системи для трьох категорій акторів: незареєстрованого відвідувача, зареєстрованого покупця та адміністратора. Незареєстрований відвідувач має доступ до перегляду каталогу, карток товарів, блогу, сторінки FAQ та контактів, а також може формувати кошик і оформлювати замовлення. Зареєстрований покупець отримує всі права незареєстрованого відвідувача та додатково — доступ до особистого кабінету з переглядом і редагуванням персональних даних і переглядом історії замовлень. Адміністратор, авторизований через захищений маршрут, має доступ до повного функціоналу адміністративної панелі. Діаграму варіантів використання системи наведено на рис. 2.1.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

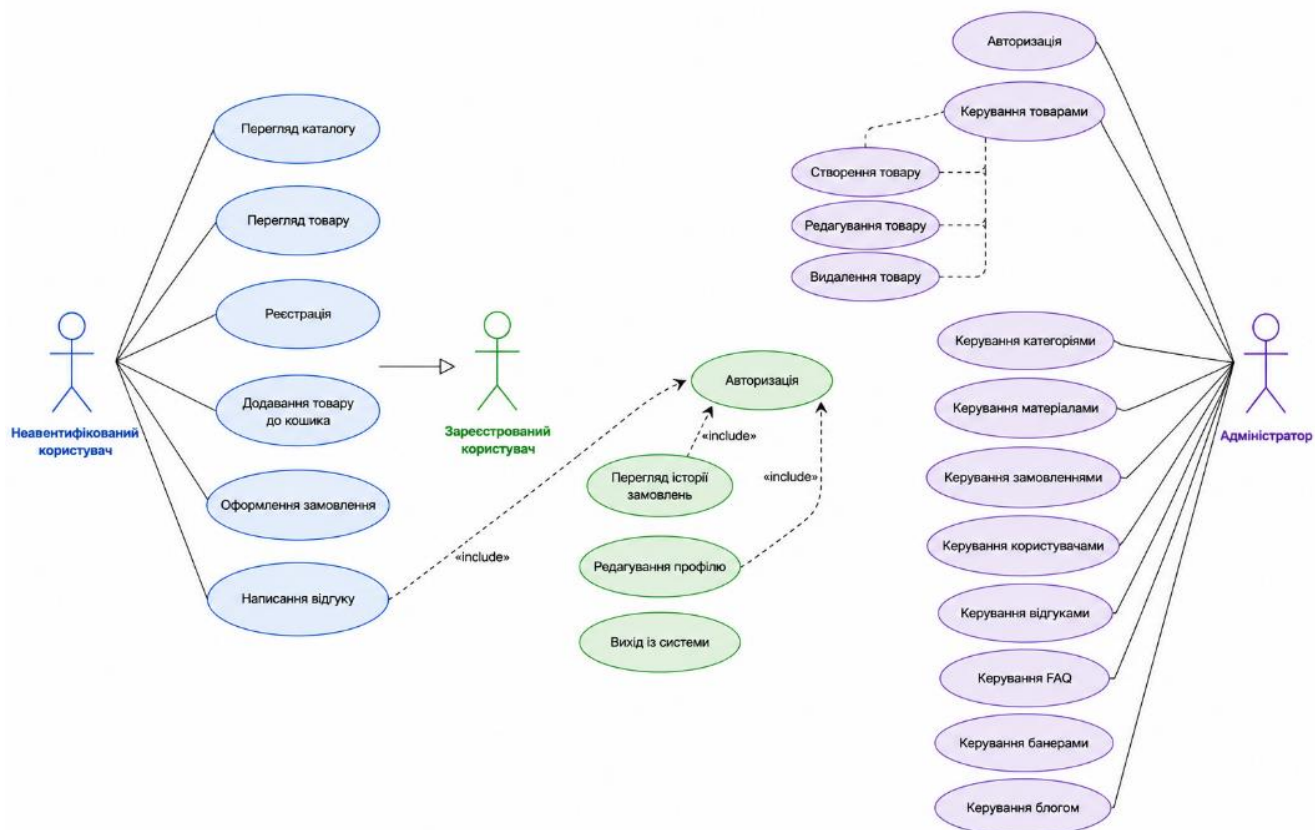


Рис. 2.1. Діаграма варіантів використання системи

2.2. Проектування бази даних системи

База даних є центральним компонентом будь-якої системи електронної комерції, від якості її проектування безпосередньо залежать цілісність даних, ефективність запитів та простота подальшого супроводу системи. При проектуванні структури бази даних для розроблюваної вебсистеми прийнято рішення на користь реляційної моделі даних, яка забезпечує жорстко визначені зв'язки між сутностями, підтримку транзакцій та референційну цілісність. Як СУБД обрано MySQL — зрілу, добре документовану систему з широкими можливостями оптимізації запитів [5, 6].

Проектування структури бази даних розпочалося з виявлення основних сутностей системи та зв'язків між ними [7]. Кожна сутність відповідає одній або кільком таблицям бази даних; зв'язки між таблицями реалізуються через зовнішні ключі з відповідними обмеженнями цілісності. ER-діаграму бази даних наведено на рис. 2.2.

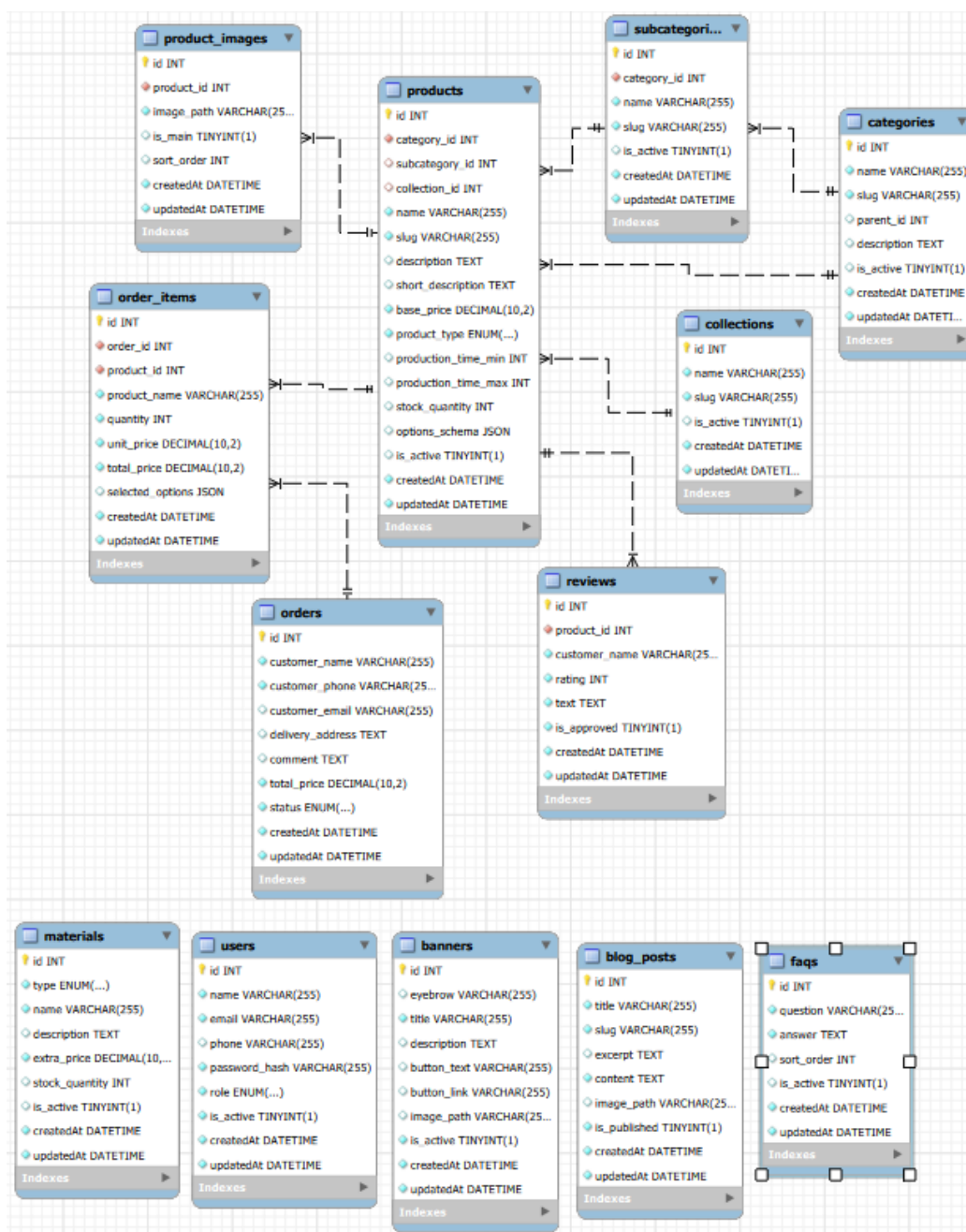


Рис. 2.2. ER-діаграма бази даних системи.

Центральне місце в структурі бази даних займає таблиця `products`, що зберігає інформацію про товари каталогу. Кожен запис описує один виріб і містить такі основні поля: назву (`name`), унікальний URL-ідентифікатор (`slug`), повний та скорочений описи, базову ціну (`base_price`), тип виробу — готовий або під замовлення (`product_type`), мінімальний і максимальний строки виготовлення у днях, кількість готового товару (`stock_quantity`), JSON-поле зі схемою доступних опцій (`options_schema`) та прапорець активності (`is_active`). Кожен

товар прив'язаний до категорії через зовнішній ключ `category_id`, а також може бути пов'язаний із конкретною підкатегорією (`subcategory_id`) та тематичною колекцією (`collection_id`) — обидва поля є необов'язковими та допускають значення NULL. Поле `options_schema` зберігає структуру доступних опцій у форматі JSON, масиви ідентифікаторів матеріалів за типами, що дозволяє гнучко налаштовувати перелік характеристик для кожного товару окремо без зміни схеми таблиці.

Ієрархія каталогу реалізована через дві окремі таблиці. Таблиця `categories` зберігає основні розділи каталогу: формові свічки, контейнерні свічки, свічки з вощини, аромасаше, підставки, магічну колекцію та категорію наявних товарів. Кожен запис містить поля назви, `slug`, опис та прапорець активності `is_active`. Таблиця `subcategories` деталізує кожну категорію і пов'язана з нею зовнішнім ключем `category_id` з каскадним видаленням. Наприклад, категорія формових свічок має підкатегорії «Торси», «Колони та куби», «Домашній затишок» та «Зимова казка». Такий підхід розділення категорій і підкатегорій на дві таблиці забезпечує чітку структуру даних і спрощує побудову навігаційного дерева на клієнтській стороні.

Таблиця `collections` зберігає тематичні колекції, до яких можуть належати товари: «Різдво», «Рош га-Шана», «Романтичні» тощо. Зв'язок між товаром та колекцією є необов'язковим, оскільки товар може не належати жодній колекції. Це дозволяє формувати тематичні добірки виробів без прив'язки до ієрархічної структури категорій.

Для підтримки кількох зображень для одного товару використовується окрема таблиця `product_images`. Це рішення є більш гнучким порівняно зі зберіганням шляхів у самій таблиці `products`, оскільки дозволяє додавати необмежену кількість фотографій та управляти їхнім порядком відображення через поле `sort_order`. Поле `is_main` ідентифікує головне зображення, що відображається як обкладинка в каталозі. Шлях до файлу зберігається у полі `image_path`.

Таблиця materials реалізує збереження всіх характеристик, доступних для кастомізації виробів: кольорів, ароматів і декоративних елементів. Тип кожного матеріалу (color, fragrance, decor) визначається ENUM-полем type. Поле extra_price зберігає цінову надбавку при виборі даного матеріалу. Поля description і stock_quantity дозволяють зберігати короткий опис матеріалу та відстежувати кількість у наявності. Поле is_active дозволяє тимчасово приховувати недоступні матеріали без їхнього видалення з бази.

Для реалізації процесу замовлень використовуються дві взаємопов'язані таблиці. Таблиця orders зберігає загальну інформацію про кожне замовлення: ім'я покупця, номер телефону, електронну пошту, адресу доставки, коментар, загальну вартість та поточний статус. Статус реалізований як ENUM-поле зі значеннями new, processing, ready, shipped, completed та cancelled, що відображають повний життєвий цикл замовлення від моменту оформлення до завершення або скасування. Таблиця order_items зберігає окремі позиції кожного замовлення з посиланнями на замовлення (order_id) і товар (product_id), зафіксованою назвою товару (product_name), кількістю, ціною за одиницю, загальною вартістю позиції та обраними характеристиками у полі selected_options формату JSON.

Таблиця users об'єднує облікові записи покупців та адміністраторів. Розмежування ролей здійснюється через ENUM-поле role зі значеннями customer та admin. Пароль зберігається виключно у вигляді bcrypt-хешу у полі password_hash. Поле is_active дозволяє заблокувати обліковий запис без його видалення, зберігаючи всю пов'язану з ним інформацію.

Для підтримки функціоналу відгуків використовується таблиця reviews, що зберігає ім'я покупця (customer_name), числову оцінку від 1 до 5, текст відгуку (text), посилання на товар (product_id) та прапорець модерації is_approved. Відображення відгуку в клієнтській частині стає можливим лише після підтвердження адміністратором. Інформаційне наповнення сайту зберігається у трьох таблицях: blog_posts — це статті блогу зі slug, анонсом

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

(excerpt), повним текстом, шляхом до зображення та прапорцем публікації, таблиця faqs містить питання й відповіді із полем сортування sort_order та прапорцем активності та banners, таблиця, що містить рекламні банери з надписом над заголовком (eyebrow), заголовком, описом, текстом кнопки, посиланням, шляхом до зображення та прапорцем активності. Перелік усіх сутностей бази даних наведено у таблиці 2.1.

Таблиця 2.1

Основні сутності бази даних.

Таблиця	Призначення
users	Облікові записи покупців та адміністраторів
products	Товари каталогу з базовими характеристиками та схемою опцій
categories	Основні розділи каталогу
subcategories	Підкатегорії, прив'язані до категорій
collections	Тематичні колекції товарів
product_images	Зображення товарів із визначенням головного фото
materials	Характеристики для кастомізації (колір, аромат, декор)
orders	Замовлення покупців із контактними даними та статусом
order_items	Позиції замовлення з обраними характеристиками
reviews	Відгуки покупців із системою модерації
blog_posts	Статті блогу
Faqs	Поширені запитання та відповіді
banners	Банери головної сторінки

Детальну структуру кожної таблиці з переліком полів, їхніх типів та описів наведено у додатках А, Б, В, Г, Ґ, Д, Е. Спроектована структура бази даних із 13 таблицями охоплює всі сутності, необхідні для підтримки бізнес-процесів магазину, та є достатньо гнучкою для подальшого розширення без суттєвих змін існуючої схеми.

2.3. Проєктування алгоритмів роботи системи

Коректна реалізація бізнес-логіки вебсистеми забезпечується алгоритмами, що визначають послідовність дій при виконанні ключових операцій. До основних алгоритмів, що підлягають детальному проєктуванню, належать: механізм реєстрації та авторизації користувачів, формування та оновлення кошика, динамічний розрахунок вартості товару та процес оформлення замовлення. Кожен із них охоплює взаємодію клієнтської частини, серверної частини та бази даних і вимагає чіткого визначення послідовності операцій та обробки граничних випадків [7].

Процес реєстрації нового користувача розпочинається із заповнення форми, у якій необхідно вказати ім'я, адресу електронної пошти та пароль. Після надсилання форми клієнтська частина передає введені дані до серверної частини для їхнього опрацювання. Сервер перевіряє коректність заповнення обов'язкових полів, правильність формату електронної пошти та відсутність користувача з такою адресою в базі даних. Якщо перевірка завершується успішно, пароль хешується за допомогою бібліотеки `bcrypt`, після чого новий обліковий запис зберігається в таблиці `users`. Після успішної реєстрації сервер формує JWT-токен, що містить ідентифікатор користувача та його роль, і повертає його клієнтській частині разом із основними даними облікового запису.

Процес авторизації розпочинається з введення користувачем адреси електронної пошти та пароля. Після надсилання форми клієнтська частина передає введені дані до серверної частини. Сервер виконує пошук користувача за вказаною адресою електронної пошти та перевіряє відповідність введеного пароля хешованому значенню за допомогою бібліотеки `bcrypt`. Якщо обліковий

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

запис не знайдено, пароль введено неправильно або користувача заблоковано, сервер повертає повідомлення про помилку. У разі успішної перевірки формується JWT-токен, який передається клієнтській частині. Отриманий токен зберігається в localStorage браузера та використовується для авторизації під час звернення до захищених ресурсів системи.

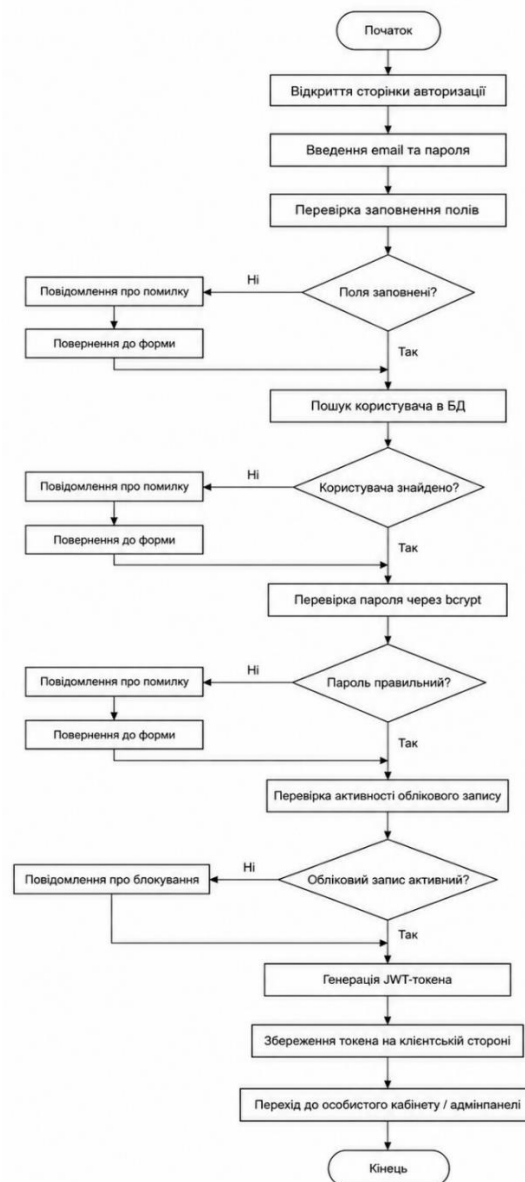


Рис. 2.3. Блок-схема алгоритму авторизації користувача

Перевірка автентичності на серверній стороні здійснюється за допомогою middleware-функції checkAuth, яка використовується для всіх маршрутів, що потребують авторизації. Функція отримує JWT-токен із заголовка Authorization,

перевіряє його коректність, термін дії та декодує дані користувача за допомогою бібліотеки `jsonwebtoken`. Якщо токен відсутній, пошкоджений або його термін дії минув, сервер повертає повідомлення про помилку, а подальше виконання запиту припиняється. Для маршрутів адміністративної панелі додатково використовується функція `checkAdmin`, яка перевіряє роль користувача та надає доступ лише адміністраторам [17].

Алгоритм формування та управління кошиком покупця цілком реалізований на клієнтській стороні з використанням `localStorage` браузера як сховища. Кошик зберігається у вигляді JSON-масиву об'єктів, кожен із яких містить ідентифікатор товару, його назву, ціну на момент додавання, кількість та обрані характеристики. При додаванні товару до кошика клієнтська частина перевіряє, чи немає вже в кошику ідентичного товару з тими самими характеристиками. Якщо такий запис знайдено, збільшується лише кількість, в іншому випадку до масиву додається новий запис. Після кожної операції оновлений масив серіалізується та зберігається у `localStorage`. Загальна вартість кошика обчислюється динамічно як сума добутків ціни та кількості для кожної позиції [19]. При успішному оформленні замовлення масив очищується.

Алгоритм динамічного розрахунку вартості товару є однією з ключових особливостей системи. При завантаженні сторінки товару клієнтська частина виконує запит до API для отримання переліку активних матеріалів, згрупованих за типами. Компонент зберігає поточний вибір матеріалів у стані React (`useState`). Початково для кожного типу матеріалу обирається перший доступний варіант, і кінцева ціна розраховується як сума базової ціни товару та `extra_price` усіх обраних матеріалів. При зміні будь-якого матеріалу стан оновлюється, і функція розрахунку ціни викликається заново. Оскільки ця функція є суто клієнтською і оперує вже завантаженими даними, перерахунок відбувається миттєво без звернення до сервера.

Алгоритм оформлення замовлення охоплює взаємодію клієнтської та серверної частин системи. Після заповнення форми клієнтська частина формує

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

дані замовлення, які містять перелік товарів із кошика, їх кількість, ціну, обрані характеристики та контактні дані покупця. Перед надсиленням виконується перевірка заповнення обов'язкових полів і коректності номера телефону. Після успішної перевірки дані передаються до серверної частини, де повторно проходять валідацію. Блок-схему алгоритму оформлення замовлення наведено на рис. 2.4.

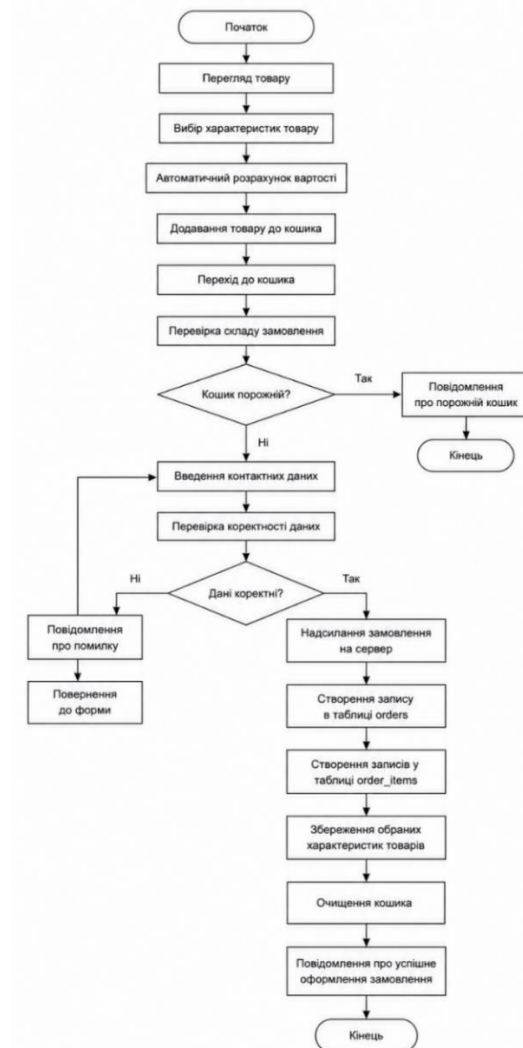


Рис. 2.4. Блок-схема алгоритму оформлення замовлення.

Після перевірки вхідних даних сервер створює запис у таблиці orders і відповідні записи в таблиці order_items у межах однієї транзакції. Використання транзакції забезпечує цілісність даних: якщо під час збереження виникає помилка, усі внесені зміни скасовуються, а замовлення не зберігається частково [6]. Фрагмент серверної реалізації алгоритму наведено на рис. 2.5.

```

let totalPrice = 0;

const order = await Order.create({
  customer_name,
  customer_phone,
  customer_email: customer_email || null,
  delivery_address: delivery_address || null,
  comment: comment || null,
  total_price: 0,
  status: 'new',
});

for (const item of items) {
  const product = await Product.findByPk(item.product_id);

  const quantity = item.quantity || 1;
  const unitPrice = Number(product.base_price);
  const itemTotal = unitPrice * quantity;

  totalPrice += itemTotal;

  await OrderItem.create({
    order_id: order.id,
    product_id: product.id,
    product_name: product.name,
    quantity,
    unit_price: unitPrice,
    total_price: itemTotal,
    selected_options: item.selected_options || null,
  });
}

await order.update({
  total_price: totalPrice,
});

```

Рис. 2.5. Фрагмент створення замовлення.

Алгоритм обробки замовлень на стороні адміністратора забезпечує керування їх виконанням. Після отримання нового замовлення адміністратор переглядає його деталі в адміністративній панелі та змінює статус відповідно до поточного етапу обробки. Після підтвердження зміни статус оновлюється в базі даних. Під час видалення замовлення система відображає модальне вікно підтвердження, що запобігає випадковому видаленню замовлення.

2.4. Реалізація програмного комплексу

Реалізацію програмного комплексу виконано із дотриманням принципів клієнт-серверної архітектури, при якій клієнтська та серверна частини є незалежними один від одного модулями, що взаємодіють виключно через REST API. Це рішення забезпечує чітке розмежування відповідальності між компонентами та спрощує як поточну розробку, так і майбутній розвиток системи.

Серверна частина реалізована на платформі Node.js із використанням фреймворку Express. Express є мінімалістичним і гнучким фреймворком, що дозволяє організовувати маршрутизацію HTTP-запитів та middleware-ланцюжки без надлишкових абстракцій [4, 11, 12]. Для взаємодії з базою даних MySQL використовується ORM Sequelize, що дозволяє описувати структуру таблиць та зв'язки між ними у вигляді JavaScript-класів моделей і виконувати запити через об'єктно-орієнтований API без написання SQL вручну.

Структура серверної частини побудована за принципом розподілу відповідальності між шарами. Файли у каталозі config/ містять налаштування підключення до бази даних та параметри, що зчитуються зі змінних середовища (файл .env). Файли у каталозі models/ описують ORM-моделі Sequelize для кожної таблиці бази даних, визначають типи полів, обмеження та асоціації між моделями (hasMany, belongsTo,hasOne). Файли у каталозі controllers/ реалізують бізнес-логіку для кожної групи ресурсів. Файли у каталозі routes/ описують маршрути REST API та прив'язують HTTP-методи і шляхи до відповідних функцій контролерів. Директорія middleware/ містить функції checkAuth та checkAdmin, а також конфігурацію Multer для обробки завантажень зображень.

Для захисту даних реалізовано механізм хешування паролів із використанням бібліотеки bcrypt та JWT-автентифікацію. При реєстрації пароль хешується функцією bcrypt.hash() перед записом до бази даних. Під час авторизації порівняння введеного пароля з хешованим значенням виконується функцією bcrypt.compare(). У разі успішної автентифікації сервер формує JWT-токен за допомогою бібліотеки jsonwebtoken, підписуючи корисне навантаження секретним ключем із файлу .env. Сформований токен передається клієнтській частині та використовується в заголовку Authorization під час виконання наступних запитів до захищених ресурсів [16, 21].

На рис. 2.6 наведено фрагмент програмного коду, що реалізує перевірку облікових даних користувача під час авторизації.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

У наведеному фрагменті програмного коду здійснюється пошук користувача за адресою електронної пошти, перевірка активності облікового запису, порівняння введеного пароля з його хешованим значенням та виклик функції генерації JWT-токена у разі успішної автентифікації.

```
const user = await User.findOne({
  where: {
    email,
    role: "customer",
    is_active: true,
  },
});

if (!user) {
  return res.status(401).json({
    message: "Невірний email або пароль",
  });
}

const isValidPassword = await bcrypt.compare(
  password,
  user.password_hash
);

if (!isValidPassword) {
  return res.status(401).json({
    message: "Невірний email або пароль",
  });
}

const token = generateToken(user);
```

Рис. 2.6. Фрагмент реалізації авторизації користувача.

У системі реалізовано розмежування прав доступу між покупцями та адміністраторами. Доступ до адміністративної частини системи надається виключно користувачам із роллю адміністратора, що перевіряється під час обробки кожного захищеного запиту. Такий підхід забезпечує захист службового функціоналу від несанкціонованого доступу та гарантує виконання адміністративних операцій лише уповноваженими користувачами.

Адміністративна панель реалізована як окрема частина вебдодатка та забезпечує централізоване керування всіма основними сутностями системи.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Через адміністративний інтерфейс реалізовано можливість створення, редагування та приховування товарів, керування категоріями, матеріалами, замовленнями, користувачами, відгуками, банерами, записами блогу та розділом поширених запитань. Для зручності роботи адміністратора реалізовано адаптивний інтерфейс, що дозволяє виконувати основні операції як з персонального комп'ютера, так і з мобільних пристроїв.

Під час реалізації програмного комплексу особливу увагу приділено забезпеченню узгодженої взаємодії між модулями вебсистеми. Кожен компонент системи виконує визначене коло функцій і взаємодіє з іншими компонентами через чітко визначені інтерфейси, що спрощує супровід і подальший розвиток програмного продукту. Такий підхід забезпечує можливість незалежної модифікації окремих модулів без впливу на роботу інших компонентів системи, а також полегшує тестування, пошук і усунення можливих помилок під час її експлуатації.

У результаті проведеної розробки створено повнофункціональну вебсистему інтернет-магазину handmade-свічок, яка забезпечує автоматизацію процесів продажу продукції, обробки замовлень, управління асортиментом та інформаційним наповненням сайту. Обрана архітектура забезпечує можливість подальшого розвитку вебсистеми та інтеграції нових функціональних модулів, зокрема підтримки онлайн-оплати, без необхідності суттєвої зміни базової структури програмного продукту.

2.5. Структура програмного комплексу

Програмний комплекс реалізовано за клієнт-серверною архітектурою та побудовано за модульним принципом, що забезпечує розділення відповідальності між окремими компонентами системи, спрощує супровід програмного коду та створює можливість подальшого розширення функціональних можливостей вебсистеми. Структура проєкту охоплює клієнтську частину інтернет-магазину, адміністративний модуль, серверну

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

частину та базу даних. Узагальнену структуру програмного комплексу наведено на рис. 2.7.

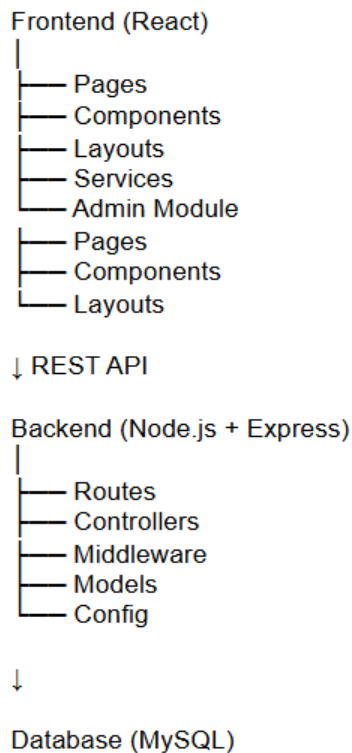


Рис. 2.7. Узагальнена структура програмного комплексу

Клієнтська частина реалізована із використанням бібліотеки React та містить набір сторінок, компонентів, шаблонів відображення й сервісних модулів. Основні сторінки системи розміщені у каталозі pages/ та забезпечують взаємодію користувача з функціоналом інтернет-магазину. Головна сторінка відображає банер, блок категорій та секцію останніх товарів. Сторінка каталогу дозволяє перегляд товарів із фільтрацією за категоріями та підкатегоріями. Сторінка окремого товару містить галерею зображень, опис виробу та блок вибору характеристик із динамічним перерахунком вартості. Сторінка кошика поєднує перелік доданих позицій із формою оформлення замовлення. Для роботи з обліковим записом реалізовано сторінки авторизації, реєстрації та особистого кабінету з можливістю перегляду історії замовлень. Інформаційна частина сайту представлена сторінками «Про нас», блогу, поширених запитань і контактів.

Повторно використовувані елементи інтерфейсу винесені до каталогу components/. До них належать шапка сайту, підвал, картка товару, слайдер категорій, список поширених запитань та інші компоненти, що використовуються на кількох сторінках одночасно. Винесення спільних компонентів до окремого каталогу усуває дублювання програмного коду та забезпечує єдиний стиль оформлення інтерфейсу. Основою структури сторінок є шаблон MainLayout, який містить спільні елементи інтерфейсу, навігаційне меню та підвал і використовується всіма сторінками клієнтської частини.

Особливістю клієнтської частини є реалізація механізму кастомізації товарів безпосередньо на сторінці картки виробу. Під час перегляду покупець обирає доступні кольори, аромати та декоративні елементи зі списків, що формуються на основі активних матеріалів із бази даних. Після кожної зміни характеристики система автоматично перераховує вартість виробу на стороні клієнта без звернення до сервера та без перезавантаження сторінки. Це дозволяє швидко реакцію інтерфейсу та зручну взаємодію з каталогом продукції.

Для організації навігації між сторінками без повного перезавантаження браузера використовується бібліотека React Router [9, 30]. Завдяки цьому переходи між розділами сайту відбуваються без повторного завантаження сторінок, що підвищує швидкодію застосунку та покращує взаємодію користувача із системою. Вміст кошика зберігається як глобальний стан за допомогою React Context API та синхронізується з localStorage браузера при кожній зміні, що забезпечує збереження доданих товарів між сеансами [19].

Усі сторінки та компоненти клієнтської частини реалізовано з урахуванням вимоги адаптивності. Інтерфейс коректно відображається на пристроях із різною роздільною здатністю екрана — від широкоформатних моніторів до смартфонів. Це забезпечує зручну роботу як покупців, які переглядають каталог із мобільних пристроїв, так і власника магазину, який може керувати роботою системи незалежно від типу пристрою.

Адміністративна частина системи реалізована як окремий модуль, розміщений у каталозі admin/. Вона має власний набір сторінок, компонентів і шаблон AdminLayout із боковим меню навігації. Доступ до адміністративного модуля обмежується компонентом ProtectedAdminRoute, який перевіряє наявність дійсного JWT-токена та роль користувача перед відображенням захищених сторінок. За відсутності токена або недостатнього рівня доступу виконується автоматичне перенаправлення на сторінку входу.

Адміністративна панель забезпечує керування всіма основними сутностями системи. Через її інтерфейс адміністратор може створювати, редагувати та приховувати товари, керувати категоріями, підкатегоріями, тематичними колекціями й матеріалами, переглядати замовлення та змінювати їхній статус. Під час видалення замовлення система відображає вікно підтвердження дії, що запобігає випадковому видаленню даних. Також реалізовано керування обліковими записами користувачів, модерацію відгуків і редагування інформаційного наповнення сайту, зокрема банерів головної сторінки, статей блогу та розділу поширених запитань. Усі ці операції виконуються через адміністративний інтерфейс без необхідності внесення змін до програмного коду.

Серверна частина програмного комплексу реалізована на платформі Node.js із використанням фреймворку Express та побудована за багаторівневим підходом. Конфігураційні файли у каталозі config/ відповідають за налаштування підключення до бази даних і зчитують параметри зі змінних середовища. Моделі Sequelize у каталозі models/ описують структуру таблиць та зв'язки між ними. Контролери у каталозі controllers/ реалізують бізнес-логіку, виконують валідацію вхідних даних, взаємодіють із моделями та формують відповіді у форматі JSON. Маршрути у каталозі routes/ організовують REST API, пов'язують HTTP-методи з відповідними функціями контролерів і застосовують необхідні middleware [18]. Каталог middleware/ містить модулі перевірки автентифікації та прав

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

адміністратора, а також конфігурацію Multer для обробки завантаження зображень.

Архітектура серверної частини за схемою routes → controllers → models забезпечує чітке розділення відповідальності між окремими шарами застосунку. Завдяки такому підходу структура серверного проєкту залишається зрозумілою незалежно від кількості реалізованих сутностей. Для додавання нового ресурсу достатньо створити відповідну модель, контролер і файл маршрутів та підключити їх до застосунку. Це спрощує супровід програмного коду, полегшує розширення функціональності системи та дозволяє розвивати проєкт без внесення суттєвих змін до вже реалізованих модулів.

Висновки до розділу 2

У другому розділі виконано проєктування вебсистеми інтернет-магазину handmade-свічок та визначено основні принципи її реалізації. На основі сформованих вимог до програмного продукту розроблено варіанти використання системи для користувачів і адміністраторів, що дозволило визначити функціональні межі програмного комплексу та основні бізнес-процеси.

Спроектовано структуру бази даних, яка забезпечує зберігання інформації про товари, користувачів, замовлення й інформаційне наповнення сайту, підтримує роботу з індивідуальними характеристиками виробів, історією замовлень і розмежуванням прав доступу між різними категоріями користувачів.

Описано алгоритми роботи основних підсистем та реалізовано програмний комплекс із використанням технологій React, Node.js, Express, MySQL і Sequelize. Сформовано структуру клієнтської та серверної частин системи, що забезпечує стабільну взаємодію між її компонентами.

Отримані результати створюють необхідну основу для функціонування вебсистеми, забезпечують автоматизацію основних процесів handmade-магазину та дозволяють здійснювати подальший розвиток програмного продукту без суттєвих змін його архітектури.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБСИСТЕМОЮ ІНТЕРНЕТ-МАГАЗИНУ HANDMADE-СВІЧОК

3.1 Встановлення та налаштування системи

Розгортання програмного комплексу передбачає послідовне виконання ряду підготовчих кроків, після яких система стає доступною для роботи через веббраузер. Загальну схему розгортання наведено на рис. 3.1.

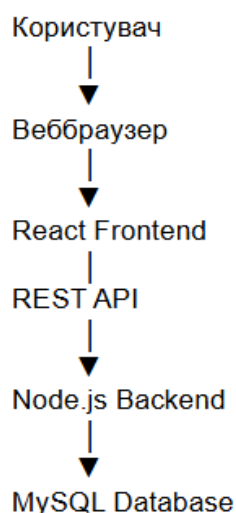


Рис. 3.1. Схема розгортання програмного комплексу

Першим кроком є підготовка програмного середовища. На сервері або робочій станції повинно бути встановлено Node.js версії 18 LTS або новішої разом із менеджером пакетів `npm`, а також MySQL 8.x. Node.js забезпечує середовище виконання серверної частини, а `npm` використовується для встановлення та керування залежностями обох частин проєкту. Перевірити наявність необхідних компонентів можна командами `node --version`, `npm --version` та `mysql --version` у терміналі — кожна з них повинна повернути відповідний номер версії без повідомлень про помилку. Для локальної розробки та адміністрування бази даних зручно додатково встановити MySQL Workbench або аналогічний графічний інструмент, що дозволяє переглядати структуру таблиць, виконувати довільні SQL-запити та імпортувати дампи без використання командного рядка.

Другим кроком є налаштування бази даних. У MySQL створюється порожня база із заданою назвою та кодуванням `utf8mb4`, що забезпечує коректну підтримку Unicode, зокрема кириличних символів і спеціальних знаків, які використовуються в описах товарів та текстах контенту. Якщо на цьому етапі використати кодування, що не підтримує повний набір символів Unicode, частина даних може зберігатися некоректно, що призведе до проблем із відображенням інформації в інтерфейсі. Після створення бази необхідно визначити користувача MySQL із відповідними привілеями — щонайменше правами на виконання операцій `SELECT`, `INSERT`, `UPDATE`, `DELETE` та `CREATE` для відповідної бази даних. Використання окремого користувача замість облікового запису `root` є рекомендованою практикою з міркувань безпеки. Якщо розгортання виконується на основі наявного дампу з тестовими або початковими даними, він імпортується на цьому ж етапі через команду `mysql` або засобами графічного інтерфейсу.

Третім кроком є налаштування файлу змінних середовища. У кореневому каталозі серверної частини створюється файл `.env`, який містить параметри, що залежать від конкретного середовища розгортання і не повинні бути зафіксовані у програмному коді. Насамперед вказуються дані для підключення до бази даних: адреса хоста, порт, назва бази, ім'я користувача та пароль. При локальному розгортанні адресою хоста є `localhost`, а стандартний порт MySQL дорівнює `3306`. Окремим важливим параметром є секретний ключ для підписання JWT-токенів. Цей рядок повинен бути достатньо довгим і непередбачуваним, оскільки його компрометація дозволила б зловмисникові підробляти токени та отримувати несанкціонований доступ до захищених ресурсів API. Також у файлі вказується порт, на якому запускатиметься серверна частина, — за замовчуванням `5000`. Файл `.env` не повинен потрапляти до системи контролю версій і має бути доданий до `.gitignore`, оскільки містить конфіденційні дані, специфічні для кожного середовища розгортання [3].

Четвертим кроком є встановлення залежностей проєкту. Для серверної частини у її кореневому каталозі виконується команда `npm install`, яка зчитує перелік залежностей із файлу `package.json` та завантажує їх із реєстру `npm`. Серверна частина використовує `Express` як основний фреймворк для побудови HTTP-сервера та маршрутизації запитів, `Sequelize` і `mysql2` для роботи з базою даних, `bcryptjs` для хешування паролів, `jsonwebtoken` для генерації та верифікації токенів, `multer` для обробки завантаження файлів і `cors` для налаштування відповідних заголовків. Встановлення залежностей клієнтської частини виконується аналогічно у її каталозі й включає `React`, `react-router-dom` для організації навігації та `Vite` як інструмент збірки. Після успішного завершення встановлення всі необхідні бібліотеки розміщуються у відповідних каталогах `node_modules`, і обидві частини проєкту готові до запуску.

П'ятим кроком є безпосередній запуск програмного комплексу. Серверна частина запускається командою `npm start` у кореневому каталозі серверного проєкту. Під час першого запуску `Sequelize` за потреби виконує синхронізацію моделей із базою даних: для кожної моделі перевіряється наявність відповідної таблиці, і якщо вона відсутня, таблиця створюється відповідно до описаної схеми. Це означає, що ручне написання SQL-скриптів для створення структури бази даних не є обов'язковим — достатньо коректно описати моделі `Sequelize`, після чого структура формується автоматично. Після успішного запуску в консолі відображається повідомлення про успішне підключення до бази даних та номер порту, на якому доступне API. Повідомлення про помилки на цьому етапі найчастіше свідчать про некоректно заповнений файл `.env` або недоступність сервера `MySQL`.

Клієнтська частина у режимі розробки запускається командою `npm run dev` у каталозі клієнтського проєкту. `Vite` запускає вбудований HTTP-сервер із підтримкою гарячого оновлення модулів: будь-яка зміна у вихідному коді миттєво відображається у браузері без ручного перезавантаження сторінки, що суттєво прискорює процес розробки [10]. Для розгортання у виробничому

середовищі клієнтська частина збирається командою `npm run build`, яка компілює та мінімізує вихідний код у статичні файли, що розміщуються у каталозі `dist/`. Ці файли можуть бути розгорнуті на будь-якому вебсервері або обслуговуватися безпосередньо серверною частиною Node.js як статичні ресурси.

Для першого входу в адміністративну панель необхідно мати обліковий запис із роллю `admin` у таблиці `users`. Оскільки реєстрація через публічну форму сайту завжди присвоює роль `customer`, перший адміністраторський запис створюється окремим скриптом або безпосереднім додаванням запису до бази даних із попередньо обчисленим хешем пароля. У базі даних зберігається саме хеш пароля, а не його відкритий текст, що є однією з базових вимог безпеки [22]. Після створення адміністративного облікового запису система повністю готова до роботи. Завдяки використанню стандартних технологій Node.js та MySQL програмний комплекс може бути розгорнутий на будь-якому сучасному хостинговому середовищі без прив'язки до конкретного постачальника послуг [11, 12].

3.2 Структура та основні елементи інтерфейсу

Інтерфейс розробленої вебсистеми побудований з урахуванням принципів зручності використання, простоти навігації та адаптивності для різних типів пристроїв. Система містить окремі інтерфейси для користувачів та адміністраторів, кожен із яких орієнтований на виконання відповідних функцій. Загальний вигляд головної сторінки вебсистеми зображено на рис.3.2, рис.3.3, рис.3.4.

Головна сторінка вебсистеми є основною точкою входу користувача до інтернет-магазину та забезпечує доступ до основних функціональних можливостей системи. У верхній частині сторінки розміщено навігаційну панель, яка містить посилання на основні розділи сайту: каталог продукції, сторінку товарів у наявності, блог та контакти. Також у хедері реалізовано перехід до особистого кабінету користувача та кошика покупця. Центральну частину верхнього блоку займає банер із короткою інформацією про магазин та

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

кнопкою переходу до каталогу продукції. На рис.3.2. наведено верхню частину головної сторінки вебсистеми.

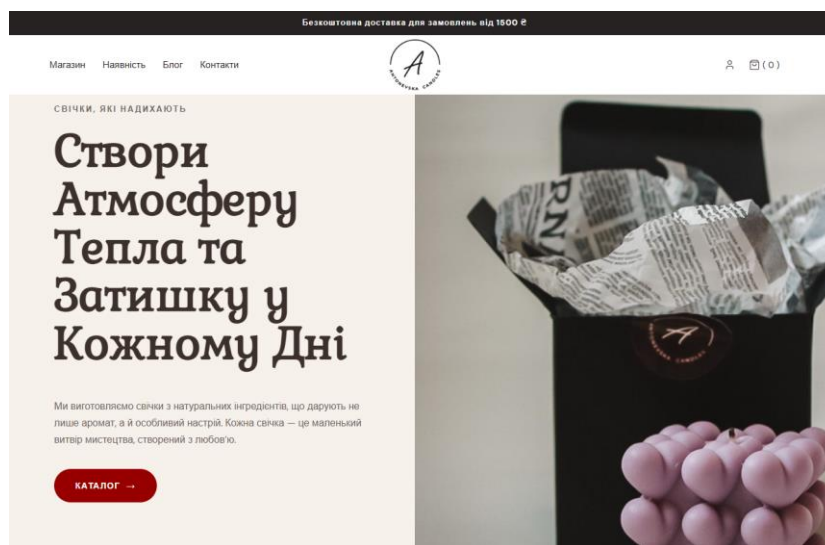


Рис. 3.2. Верхня частина головної сторінки вебсистеми.

Середня частина головної сторінки містить блок категорій товарів, реалізований у вигляді горизонтального слайдера. Таке рішення забезпечує компактне відображення основних категорій продукції та дозволяє користувачеві швидко перейти до потрібного розділу каталогу. До переліку категорій належать формові свічки, контейнерні свічки, свічки з вощини, аромасаше та гіпсові підставки.

Нижче розташовано секцію «Новинки», у якій автоматично відображаються останні додані товари. Формування цього блоку здійснюється на основі даних, отриманих із бази даних, що забезпечує його автоматичне оновлення після додавання нових позицій через адміністративну панель. Для кожного виробу відображаються фотографія, назва, категорія, вартість та кнопка переходу до детальної сторінки товару. Такий підхід дозволяє привернути увагу користувачів до нових товарів і спрощує навігацію каталогом.

На рис. 3.3 показано центральну частину головної сторінки, де розміщено блок категорій товарів та секцію нових надходжень.

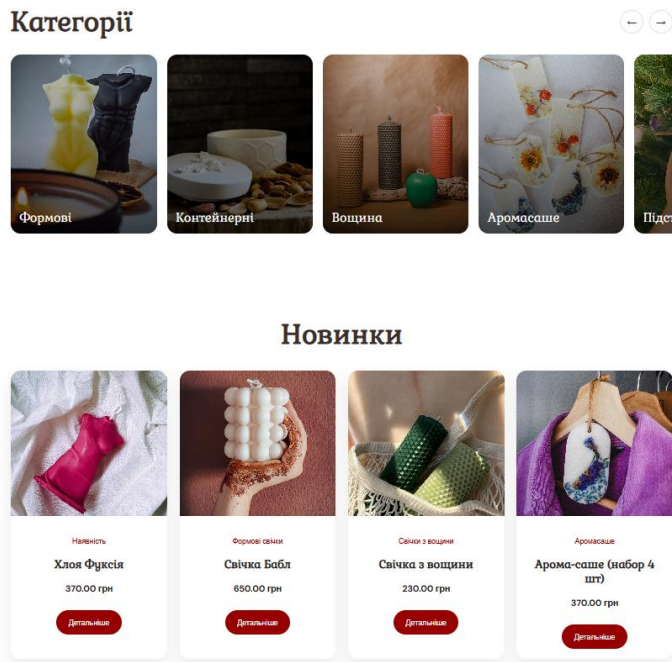


Рис. 3.3. Середня частина головної сторінки вебсистеми.

Нижня частина сторінки містить інформаційні елементи, спрямовані на підвищення зручності використання вебсистеми. Для відповідей на типові запитання користувачів реалізовано блок поширених запитань (FAQ), який містить інформацію щодо продукції, доставки та оформлення замовлень. Наповнення цього розділу здійснюється через адміністративну панель, що дозволяє оперативно оновлювати інформацію без внесення змін до програмного коду.

Завершує сторінку підвал сайту, де розміщено контактну інформацію, навігаційні посилання та додаткові відомості про діяльність магазину. Підвал забезпечує швидкий доступ до основних розділів вебсистеми, контактних даних і сторінок із додатковою інформацією, що покращує навігацію та загальну зручність користування сайтом.

На рис. 3.4 представлено нижню частину головної сторінки, яка містить блок поширених запитань та підвал сайту з контактною інформацією.

Поширені запитання

Чи можна повернути або обміняти товар?

+

Які є способи доставки?

+

Чи можна зробити велике замовлення для свята?

+

Чи переодично ви свічки?

+

У вас усе під замовлення чи є товари в наявності?

-

Більшість продукції виготовляється вручну під замовлення. Іноді є невелика кількість готових свічок – їх можна переглянути в розділі «Наявність».

Чи безпечні свічки для дому з дітьми?

+

З яких матеріалів виготовляються свічки?

+



Рис. 3.4. Нижня частина головної сторінки вебсистеми.

Сторінка каталогу товарів є центральним модулем інтерфейсу розробленої системи електронної комерції, який забезпечує повноцінну взаємодію користувача з асортиментом авторської продукції. Архітектура цієї сторінки розроблена з урахуванням сучасних вимог до користувацького досвіду і містить розвинений інструментарій для сегментації, фільтрації та сортування виробів. Основу навігаційної структури каталогу складає чіткий поділ на тематичні категорії, серед яких представлені формові, контейнерні та вошинні свічки, аромасаше, підставки, а також спеціалізована колекція товарів «Магічний світ». На рис. 3.5. наведено сторінку каталогу товарів із реалізованими засобами навігації, фільтрації та перегляду асортименту.

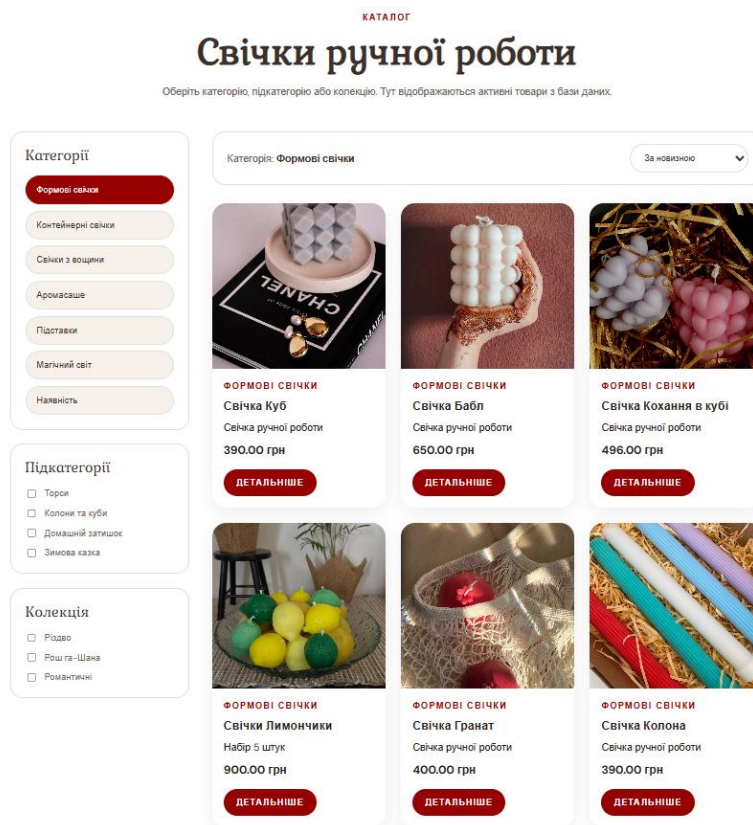


Рис. 3.5. Сторінка каталогу товарів.

Для забезпечення глибокої та точнішої навігації в системі реалізовано багаторівневу структуру, де кожна велика категорія, як-от формові свічки, додатково деталізується на специфічні підкатегорії, що включають торси, колони та куби, вироби для домашнього затишку чи святкові серії на кшталт зимової казки. Крім того, архітектура системи підтримує групування асортименту за критерієм приналежності до тематичних колекцій, що дозволяє користувачеві швидко знаходити продукцію, адаптовану під конкретні події чи свята, зокрема Різдво, Рош га-Шана або романтичні серії.

Кожна позиція у цій зоні презентується у вигляді інтерактивного візуального блока, який містить назву виробу, маркування щодо його приналежності до товарів ручної роботи, актуальну вартість у національній валюті, а також забезпечує миттєвий перехід до картки детального перегляду. Усі ці елементи взаємодіють як єдиний комплекс, що мінімізує час на пошук

необхідного виробу, спрощує шлях користувача до моменту додавання товару в кошик та забезпечує високу технологічну гнучкість платформи.

Сторінка товару призначена для детального ознайомлення користувача з обраним виробом та подальшого оформлення замовлення. У верхній частині сторінки розміщено основну інформацію про товар, включаючи його назву, категорію, опис та актуальну вартість.

Ліва частина інтерфейсу містить галерею зображень товару. Для зручності перегляду реалізовано відображення головного зображення та додаткових фотографій, між якими користувач може перемикатися. Це дозволяє ознайомитися з виробом з різних ракурсів та переглянути окремі деталі його оформлення.

Права частина сторінки містить детальну інформацію про товар та елементи налаштування індивідуальних характеристик. Користувач може обирати доступні кольори, аромати та декоративні елементи залежно від типу виробу. Для кожного параметра відображаються лише доступні варіанти, налаштовані адміністратором через адміністративну панель.

Однією з основних особливостей сторінки є підтримка динамічного розрахунку вартості. У разі вибору характеристик, для яких передбачена додаткова вартість, система автоматично виконує перерахунок ціни без перезавантаження сторінки. Це дозволяє користувачу одразу бачити підсумкову вартість обраної конфігурації товару.

Після вибору необхідних характеристик користувач може додати товар до кошика та перейти до подальшого оформлення замовлення. Такий підхід забезпечує зручну взаємодію з продукцією та спрощує процес придбання виробів ручної роботи. На рис. 3.6. показано картку товару з галереєю зображень, описом виробу та елементами вибору індивідуальних характеристик.

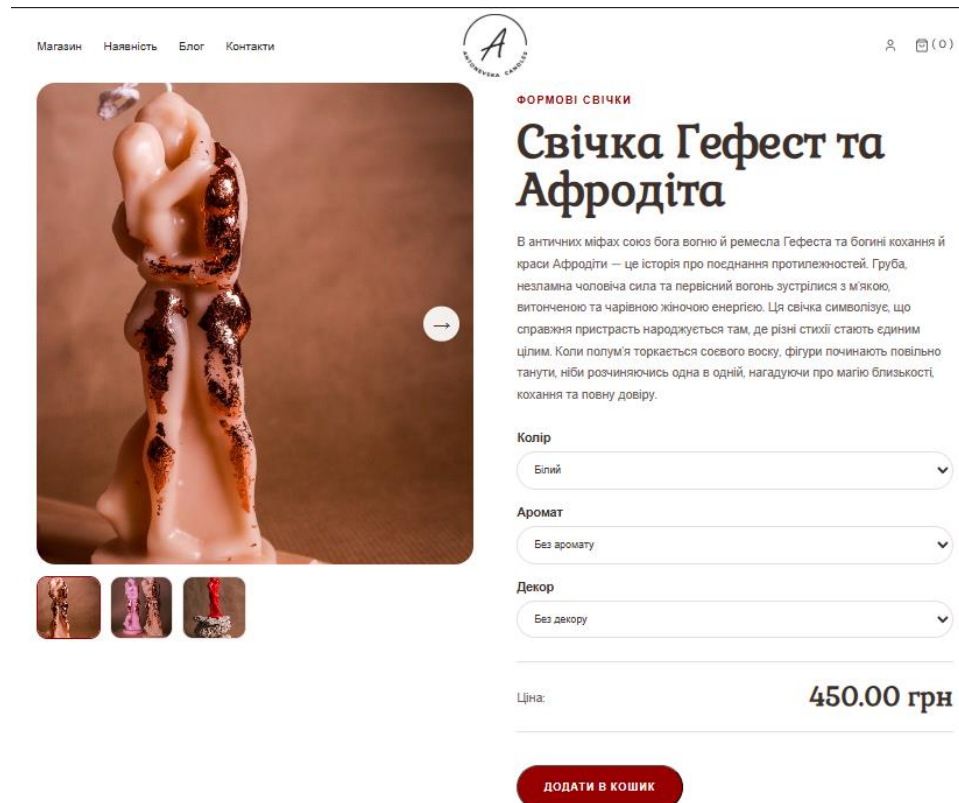


Рис. 3.6. Картка товару.

Сторінка кошика призначена для перегляду та редагування сформованого замовлення перед його остаточним оформленням. У цьому розділі користувач може переглянути перелік обраних товарів, їх характеристики, кількість та підсумкову вартість.

Для кожної позиції відображається назва товару, основне зображення, обрані параметри, ціна та кількість одиниць у замовленні. Користувач має можливість змінювати кількість товарів або видаляти окремі позиції без необхідності повторного формування кошика. Після кожної зміни система автоматично виконує перерахунок загальної вартості замовлення.

У правій частині сторінки розташована форма оформлення замовлення. Для створення замовлення користувач повинен вказати контактні дані, номер телефону, адресу доставки та за необхідності залишити коментар до замовлення. Перед відправленням виконується перевірка коректності введених даних, що дозволяє уникнути створення неповних або помилкових замовлень.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Після успішного заповнення форми та підтвердження замовлення система зберігає інформацію в базі даних і створює новий запис у таблиці замовлень. Одночасно зберігаються відомості про склад замовлення, обрані характеристики товарів та контактні дані покупця. Після успішного створення замовлення кошик автоматично очищується, а користувач отримує повідомлення про успішне виконання операції. На рис.3.7. представлено сторінку кошика, що забезпечує перегляд обраних товарів і оформлення замовлення.

Рис. 3.7. Сторінка кошика.

Сторінка особистого кабінету призначена для роботи зареєстрованих користувачів із власними даними та замовленнями. Доступ до цього розділу надається після успішної авторизації в системі.

У верхній частині сторінки відображається основна інформація про користувача, зокрема ім'я, адреса електронної пошти та контактний номер телефону. Для забезпечення актуальності інформації реалізовано можливість редагування персональних даних через спеціальну форму. Під час збереження змін виконується перевірка коректності введених даних, що дозволяє запобігти збереженню некоректної інформації.

Однією з основних функцій особистого кабінету є перегляд історії замовлень. Користувач може отримати інформацію про всі раніше оформлені замовлення, їх склад та поточний статус виконання. Це дозволяє контролювати процес обробки замовлень та відстежувати їх виконання без необхідності додаткового звернення до адміністратора магазину.

Усі дані особистого кабінету отримуються з бази даних через серверну частину системи та відображаються відповідно до облікового запису авторизованого користувача. Такий підхід забезпечує персоналізований доступ до інформації та підвищує зручність взаємодії користувача з вебсистемою. На рис. 3.8 наведено сторінку особистого кабінету користувача, яка відображається після успішної авторизації в системі.

Безкоштовна доставка для замовлень від 1500 ₴

Магазин

Наявність

Блог

Контакти



🔍

👤

🛒 (0)

ОСОБИСТИЙ КАБІNET

Вийти

Вітаємо, Марія

Тут можна переглянути дані акаунта, оновити контактний телефон та переглянути історію замовлень.

Мої дані

Ім'я

Марія

Email

zip2221_lmo@student.zhu.edu.ua

Телефон

+380111111112

Зберегти дані

Мої замовлення

Замовлення #10

800.00 грн

Статус: Відправлено

1 товарів

Свічка Каліюпа

800.00 грн

Кількість: 1 × 800.00 грн

Копір: Білий

Аромат: Ванільна карамель

Декор: Срібна поталь

Замовлення #7

496.00 грн

Статус: В обробці

1 товарів

Свічка Кохання в кубі

496.00 грн

Кількість: 1 × 496.00 грн

Копір: Білий

Аромат: Без аромату

Декор: Без декору

Рис. 3.8. Особистий кабінет користувача.

Для керування програмним комплексом реалізовано окрему адміністративну панель. Доступ до неї мають лише користувачі з роллю

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

адміністратора. Після авторизації адміністратор отримує доступ до модулів керування товарами, категоріями, матеріалами, замовленнями, користувачами, відгуками, блогом, банерами та розділом поширених запитань.

На сторінці «Дашборд» відображаються ключові показники роботи магазину, зокрема кількість товарів, замовлень, користувачів та відгуків. Також реалізовано швидкий перехід до найбільш використовуваних адміністративних функцій, що спрощує роботу власника магазину та скорочує час на виконання типових операцій. На рис. 3.9 наведено головну сторінку адміністративної панелі.

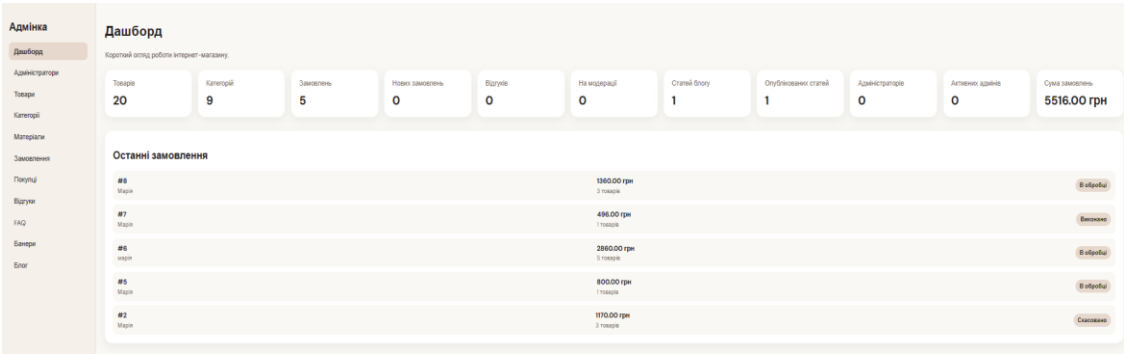


Рис. 3.9. Головна сторінка адміністративної панелі.

Для структурування каталогу реалізовано модуль керування категоріями. Адміністратор може створювати нові категорії товарів, редагувати їх назви, описи та керувати їх відображенням на сайті. Використання категорій дозволяє впорядкувати каталог продукції та спрощує навігацію для користувачів. На рис. 3.10 представлено сторінку керування категоріями.

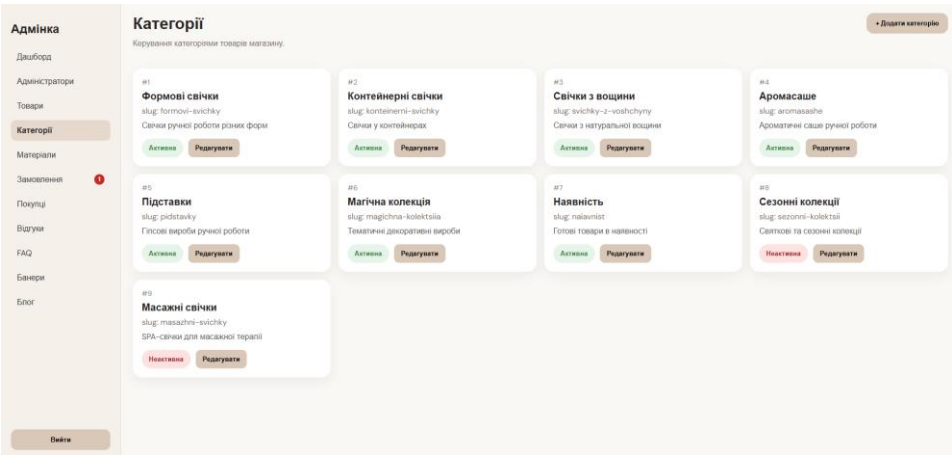


Рис. 3.10. Сторінка керування категоріями.

Окремий модуль призначений для керування матеріалами та характеристиками продукції. Через нього адміністратор може додавати нові кольори, аромати та декоративні елементи, які надалі використовуються під час створення товарів. Для кожного параметра може бути визначена додаткова вартість, що використовується системою під час автоматичного розрахунку ціни виробу. На рис. 3.11 представлено сторінку керування матеріалами.

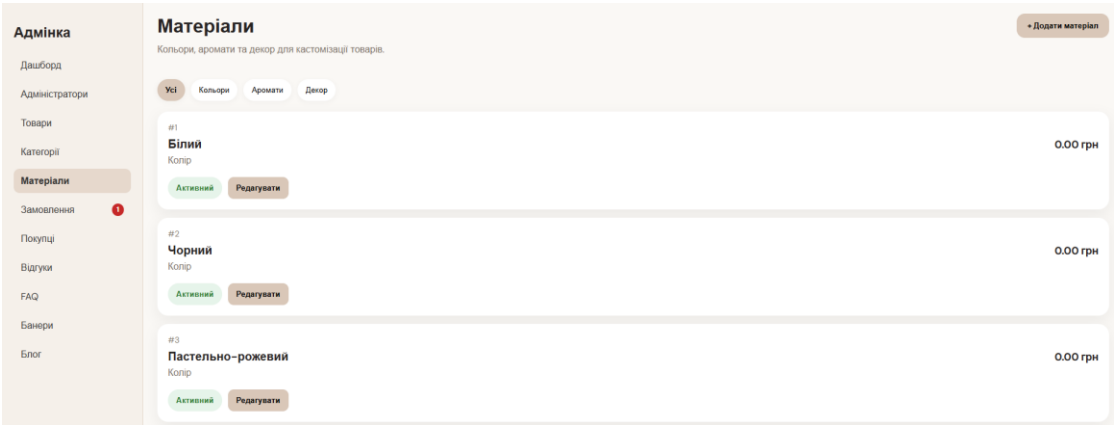


Рис.3.11. Сторінка керування матеріалами.

Модуль керування товарами дозволяє створювати нові товари, редагувати їх характеристики, змінювати вартість та керувати зображеннями. Для кожного товару підтримується можливість завантаження декількох фотографій та налаштування доступних параметрів кастомізації. На рис. 3.12 представлено сторінку керування товарами.

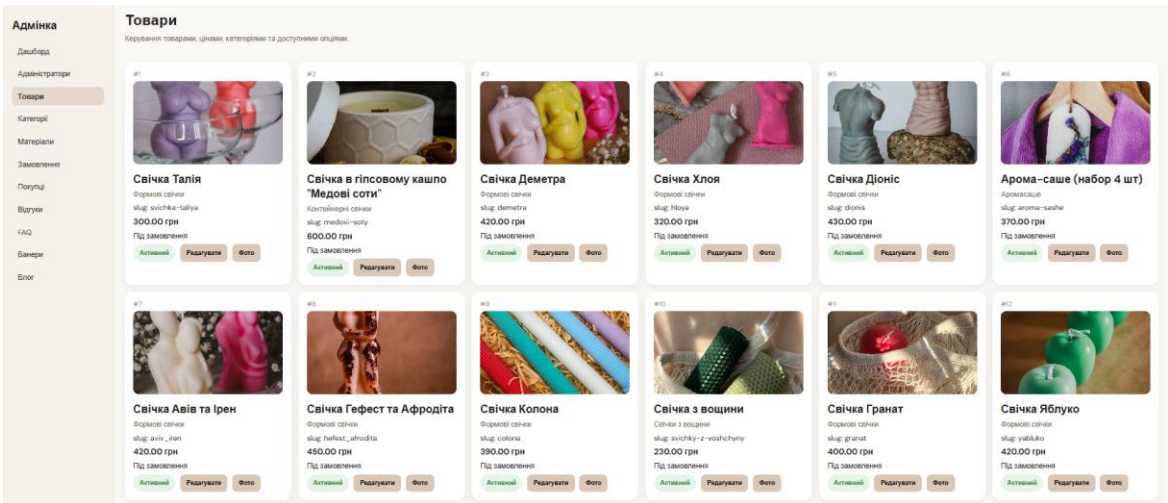


Рис. 3.12. Сторінка керування товарами.

Для роботи із замовленнями реалізовано окремий адміністративний модуль, який забезпечує перегляд інформації про покупців, склад замовлень та зміну статусів виконання. Це дозволяє контролювати процес обробки замовлень на всіх етапах їх виконання. На рис. 3.13 представлено сторінку керування замовленнями.

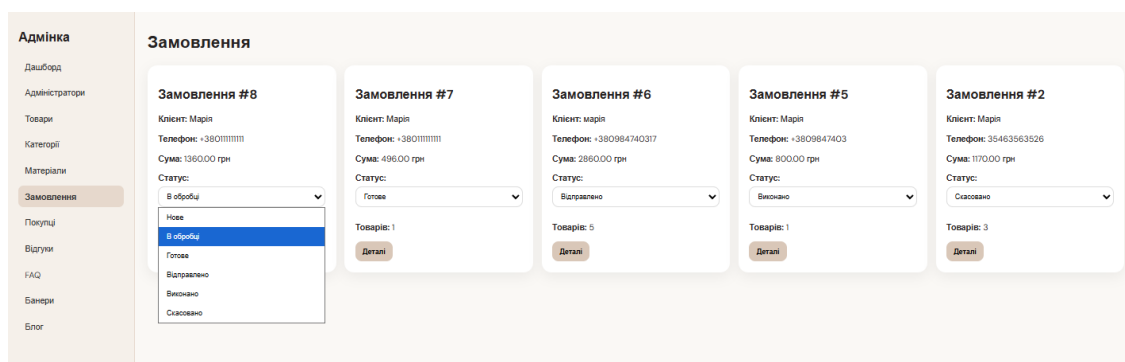


Рис.3.13. Сторінка керування замовленнями.

Для керування користувачами реалізовано окремий адміністративний модуль контролю користувачів. Він дозволяє переглядати зареєстрованих користувачів системи, контролювати їх активність та за необхідності обмежувати доступ до функціональних можливостей вебсистеми. На рис. 3.14 представлено сторінку керування користувачами.

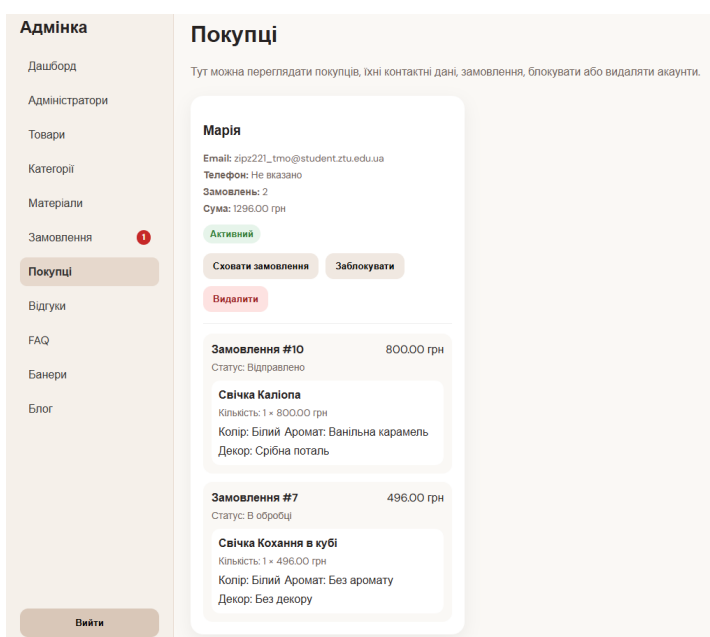


Рис. 3.14. Сторінка керування користувачами.

Для забезпечення зворотного зв'язку з покупцями реалізовано модуль керування відгуками. Адміністратор може переглядати залишені відгуки, виконувати їх модерацію та приймати рішення щодо публікації на сайті. Це дозволяє підтримувати актуальність інформації та запобігати розміщенню небажаного контенту. На рис. 3.15 представлено сторінку керування відгуками.

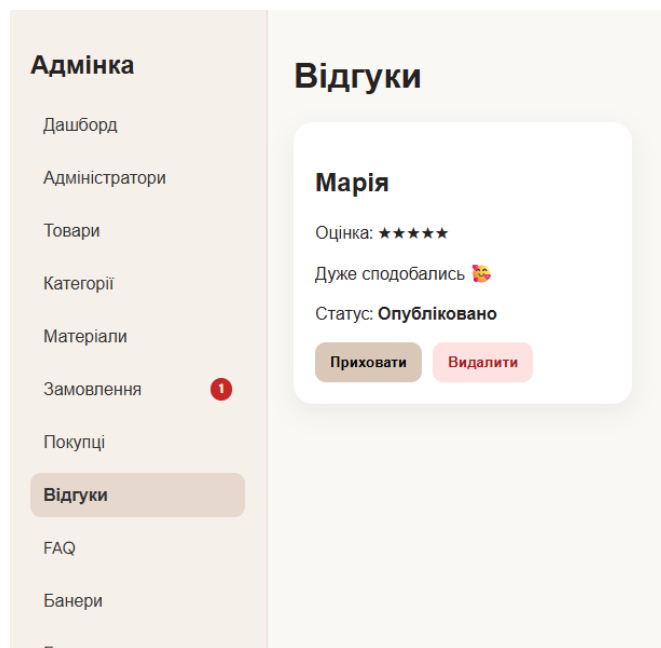


Рис. 3.15. Сторінка керування відгуками.

Інформаційне наповнення вебресурсу забезпечується за допомогою модулів керування блогом, банерами та розділом поширених запитань. Через адміністративну панель власник магазину може створювати нові статті блогу, оновлювати банери головної сторінки та редагувати перелік запитань і відповідей у розділі FAQ без внесення змін до програмного коду системи.

На рис. 3.16 показано модуль керування розділом FAQ, який призначений для додавання, редагування та видалення відповідей на поширені запитання користувачів. Використання цього модуля сприяє підвищенню зручності роботи із вебсистемою та зменшує кількість типових звернень до адміністратора.

Рис. 3.16. Сторінка редагування блоку «Часті запитання».

На рис. 3.17 наведено приклад роботи модуля керування блогом, який забезпечує створення, редагування, публікацію та видалення інформаційних статей. Це дозволяє адміністратору підтримувати актуальність контенту вебсистеми та наповнювати сайт новими матеріалами.

Рис. 3.17. Сторінка для розміщення банерів на головній сторінці.

На рис. 3.18 наведено приклад роботи модуля керування блогом, який забезпечує створення, редагування, публікацію та видалення інформаційних статей. Це дозволяє адміністратору підтримувати актуальність контенту вебсистеми та наповнювати сайт новими матеріалами.

Рис. 3.18. Сторінка для написання та публікації статей для розділу Блог.

3.3. Тестування програмного комплексу.

Після завершення розробки вебсистеми було проведено тестування основних функціональних можливостей програмного комплексу з метою перевірки коректності роботи реалізованого функціоналу та відповідності системи поставленим вимогам.

Під час тестування особливу увагу було приділено перевірці роботи механізмів реєстрації та авторизації користувачів. Було перевірено можливість створення нового облікового запису, виконання входу до системи із коректними обліковими даними, а також реакцію системи на введення неправильного логіна або пароля. Результати тестування підтвердили коректну роботу механізмів автентифікації та розмежування прав доступу між покупцями та адміністраторами.

Окремо було проведено тестування функціоналу роботи з каталогом товарів. Перевірено коректність відображення товарів, відкриття сторінки окремого виробу, перегляд додаткових зображень та роботу механізму вибору характеристик товару. Під час тестування підтверджено правильність автоматичного перерахунку вартості виробу залежно від обраних параметрів.

Наступним етапом стало тестування роботи кошика та оформлення замовлень. Було перевірено додавання товарів до кошика, зміну кількості позицій, видалення товарів та формування замовлення. Також виконано

перевірку правильності збереження інформації про склад замовлення, обрані характеристики товарів та контактні дані покупця. Після успішного оформлення замовлення підтверджено автоматичне очищення кошика та створення нового запису в базі даних.

Для зареєстрованих користувачів було протестовано функціонал особистого кабінету. Під час тестування перевірено коректність відображення персональних даних користувача, можливість редагування контактної інформації та перегляду історії замовлень. Результати тестування підтвердили правильність роботи механізму взаємодії між обліковим записом користувача та його замовленнями.

Також проведено тестування адміністративної панелі. Перевірено роботу модулів керування товарами, категоріями, матеріалами, замовленнями, користувачами, відгуками, банерами, блогом та розділом поширених запитань. Особливу увагу приділено перевірці зміни статусів замовлень, створенню нових товарів, завантаженню зображень та керуванню інформаційним наповненням сайту. Усі адміністративні функції відпрацювали відповідно до поставлених вимог.

Результати тестування основних функціональних можливостей системи наведено в таблиці 3.1.

Таблиця 3.1.

Результати тестування програмного комплексу

Функція	Очікуваний результат	Результат
Реєстрація користувача	Створення нового облікового запису	Успішно
Авторизація користувача	Вхід до системи	Успішно

продовж. табл. 3.1.

Перегляд каталогу	Відображення товарів	Успішно
Перегляд товару	Відображення інформації про товар	Успішно
Вибір характеристик	Зміна параметрів товару	Успішно
Розрахунок вартості	Автоматична зміна ціни	Успішно
Робота кошика	Додавання та редагування товарів	Успішно
Оформлення замовлення	Створення замовлення в системі	Успішно
Особистий кабінет	Перегляд та редагування даних	Успішно
Історія замовлень	Відображення замовлень користувача	Успішно
Керування товарами	Створення та редагування товарів	Успішно
Історія замовлень	Відображення замовлень користувача	Успішно
Керування товарами	Створення та редагування товарів	Успішно
Керування замовленнями	Зміна статусів замовлень	Успішно
Керування користувачами	Блокування та активація користувачів	Успішно

Під час проведення тестування використовувався підхід функціонального тестування, який передбачає перевірку відповідності роботи системи встановленим вимогам. Для кожної функціональної можливості було сформовано окремий тестовий сценарій із визначенням очікуваного результату та аналізом фактичної поведінки системи.

Тестування виконувалося в умовах, максимально наближених до реальної експлуатації програмного комплексу. Перевірка роботи вебсистеми проводилася

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

на декількох типах пристроїв із різними розмірами екранів. Для тестування використовувалися персональний комп'ютер із широкоформатним монітором, смартфон Apple iPhone 15 Plus та планшет Lenovo Tab M11.

На персональному комп'ютері перевірялася коректність роботи всіх модулів системи, включаючи каталог товарів, кошик, оформлення замовлень, особистий кабінет користувача та адміністративну панель. На мобільних пристроях виконувалося тестування адаптивності інтерфейсу, коректності відображення сторінок та зручності навігації. Було перевірено роботу головної сторінки, каталогу товарів, сторінок окремих виробів, кошика, форми оформлення замовлення, особистого кабінету користувача та основних модулів адміністративної панелі.

Під час тестування адміністративного інтерфейсу перевірялася коректність відображення сторінок керування товарами, замовленнями, категоріями, матеріалами, відгуками та користувачами на різних типах пристроїв. Результати перевірки підтвердили працездатність адміністративної частини системи та можливість виконання основних операцій незалежно від розміру екрана пристрою.

Результати перевірки адаптивності наведено в таблиці 3.2.

Таблиця 3.2.

Результати перевірки адаптивності

Пристрій	Тип пристрою	Роздільна здатність екрана	Результат
Персональний комп'ютер	Робоча станція	1920×1080 px	Успішно
Apple iPhone 15 Plus	Смартфон	1290×2796 px	Успішно
Lenovo Tab M11	Планшет	1920×1200 px	Успішно

У процесі тестування також було перевірено обробку некоректних даних. Зокрема, протестовано поведінку системи при введенні неправильного пароля,

спробі авторизації з порожніми полями, введенні некоректної адреси електронної пошти під час реєстрації та редагуванні профілю з неправильним форматом номера телефону. В усіх випадках система коректно відображала повідомлення про помилку та не дозволяла зберігати некоректні дані. Окремо перевірено роботу механізмів валідації під час оформлення замовлень. Система не дозволяє створювати замовлення без заповнення обов'язкових полів, а також забезпечує перевірку коректності введених даних перед їх збереженням у базі даних.

Також перевірено обробку недоступних об'єктів системи. Товари та матеріали, які були деактивовані адміністратором, не відображаються у каталозі та недоступні для оформлення замовлень, що забезпечує цілісність даних і коректність роботи програмного комплексу.

Додатково було виконано перевірку узгодженості даних між клієнтською та серверною частинами системи. Після створення замовлень, додавання товарів, редагування контенту та зміни статусів інформація коректно відображалася як у базі даних, так і в користувацькому та адміністративному інтерфейсах.

Проведене тестування підтвердило працездатність розробленої вебсистеми та коректність реалізації основних функціональних можливостей. Отримані результати свідчать про відповідність програмного комплексу поставленим вимогам та можливість його практичного використання для автоматизації діяльності малого бізнесу з виготовлення та продажу handmade-свічок.

3.4. Перевірка безпеки вебсистеми

Перевірка безпеки програмного комплексу проводилася за допомогою інструменту OWASP ZAP (Zed Attack Proxy) — спеціалізованого засобу для автоматизованого виявлення вразливостей вебдодатків, що широко використовується як у комерційній, так і в дослідницькій практиці [21, 31]. OWASP ZAP є відкритим проєктом організації OWASP і надає широкий набір інструментів для аналізу безпеки: перехоплення та модифікацію HTTP-трафіку, автоматичне сканування вразливостей, фаззінг параметрів запитів та аналіз

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

заголовків відповідей. Інструмент реалізує кілька режимів сканування. Пасивний режим аналізує перехоплений трафік без активного втручання у роботу системи — він фіксує потенційні проблеми конфігурації та небезпечні патерни у відповідях сервера, не виконуючи жодних атак. Активний режим самостійно надсилає спеціально сформовані запити для перевірки реакції системи на потенційно шкідливі вхідні дані. Для перевірки безпеки розробленої системи використовувалося пасивне сканування, яке є менш деструктивним і достатнім для виявлення типових проблем конфігурації та архітектурних недоліків, не піддаючи систему реальним атакуючим запитам.

Процес виконання сканування наведено на рис. 3.19. Додаток Є. Сканування охопило всі основні маршрути клієнтської частини, відкриті ресурси REST API та заголовки HTTP-відповідей сервера. У ході сканування OWASP ZAP перехоплював запити між браузером і сервером, аналізував структуру відповідей та фіксував будь-які ознаки потенційних вразливостей. Результати аналізу, отримані після завершення сканування, наведено на рис. 3.20. Додаток Є.

За результатами сканування критичних та високоризикових вразливостей виявлено не було. Сканер зафіксував кілька попереджень рівня «інформаційний» та «низький», що стосуються налаштування окремих HTTP-заголовків безпеки. Зокрема, інструмент рекомендував розглянути можливість додавання заголовка Content-Security-Policy, який обмежує перелік дозволених джерел для завантаження скриптів, стилів та інших ресурсів сторінки і є ефективним засобом додаткового захисту від XSS-атак на рівні браузера. Також було рекомендовано заголовок Permissions-Policy, що дозволяє обмежити доступ сторінки до окремих браузерних API — камери, мікрофону, геолокації тощо. Відсутність цих заголовків не є вразливістю в класичному розумінні, а лише відсутністю додаткового захисного шару. Їхнє впровадження є рекомендованою практикою при розгортанні системи у виробничому середовищі з широкою аудиторією,

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

однак не є критичним для коректної та безпечної роботи системи в поточному контексті.

Захист від несанкціонованого доступу реалізовано через багаторівневий механізм автентифікації та авторизації. Після успішного входу сервер генерує JWT-токен, підписаний секретним ключем із файлу змінних середовища. Токен містить ідентифікатор користувача та його роль і передається клієнтській частині, яка зберігає його у localStorage та включає до заголовка Authorization кожного наступного запиту до захищених ресурсів. На стороні сервера middleware-функція checkAuth верифікує підпис токена та його термін дії при кожному запиті. Якщо токен відсутній, пошкоджений або прострочений, сервер повертає відповідь із кодом 401 і подальша обробка запиту не виконується. Для маршрутів адміністративної панелі додатково застосовується функція checkAdmin, яка перевіряє, що роль у декодованому токені відповідає значенню admin. Такий підхід гарантує, що навіть авторизований покупець не зможе отримати доступ до адміністративних ресурсів API, оскільки перевірка відбувається на серверній стороні незалежно від стану клієнтського інтерфейсу.

Паролі користувачів зберігаються у базі даних виключно у вигляді bcrypt-хешів із коефіцієнтом складності 10 [23]. Алгоритм bcrypt є адаптивним: коефіцієнт складності визначає кількість раундів обчислення, що робить перебір паролів методом грубої сили практично неможливим навіть при наявності спеціалізованого обладнання. На відміну від простих алгоритмів хешування на зразок MD5 або SHA-1, bcrypt додає до кожного хешу унікальну сіль, що виключає можливість використання заздалегідь обчислених таблиць хешів для відновлення паролів. Це означає, що навіть у разі повної компрометації бази даних зловмисник не зможе отримати оригінальні паролі користувачів у прийнятні строки.

Захист від SQL-ін'єкцій забезпечується використанням ORM Sequelize для всіх операцій із базою даних [14]. Sequelize автоматично параметризує запити: вхідні дані передаються окремо від тексту SQL-запиту і ніколи не вбудовуються

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

безпосередньо у рядок команди. Таким чином, навіть якщо поле введення міститиме фрагмент SQL-коду, він буде оброблений як звичайний текстовий рядок, а не виконаний базою даних. Захист від XSS-атак забезпечується вбудованим механізмом React, який автоматично екранує вміст усіх змінних, що підставляються у JSX-розмітку [8, 30]. Будь-який рядок, що містить HTML-теги або JavaScript-код і отриманий із бази даних або введений користувачем, відображається як звичайний текст — браузер не інтерпретує його як виконуваний код. Це унеможливорює класичний сценарій збереженого XSS, коли зломисник намагається зберегти шкідливий скрипт у базі через форму введення і згодом виконати його у браузері іншого користувача.

Конфіденційність даних при передачі між браузером та сервером забезпечується протоколом HTTPS із SSL-сертифікатом. Шифрування на транспортному рівні унеможливорює перехоплення або підміну даних у мережі, що є особливо важливим при передачі токенів автентифікації та персональних даних покупців. Механізм CORS налаштовано таким чином, щоб API приймало запити лише з дозволених доменів, що унеможливорює несанкціоновані міжсайтові запити з інших ресурсів [26]. База даних MySQL не має прямого доступу з мережі Інтернет і є доступною виключно з серверної частини застосунку на тому самому хості, що усуває можливість прямих мережових атак на базу в обхід серверного рівня.

Комплексне застосування JWT-автентифікації, bcrypt-хешування паролів, ORM-параметризації запитів, вбудованого захисту React від XSS, HTTPS та обмежень CORS формує багаторівневу систему захисту, що відповідає рекомендаціям OWASP Top 10 для сучасних вебдодатків. Кожен із реалізованих механізмів спрямований на нейтралізацію конкретного класу загроз, а їхнє поєднання забезпечує захист системи навіть у випадку, коли окремий механізм виявиться недостатнім у конкретній ситуації. Результати перевірки засобами OWASP ZAP підтвердили, що реалізована система не має критичних вразливостей та готова до виробничого розгортання.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 3

У третьому розділі було розглянуто порядок встановлення, налаштування та використання розробленої вебсистеми інтернет-магазину handmade-свічок. Наведено опис основних елементів користувацького та адміністративного інтерфейсів, а також продемонстровано особливості взаємодії користувачів із функціональними можливостями програмного комплексу.

Проведене функціональне тестування підтвердило коректність роботи основних модулів системи, зокрема механізмів реєстрації та авторизації користувачів, роботи каталогу товарів, кошика, оформлення замовлень, особистого кабінету та адміністративної панелі. Усі перевірені функції відпрацювали відповідно до встановлених вимог. Додатково було виконано перевірку безпеки вебсистеми із застосуванням інструменту пошуку вразливостей у вебсистемах - OWASP ZAP та аналізом реалізованих механізмів захисту.

Результати перевірки підтвердили відсутність критичних вразливостей, а використання JWT-автентифікації, хешування паролів за допомогою bcrypt, ORM Sequelize, HTTPS та CORS забезпечує належний рівень захисту даних користувачів і програмного комплексу. Отримані результати підтвердили працездатність, надійність та готовність розробленої вебсистеми до практичного використання для автоматизації діяльності малого бізнесу з виготовлення та продажу handmade-свічок.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті виконання випускної кваліфікаційної роботи було розроблено вебсистему електронної комерції для продажу handmade-свічок, яка забезпечує автоматизацію основних бізнес-процесів малого крафтового виробництва. Проведений аналіз предметної області дозволив визначити особливості функціонування handmade-магазинів та сформуванати перелік функціональних і нефункціональних вимог до програмного комплексу.

У процесі проєктування було розроблено архітектуру системи, структуру бази даних та алгоритми взаємодії між окремими компонентами програмного забезпечення. Для реалізації програмного комплексу використано сучасний стек вебтехнологій, що включає React, Node.js, Express, MySQL та ORM Sequelize. Обрана архітектура забезпечує масштабованість, зручність супроводу та можливість подальшого розвитку функціональних можливостей системи.

У межах роботи реалізовано клієнтську частину вебсистеми, що включає каталог товарів, сторінки детального перегляду продукції, кошик, механізм оформлення замовлень, систему реєстрації та авторизації користувачів, а також особистий кабінет із можливістю перегляду історії замовлень і редагування персональних даних. Особливістю розробленого рішення є підтримка індивідуального налаштування характеристик товарів з автоматичним перерахунком вартості залежно від обраних параметрів.

Також реалізовано адміністративну панель, яка забезпечує керування товарами, категоріями, матеріалами, замовленнями, користувачами, відгуками, банерами, блогом та розділом поширених запитань. Це дозволяє власнику магазину самостійно адмініструвати вебресурс без внесення змін до програмного коду.

Проведене функціональне тестування підтвердило коректність роботи всіх основних модулів системи. Додатково виконано перевірку безпеки вебдодатка за допомогою інструменту OWASP ZAP, результати якої підтвердили відсутність

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

критичних вразливостей та відповідність реалізованих механізмів захисту базовим вимогам до сучасних вебсистем.

Таким чином, поставлену мету роботи досягнуто, а всі визначені у вступі завдання виконано в повному обсязі. Розроблений програмний комплекс може бути використаний для автоматизації діяльності малого handmade-бізнесу та слугувати основою для подальшого розвитку вебсистеми. Перспективами її вдосконалення є реалізація підтвердження електронної пошти під час реєстрації користувачів, інтеграція сервісів онлайн-оплати, автоматизація взаємодії зі службами доставки, впровадження системи електронних сповіщень про зміну статусу замовлення, а також розширення аналітичних можливостей адміністративної панелі.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ EN ISO 9241-210:2022. Ергономіка взаємодії людина-система. Частина 210. Людино-орієнтоване проєктування інтерактивних систем. – Київ : ДП «УкрНДНЦ», 2022.
2. UML 2.5.1 Specification [Електронний ресурс]. – Режим доступу: <https://www.omg.org/spec/UML/2.5.1/> (дата звернення: 11.04.2026).
3. Martin R. Clean Architecture: A Craftsman's Guide to Software Structure and Design. – Boston : Prentice Hall, 2017. – 432 с.
4. Cantelon M., Harter M., Holowaychuk T., Rajlich N. Node.js in Action. 2nd Edition. – Shelter Island : Manning Publications, 2017. – 392 с.
5. Ramakrishnan R., Gehrke J. Database Management Systems. 4th Edition. – New York : McGraw-Hill Education, 2022. – 1100 с.
6. Coronel C., Morris S. Database Systems: Design, Implementation and Management. 14th Edition. – Boston : Cengage Learning, 2022. – 832 с.
7. Pilone D., Pitman N. UML 2.0 in a Nutshell. 2nd Edition. – Sebastopol : O'Reilly Media, 2023. – 240 с.
8. React Documentation [Електронний ресурс]. – Режим доступу: <https://react.dev/> (дата звернення: 15.04.2026).
9. React Router Documentation [Електронний ресурс]. – Режим доступу: <https://reactrouter.com/> (дата звернення: 18.04.2026).
10. Vite Documentation [Електронний ресурс]. – Режим доступу: <https://vite.dev/> (дата звернення: 21.04.2026).
11. Node.js Documentation [Електронний ресурс]. – Режим доступу: <https://nodejs.org/docs/latest/api/> (дата звернення: 24.04.2026).
12. Express.js Documentation [Електронний ресурс]. – Режим доступу: <https://expressjs.com/> (дата звернення: 28.04.2026).
13. Sequelize Documentation [Електронний ресурс]. – Режим доступу: <https://sequelize.org/> (дата звернення: 02.05.2026).

					<i>ІПЗ.КР.Б – 121 – 16 – ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

- 14.MySQL Documentation [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/> (дата звернення: 05.05.2026).
- 15.npm Documentation [Електронний ресурс]. – Режим доступу: <https://docs.npmjs.com/> (дата звернення: 08.05.2026).
- 16.JSON Web Token Introduction [Електронний ресурс]. – Режим доступу: <https://jwt.io/introduction> (дата звернення: 11.05.2026).
- 17.MDN Web Docs. JavaScript Guide [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (дата звернення: 14.05.2026).
- 18.MDN Web Docs. Fetch API [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API (дата звернення: 18.05.2026).
- 19.MDN Web Docs. Web Storage API [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API (дата звернення: 21.05.2026).
- 20.OWASP Top 10: The Ten Most Critical Web Application Security Risks [Електронний ресурс]. – Режим доступу: <https://owasp.org/www-project-top-ten/> (дата звернення: 24.05.2026).
- 21.OWASP Authentication Cheat Sheet [Електронний ресурс]. – Режим доступу: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html (дата звернення: 27.05.2026).
- 22.OWASP Password Storage Cheat Sheet [Електронний ресурс]. – Режим доступу: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html (дата звернення: 30.05.2026).
- 23.Fielding R. Architectural Styles and the Design of Network-Based Software Architectures : Doctoral Dissertation. – Irvine : University of California, 2000. – 162 с.

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

- 24.Masse M. REST API Design Rulebook. – Sebastopol : O'Reilly Media, 2021. – 116 с.
- 25.Mozilla Developer Network. HTTP Overview [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview> (дата звернення: 02.06.2026).
- 26.MDN Web Docs. CORS (Cross-Origin Resource Sharing) [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS> (дата звернення: 04.06.2026).
- 27.Horoshop. Платформа для створення інтернет-магазинів [Електронний ресурс]. – Режим доступу: <https://horoshop.ua/ua/> (дата звернення: 05.06.2026).
- 28.Shopify Help Center [Електронний ресурс]. – Режим доступу: <https://help.shopify.com/> (дата звернення: 08.06.2026).
- 29.OpenCart Documentation [Електронний ресурс]. – Режим доступу: <https://docs.opencart.com/> (дата звернення: 10.06.2026).
- 30.Wieruch R. The Road to React. – Self-published, 2024. – 350 с. [Електронний ресурс]. – Режим доступу: <https://www.roadtoreact.com/> (дата звернення: 12.06.2026).
- 31.OWASP ZAP Documentation [Електронний ресурс]. – Режим доступу: <https://www.zaproxy.org/docs/> (дата звернення: 10.06.2026).

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					ІПЗ.КР.Б – 121 – 16 – ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця А.1. Таблиця users (Користувачі)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор (первинний ключ)
name	VARCHAR(255) NOT NULL	Ім'я користувача
email	VARCHAR(255) UNIQUE NOT NULL	Електронна пошта (унікальна)
phone	VARCHAR(255)	Номер телефону
password_hash	VARCHAR(255) NOT NULL	Всcrypt-хеш пароля
role	ENUM('admin','customer')	Роль в системі (за замовч. customer)
is_active	TINYINT(1) DEFAULT 1	Статус акаунта (1 — активний, 0 — заблокований)
createdAt	DATETIME NOT NULL	Дата та час створення запису
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця А.2. Таблиця categories (Категорії)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
name	VARCHAR(255) NOT NULL	Назва категорії
slug	VARCHAR(255) UNIQUE NOT NULL	URL-ідентифікатор
parent_id	INT DEFAULT NULL	Не використовується (зарезервовано)
description	TEXT	Опис категорії
is_active	TINYINT(1) DEFAULT 1	Видимість категорії на сайті
createdAt	DATETIME NOT NULL	Дата та час створення
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця Б.1. Таблиця subcategories (Підкатегорії)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
category_id	INT NOT NULL	Зовнішній ключ до categories (CASCADE)
name	VARCHAR(255) NOT NULL	Назва підкатегорії
slug	VARCHAR(255) UNIQUE NOT NULL	URL-ідентифікатор
is_active	TINYINT(1) DEFAULT 1	Видимість підкатегорії на сайті
createdAt	DATETIME NOT NULL	Дата та час створення
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця Б.2. Таблиця collections (Тематичні колекції)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
name	VARCHAR(255) NOT NULL	Назва колекції
slug	VARCHAR(255) UNIQUE NOT NULL	URL-ідентифікатор
is_active	TINYINT(1) DEFAULT 1	Видимість колекції
createdAt	DATETIME NOT NULL	Дата та час створення
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця В.1. Таблица products (Товари)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
category_id	INT NOT NULL	Зовнішній ключ до categories
subcategory_id	INT DEFAULT NULL	Зовнішній ключ до subcategories (необов.)
collection_id	INT DEFAULT NULL	Зовнішній ключ до collections (необов.)
name	VARCHAR(255) NOT NULL	Назва товару
slug	VARCHAR(255) UNIQUE NOT NULL	URL-ідентифікатор
description	TEXT	Повний опис
short_description	TEXT	Короткий опис для каталогу
base_price	DECIMAL(10,2) NOT NULL	Базова ціна
product_type	ENUM('made_to_order','ready')	Тип товару (за замовч. made_to_order)
production_time_min	INT	Мінімальний строк виготовлення (днів)
production_time_max	INT	Максимальний строк виготовлення (днів)
stock_quantity	INT DEFAULT 0	Кількість на складі
options_schema	JSON	Схема доступних опцій кастомізації
is_active	TINYINT(1) DEFAULT 1	Видимість на сайті
createdAt	DATETIME NOT NULL	Дата додавання
updatedAt	DATETIME NOT NULL	Дата останнього редагування

Таблиця Г.1. Таблиця product_images (Зображення товарів)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
product_id	INT NOT NULL	Зовнішній ключ до products (CASCADE)
image_path	VARCHAR(255) NOT NULL	Відносний шлях до файлу зображення
is_main	TINYINT(1) DEFAULT 0	Ознака головного зображення
sort_order	INT DEFAULT 0	Порядок відображення в галереї
createdAt	DATETIME NOT NULL	Дата та час завантаження
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця Г.2. Таблиця materials (Матеріали та характеристики)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
type	ENUM('color', 'fragrance', 'decor')	Тип характеристики
name	VARCHAR(255) NOT NULL	Назва матеріалу або характеристики
description	TEXT	Короткий опис матеріалу
extra_price	DECIMAL(10,2) DEFAULT 0	Додаткова вартість при виборі
stock_quantity	INT DEFAULT 0	Кількість у наявності
is_active	TINYINT(1) DEFAULT 1	Доступність для вибору покупцем
createdAt	DATETIME NOT NULL	Дата та час створення
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця Г.1. Таблиця orders (Замовлення)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
customer_name	VARCHAR(255) NOT NULL	Ім'я покупця
customer_phone	VARCHAR(255) NOT NULL	Номер телефону
customer_email	VARCHAR(255)	Електронна пошта
delivery_address	TEXT	Адреса доставки
comment	TEXT	Коментар до замовлення
total_price	DECIMAL(10,2) DEFAULT 0	Загальна вартість замовлення
status	ENUM('new','processing','ready','shipped','completed','cancelled')	Поточний статус виконання
createdAt	DATETIME NOT NULL	Дата та час оформлення
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця Г.2. Таблиця order_items (Позиції замовлення)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
order_id	INT NOT NULL	Зовнішній ключ до orders (CASCADE)
product_id	INT NOT NULL	Зовнішній ключ до products (CASCADE)
product_name	VARCHAR(255) NOT NULL	Зафіксована назва товару на момент замовлення
quantity	INT DEFAULT 1	Кількість одиниць
unit_price	DECIMAL(10,2) NOT NULL	Ціна за одиницю з урахуванням обраних опцій
total_price	DECIMAL(10,2) NOT NULL	Загальна вартість позиції
selected_options	JSON	Обрані покупцем характеристики
createdAt	DATETIME NOT NULL	Дата та час створення запису
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця Д.1. Таблиця reviews (Відгуки)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
product_id	INT NOT NULL	Зовнішній ключ до products (CASCADE)
customer_name	VARCHAR(255) NOT NULL	Ім'я автора відгуку
rating	INT DEFAULT 5	Числова оцінка від 1 до 5
text	TEXT NOT NULL	Текст відгуку
is_approved	TINYINT(1) DEFAULT 0	Статус модерації (0 — очікує, 1 — опубліковано)
createdAt	DATETIME NOT NULL	Дата та час залишення відгуку
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця Д.2. Таблиця banners (Банери)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
eyebrow	VARCHAR(255)	Надпис над заголовком банера
title	VARCHAR(255) NOT NULL	Заголовок банера
description	TEXT	Підзаголовок або короткий опис
button_text	VARCHAR(255)	Текст кнопки виклику до дії
button_link	VARCHAR(255)	URL-посилання кнопки
image_path	VARCHAR(255)	Відносний шлях до зображення банера
is_active	TINYINT(1) DEFAULT 1	Видимість банера на головній сторінці
createdAt	DATETIME NOT NULL	Дата та час створення
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

Таблиця Е.1. Таблиця blog_posts (Статті блогу)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
title	VARCHAR(255) NOT NULL	Заголовок статті
slug	VARCHAR(255) UNIQUE NOT NULL	URL-ідентифікатор
excerpt	TEXT	Анонс статті (короткий опис)
content	TEXT NOT NULL	Повний текст статті
image_path	VARCHAR(255)	Відносний шлях до зображення статті
is_published	TINYINT(1) DEFAULT 1	Статус публікації
createdAt	DATETIME NOT NULL	Дата та час публікації
updatedAt	DATETIME NOT NULL	Дата та час останнього редагування

Таблиця Е.2. Таблиця faqs (Поширені запитання)

Поле	Тип даних	Опис
id	INT AUTO_INCREMENT	Унікальний ідентифікатор
question	VARCHAR(255) NOT NULL	Текст запитання
answer	TEXT NOT NULL	Текст відповіді
sort_order	INT DEFAULT 0	Порядок відображення
is_active	TINYINT(1) DEFAULT 1	Видимість запитання на сайті
createdAt	DATETIME NOT NULL	Дата та час створення
updatedAt	DATETIME NOT NULL	Дата та час останнього оновлення

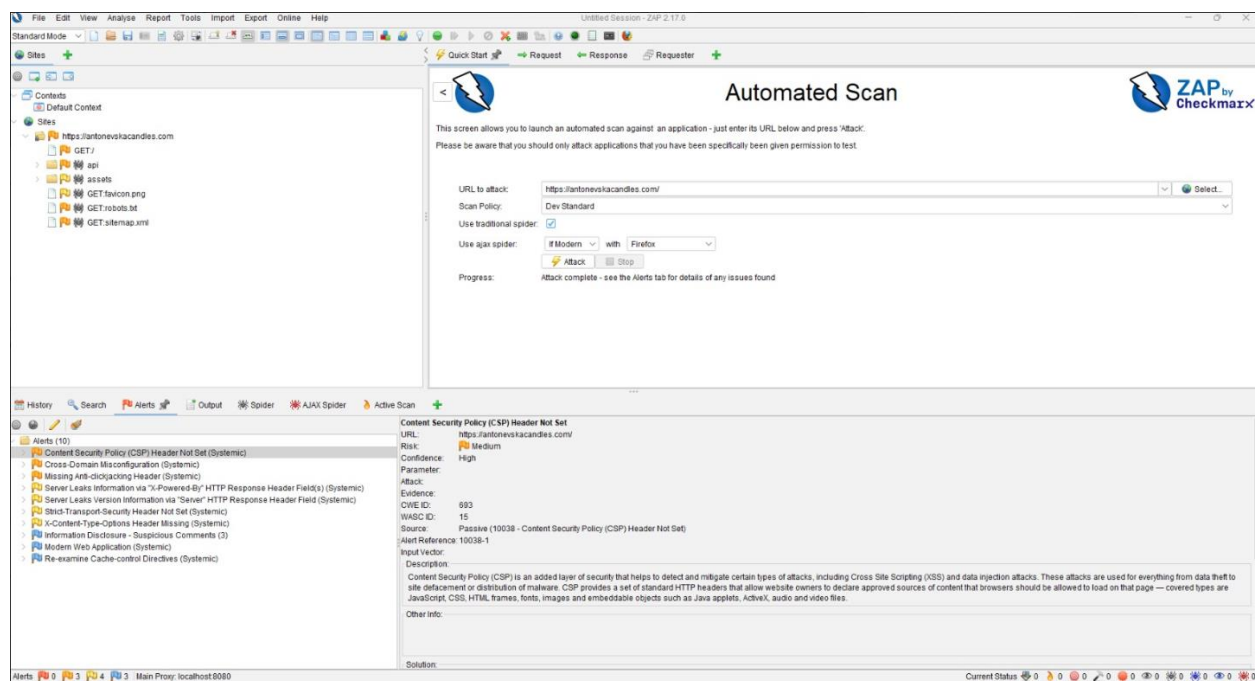


Рис. 3.19 – Виконання автоматизованого сканування вебсистеми в OWASP ZAP

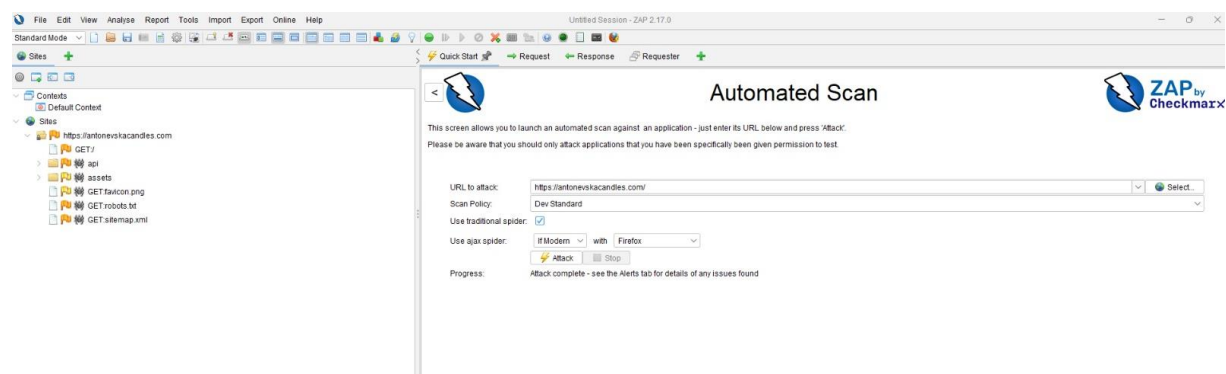


Рис. 3.20 – Результати аналізу безпеки вебсистеми в OWASP ZAP