

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Проектування інформаційної системи інтернет-магазину товарів
для творчості»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2022
Музичин Костянтин Дмитрович
Керівник: Протас Надія Михайлівна
Рецензент: Муравльов Володимир
Вячеславович

Полтава – 2026 року

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

Освітня програма Інформаційні управляючі системи
Спеціальність 126 Інформаційні системи та технології
Рівень вищої освіти перший (бакалаврський)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ **Юрій УТКІН**
«10» червня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Музичину Костянтину Дмитровичу

1. Тема роботи:
«Проектування інформаційної системи інтернет-магазину товарів для творчості»,
керівник роботи к.с.-г.н., доцент, доцент кафедри інформаційних систем та технологій Протас Надія Михайлівна.
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 383 ст
2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року
3. Вихідні дані до роботи стандарти проектування та розробки веб-застосунків, дані офіційних сайтів <https://www.php.net/>, <https://www.mongodb.com/>, <https://getbootstrap.com/>, <https://developer.mozilla.org/> (для JavaScript та Fetch API), середовища прикладних програм OpenServer, MongoDB Compass, Visual Studio Code.
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Розділ 1. Аналіз предметної області та вибір засобів реалізації інтернет-магазину товарів для творчості
Розділ 2. Проектування інформаційної системи інтернет-магазину товарів для творчості
Розділ 3. Програмна реалізація та оцінка ефективності інформаційної системи магазину товарів для творчості
5. Перелік графічного матеріалу: схеми, рисунки, графіки, діаграми за темою та об'єктом дослідження.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
Оцінювання економічної ефективності результатів дослідження	Дивнич О. Д., к. е. н., доцент, доцент кафедри економіки та публічного управління	01.04.2026 р.	29.05.2026 р.

7. Дата видачі завдання «10» червня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Вибір і затвердження теми роботи	02.06.2025 р. – 04.06.2025	
2	Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу	05.06.2025 р. – 10.06.2025 р.	
3	Опрацювання джерел інформації	11.06.2025 р. – 15.06.2025 р.	
4	Збір, вивчення і обробка інформації, необхідної для виконання роботи	16.06.2025 р. – 20.06.2025 р.	
5	Виконання теоретико-методологічного розділу роботи	08.09.2025 р. – 01.11.2025 р.	
6	Виконання дослідницько-аналітичного розділу роботи	19.01.2026 р. – 14.02.2026 р.	
7	Виконання проєктно-рекомендаційного розділу роботи	16.02.2026 р. – 31.03.2026 р.	
8	Розрахунок економічної ефективності результатів дослідження	01.04.2026 р. – 29.05.2026 р.	
9	Оформлення тексту роботи	01.06.2026 р. – 06.06.2026р.	
10	Попередній захист роботи на кафедрі	08.06.2026 р.	
11	Доопрацювання роботи з урахуванням зауважень і пропозицій	09.06.2026 р. – 10.06.2026 р.	
12	Нормоконтроль	11.06.2026 р. – 15.06.2026 р.	
13	Захист кваліфікаційної роботи	16.06.2026 р. - 18.06.2026 р.	

Здобувач вищої освіти

Костянтин МУЗИЧИН

Керівник роботи

Надія ПРОТАС

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ІНТЕРНЕТ-МАГАЗИНУ ТОВАРІВ ДЛЯ ТВОРЧОСТІ	9
1.1 Аналіз ринку онлайн-торгівлі товарами для творчості.....	9
1.2 Огляд та порівняльний аналіз існуючих аналогів інтернет-магазинів	12
1.3 Обґрунтування вибору технологій та середовища розробки	15
1.4 Постановка задачі та формування вимог до системи.....	19
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНТЕРНЕТ- МАГАЗИНУ ТОВАРІВ ДЛЯ ТВОРЧОСТІ	22
2.1 Моделювання бізнес-процесів та побудова діаграм варіантів використання	22
2.2 Проєктування структури бази даних засобами MongoDB.....	26
2.3 Архітектура серверної та клієнтської частин системи.....	29
2.4 Проєктування інтерфейсу користувача та адміністративної панелі....	32
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ МАГАЗИНУ ТОВАРІВ ДЛЯ ТВОРЧОСТІ	36
3.1 Реалізація серверної частини та взаємодії з базою даних	36
3.2 Реалізація клієнтської частини та функціоналу інтернет-магазину	39
3.3 Техніко-економічне обґрунтування ефективності розробленої системи	43
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТКИ.....	54

ВСТУП

Актуальність. Електронна комерція є одним із найбільш динамічних сегментів сучасної цифрової економіки, що стрімко розвивається як у світовому, так і в українському вимірі. За аналітичними даними, обсяг світового ринку e-commerce у 2024 р. перевищив 6 трлн доларів США, а в Україні загальний обсяг онлайн-продажів склав близько 239 млрд гривень – на 25% більше, ніж у 2023 р. [1, 2]. Серед секторів онлайн-торгівлі, що демонструють стабільне зростання, особливе місце займає ринок товарів для творчості: фарби, пензлі, полотна, матеріали для рукоділля та скрапбукінгу користуються зростаючим попитом, а кількість спеціалізованих інтернет-магазинів цього профілю в Україні невідпинно збільшується [3, 4]. В умовах повномасштабного воєнного стану творчі заняття стали не лише хобі, а й засобом психологічної реабілітації, що додатково стимулює попит на відповідні товари в онлайн-каналі.

Попри зростання ринку, значна частина існуючих інтернет-магазинів товарів для творчості в Україні має суттєві обмеження: недостатньо розвинені фільтри за специфічними характеристиками товарів (тип фарб, розмір полотна, матеріал ворсу тощо), відсутність повноцінних адміністративних панелей для оперативного управління каталогом, часткова або відсутня мобільна адаптивність, а також недостатня гнучкість схем даних при описі різноманітного асортименту [8, 9]. Це зумовлює потребу у розробці інформаційної системи, що поєднує сучасний технологічний стек, адаптованість до реалій українського ринку (інтеграція з Новою Поштою, LiqPay, WayForPay) та гнучкість документоорієнтованої бази даних для опису товарів з різноманітними наборами атрибутів.

Метою кваліфікаційної роботи є проєктування та розробка інформаційної системи інтернет-магазину товарів для творчості з використанням технологій PHP та MongoDB в середовищі OpenServer, що забезпечує автоматизацію процесів онлайн-продажу, зручність для покупців і адміністраторів, а також економічну ефективність впровадження.

Відповідно до мети роботи необхідно вирішити наступні *завдання*:

1. Провести аналіз ринку онлайн-торгівлі товарами для творчості в Україні, виконати порівняльний огляд існуючих аналогів та обґрунтувати вибір технологій розробки;
2. Спроекувати архітектуру системи, включаючи моделювання бізнес-процесів, структуру колекцій бази даних MongoDB, трирівневу клієнт-серверну архітектуру та прототипи інтерфейсу;
3. Реалізувати інформаційну систему – серверну та клієнтську частини, базу даних із тестовими даними – та провести техніко-економічне обґрунтування ефективності розробленої системи.

Об'єкт дослідження – процеси проєктування та розробки інформаційних систем електронної комерції з клієнт-серверною архітектурою та документоорієнтованим сховищем даних.

Предмет дослідження – проєктування та реалізація інформаційної системи інтернет-магазину товарів для творчості з використанням PHP, MongoDB та середовища OpenServer, орієнтованої на реалії українського ринку.

Методи досліджень – дослідження базуються на методах програмної інженерії, аналізу вимог та варіантів використання (UML), моделювання бізнес-процесів (BPMN 2.0), проєктування документоорієнтованих баз даних (MongoDB schema design, патерни embedding та referencing), розробки веб-застосунків із використанням патерну MVC, а також техніко-економічного аналізу IT-проєктів (методики NPV, ROI).

Інформаційна база – інтернет-ресурси, наукові статті та довідкові матеріали з веб-технологій (PHP 8.x, MongoDB, Bootstrap 5, JavaScript Fetch API, OpenServer), офіційна документація MongoDB та PHP, статистичні дані про ринок електронної комерції в Україні та світі, а також матеріали аналізу вітчизняних інтернет-магазинів товарів для творчості.

Практична значущість – розроблена інформаційна система інтернет-магазину товарів для творчості забезпечує повний цикл онлайн-продажу: зручний каталог із фільтрами та повнотекстовим пошуком, AJAX-кошик без

перезавантаження сторінки, інтеграцію з вітчизняними сервісами доставки та оплати, а також інтуїтивну адміністративну панель для управління товарами та замовленнями. Гнучка схема даних MongoDB дозволяє описувати різномірний асортимент без модифікації структури бази, а модульна MVC-архітектура спрощує подальше масштабування та підтримку системи, роблячи її придатною для практичного використання малим та середнім бізнесом у сфері e-commerce.

Результати роботи апробовані в рамках наукової конференції здобувачів вищої освіти за результатами науково-дослідної роботи у 2025-2026 рр.

Структура кваліфікаційної роботи логічно пов'язана з задачами досліджень і містить перелік умовних позначень, вступ, три розділи основної частини, висновки, список використаних джерел. Загальний обсяг текстової частини кваліфікаційної роботи складає 53 сторінки формату А4. Вона містить 14 рисунків і 12 таблиць.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ІНТЕРНЕТ-МАГАЗИНУ ТОВАРІВ ДЛЯ ТВОРЧОСТІ

1.1 Аналіз ринку онлайн-торгівлі товарами для творчості

Електронна комерція є одним із найбільш динамічних сегментів сучасної цифрової економіки. Наразі третина населення світу здійснює покупки в інтернеті, що забезпечує оборот галузі в 6 трлн доларів США, які до 2027 року прогнозовано зростуть до 8 трлн. Ключовою перевагою інтернет-торгівлі є можливість здійснювати покупки в будь-який час та з будь-якої точки світу, переглядати широкий асортимент товарів, порівнювати ціни та отримувати персоналізовані рекомендації на основі поведінки користувача [1].

Ринок електронної комерції в Україні демонструє стабільне зростання навіть в умовах повномасштабного воєнного стану. Загальний обсяг ринку електронної комерції за 2024 рік склав 239 мільярдів гривень, що на 25% перевищує показники 2023 року. На даний момент в Україні налічується близько 11 мільйонів активних інтернет-покупців, кожен з яких у середньому здійснює 17 онлайн-покупок на рік із середньою сумою однієї транзакції 1300 гривень. Така тенденція свідчить про суттєве зміщення споживчої поведінки від офлайн-до онлайн-торгівлі [2].

Структура українського ринку e-commerce характеризується домінуванням кількох великих гравців. Найбільша частка продажів в Україні зараз припадає на три маркетплейси: OLX.ua (40%), Rozetka.com.ua (32,5%) та Prom.ua (26,5%). Водночас значна кількість покупців надає перевагу спеціалізованим незалежним інтернет-магазинам, де асортимент та сервіс орієнтовані на конкретну нішу [3]. Саме такою нішею є ринок товарів для творчості – фарб, пензлів, полотен, матеріалів для рукоділля, скрапбукінгу, декупажу та суміжних напрямів.

Особливістю українського ринку товарів для творчості є поєднання широкого асортименту продукції для аматорів та професіоналів із зростаючим попитом з боку нових аудиторій. В умовах воєнного часу інтерес до творчих занять як способу психологічної розрядки суттєво зріс, що підтверджується активністю таких платформ, як Artizo, Masterica, Akvarel та інших [4]. Ці магазини пропонують товари від провідних світових виробників та забезпечують доставку по всій Україні переважно через сервіси «Нова Пошта» та «Укрпошта» – найпоширеніші логістичні рішення для дрібнороздрібної e-commerce в Україні.

Типовий інтернет-магазин товарів для творчості на українському ринку передбачає інтеграцію з популярними вітчизняними платіжними сервісами, зокрема LiqPay (від ПриватБанку) та WayForPay. Україна наразі має один із найбільш диджиталізованих ринків e-commerce: тут поширена схема Direct-to-Consumer, активно використовується соціальна комерція, а покупці зручно здійснюють оплату через Apple Pay, Google Pay та мобільний банкінг. Це ставить певні вимоги до розробки платіжного модуля та інтерфейсу оформлення замовлення [5].

Таблиця 1.1 – Порівняльна характеристика провідних українських онлайн-магазинів товарів для творчості

Назва магазину	Асортимент	Спосіб доставки	Оплата	Особливості
Artizo (artizo.com.ua)	Фарби, полотна, пензлі, матеріали для ліплення	Нова Пошта, кур'єр	Картка, LiqPay	Широкий вибір художніх матеріалів
Akvarel (akvarel.com)	Художні матеріали, канцтовари	Нова Пошта, власні магазини	Картка, після-плата	Функція «віртуальна поличка»
Masterica (masterica.com.ua)	Рукоділья, хенд-мейд, скрапбукінг	Нова Пошта, Укрпошта	Картка, WayForPay	Орієнтація на hand-made
CraftSkrynia (craftskrynia.com.ua)	Матеріали для рукоділья	Нова Пошта	Картка	Доставка по всій Україні

З таблиці 1.1 видно, що більшість гравців ринку використовують схожу інфраструктуру: доставка через «Нова Пошта», оплата карткою або через LiqPay/WayForPay. Це дозволяє визначити мінімальний обов'язковий

функціональний набір для нового інтернет-магазину, орієнтованого на український ринок.

Серед основних характеристик, які визначають конкурентоспроможність інтернет-магазину товарів для творчості в Україні, можна виділити наступні: зручна каталогова структура з фільтрацією за видом творчості, брендом та ціною; коректне відображення залишків на складі; підтримка декількох способів оплати та доставки; адаптивний дизайн для мобільних пристроїв. У 2024 році мобільний шопінг складатиме 72,9% від загального обсягу електронної комерції, що робить мобільну адаптивність обов'язковою вимогою [6].

Для оцінки ємності цільового ринку можна скористатися спрощеною формулою потенційного доходу магазину:

$$D = N \cdot f \cdot \bar{c}, \quad (1.1)$$

де D – потенційний річний дохід (грн),
 N – кількість активних покупців у ніші,
 f – середня частота покупок на рік,
 $\bar{c}$ – середній чек транзакції (грн).

Враховуючи, що загальний середній чек по ринку становить 1300 грн [2], а покупці товарів для творчості зазвичай здійснюють цільові покупки рідше, але на більшу суму, реалістичне значення $\bar{c}$ для даної ніші може становити 800–2000 грн залежно від категорії товару. На рисунку 1.1 показано структуру основних категорій товарів, що реалізуються в українських інтернет-магазинах для творчості.



Рисунок 1.1 – Основні категорії товарів інтернет-магазину для творчості

Таким чином, ринок онлайн-торгівлі товарами для творчості в Україні є перспективним і зростаючим сегментом, що характеризується сформованою аудиторією, стандартизованою платіжно-логістичною інфраструктурою та чіткими вимогами до функціоналу. Це обумовлює актуальність розробки спеціалізованої інформаційної системи інтернет-магазину, адаптованої до реалій вітчизняного ринку.

1.2 Огляд та порівняльний аналіз існуючих аналогів інтернет-магазинів

Перед проєктуванням нової інформаційної системи необхідно провести ретельний аналіз наявних рішень у відповідному ринковому сегменті. Огляд існуючих аналогів дозволяє виявити типові архітектурні рішення, переважаючий функціонал, а також слабкі місця, усунення яких здатне сформувати конкурентну перевагу нового продукту. У процесі аналізу розглянуто чотири провідних українських онлайн-магазини товарів для творчості, що представляють різні сегменти ринку [8].

Artizo (artizo.com.ua) – спеціалізований інтернет-магазин художніх матеріалів, що працює з 2008 року. Пропонує фарби, полотна, пензлі, папір, матеріали для ліплення та широкий спектр супутніх товарів. На головній сторінці реалізовано горизонтальне меню категорій, блок акційних пропозицій та рекомендовані новинки. Картка товару містить детальний опис, характеристики, фотогалерею та пов'язані товари. Реалізовано фільтрацію за брендом, ціновим діапазоном та видом матеріалу. Серед переваг – якісна структура каталогу та швидка доставка по Україні. До недоліків слід віднести відсутність адміністративного розділу для зручного управління асортиментом, а також те, що деякі категорії перетинаються, що ускладнює навігацію [9].

Akvarel (akvarel.com) – мережева структура з фізичними магазинами в 15 містах України та власним онлайн-майданчиком. Характерною особливістю є функція «віртуальна полиця», що дозволяє замовити товар через сайт і забрати його у найближчому фізичному магазині за ціною інтернет-магазину. Платіжна система підтримує постоплату без комісії для клієнта. Сайт пропонує зручний пошук, сортування та фільтри. Слабким місцем є перевантажений інтерфейс: велика кількість категорій і банерів ускладнює сприйняття першої сторінки [9].

Masterica (masterica.com.ua) – магазин, орієнтований переважно на аудиторію рукоділля та hand-made. Пропонує широкий асортимент матеріалів для вишивки, скрапбукінгу, декупажу, виготовлення прикрас. Сайт реалізує реєстрацію та особистий кабінет, систему знижок і акцій, доставку Новою Поштою та Укрпоштою. Перевагами є зрозуміла структура та орієнтованість на конкретну аудиторію. Недолік – застарілий візуальний стиль та відсутність мобільної оптимізації на ряді сторінок [9].

Master Kisti (masterkisti.com.ua) – інтернет-магазин, що спеціалізується виключно на матеріалах для живопису та малювання. Реалізує систему промокодів, програму лояльності та акційні пропозиції. Інтерфейс є лаконічним, але категорійна структура охоплює лише вузьку нішу, що обмежує аудиторію. Зведений порівняльний аналіз перелічених магазинів подано у табл. 1.2.

Таблиця 1.2 – Порівняльний аналіз функціоналу існуючих аналогів

Критерій	Artizo	Akvarel	Masterica	Master Kisti
Особистий кабінет	✓	✓	✓	✓
Фільтрація за брендом/ціною	✓	✓	✓	✓
Мобільна адаптивність	✓	✓	частково	✓
Кошик із збереженням	✓	✓	✓	✓
Система акцій/промокодів	✓	✓	✓	✓
Інтеграція з Новою Поштою	✓	✓	✓	✓
Онлайн-оплата (LiqPay/WayForPay)	✓	✓	✓	✓
Адміністративна панель	не публічна	не публічна	не публічна	не публічна
Оцінки та відгуки на товар	частково	✗	✗	✗
Рекомендовані / схожі товари	✓	частково	✗	✓

Як видно з таблиці 1.2, усі аналізовані магазини реалізують базовий набір функцій: особистий кабінет, кошик, фільтрацію, доставку через «Нова Пошта» та онлайн-оплату. Разом з тим жоден із них не надає публічного доступу до адміністративної панелі, а функції відгуків та рекомендацій товарів реалізовані лише частково. Ці прогалини слід врахувати при проєктуванні власного рішення.

З точки зору якості інтерфейсу, усі розглянуті магазини певною мірою відповідають принципу «трьох кліків» – коли користувач може дістатися будь-якого розділу не більш ніж за три навігаційні дії [10]. Структура інтернет-магазину та навігація мають бути логічно організованими: важливо дотримуватися принципу «3 кліки», щоб відвідувач міг потрапити до будь-якого розділу ресурсу всього за три натискання. Проте на практиці реалізація цього принципу є нерівномірною, особливо в мобільній версії [10]. На рисунку 1.2 проілюстровано типову структуру сторінок інтернет-магазину товарів для творчості, що є спільною для більшості аналізованих рішень.

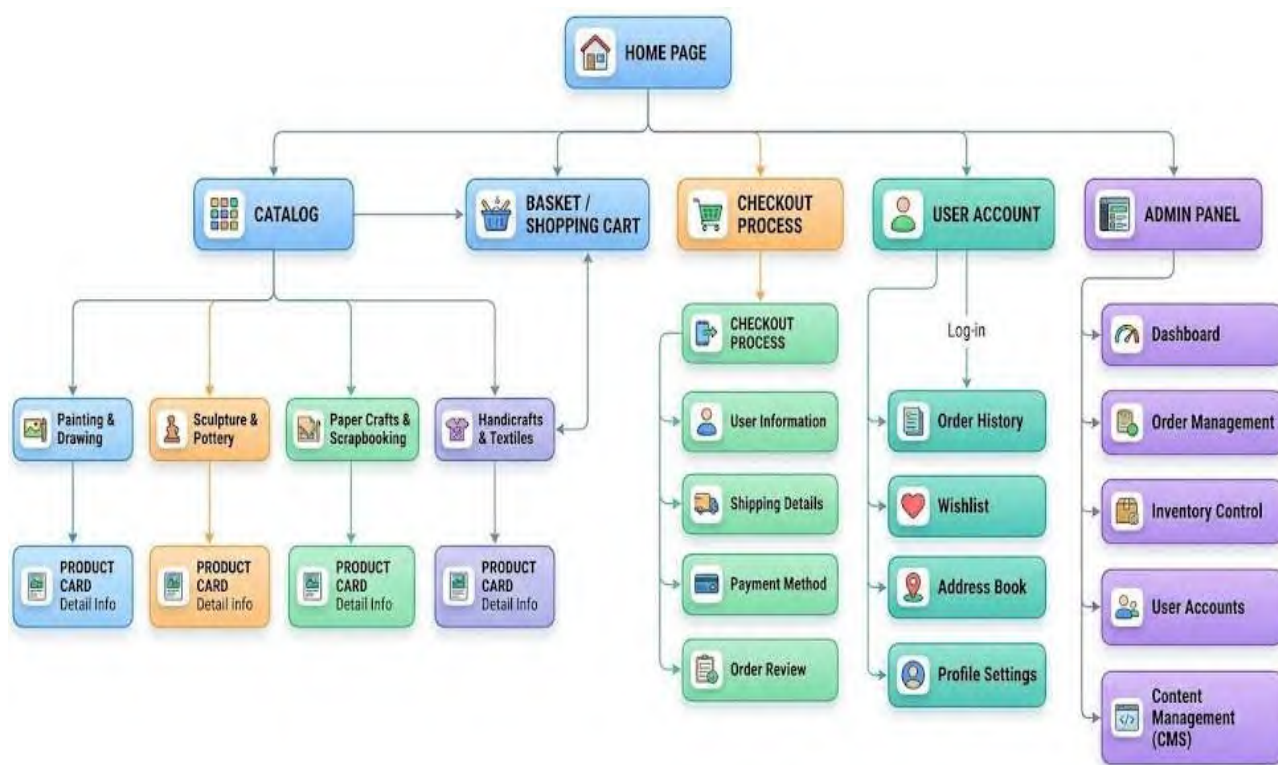


Рисунок 1.2 – Типова структура сторінок інтернет-магазину товарів для творчості

Аналіз аналогів дозволяє визначити індекс покриття функцій (ІПФ) – відношення числа реалізованих функцій із бажаного переліку до їхньої загальної кількості:

$$IPF = \frac{N_{impl}}{N_{total}} \cdot 100\%, \quad (1.2)$$

де N_{impl} – кількість реалізованих функцій у конкретному магазині,

N_{total} – загальна кількість еталонних функцій у переліку.

Для аналізованих аналогів значення ІПФ коливається в межах 60–75%, що свідчить про наявність суттєвого простору для вдосконалення, насамперед у частині рекомендаційних механізмів, системи відгуків та зручності адміністративного управління каталогом [8].

Таким чином, проведений огляд наявних аналогів підтверджує доцільність розробки власного рішення з врахуванням виявлених недоліків. Нова система має забезпечити повноцінний функціональний набір [10]: зручний каталог з фільтрами, особистий кабінет покупця, інтеграцію з популярними вітчизняними сервісами доставки та оплати, а також інтуїтивно зрозумілу адміністративну панель для управління товарами та замовленнями.

1.3 Обґрунтування вибору технологій та середовища розробки

Вибір технологічного стеку є одним із найважливіших рішень на етапі проєктування будь-якої інформаційної системи. Від цього вибору безпосередньо залежать продуктивність, зручність підтримки та подальша масштабованість готового продукту. У даній роботі для реалізації серверної частини інтернет-магазину обрано мову програмування PHP у поєднанні з документоорієнтованою СУБД MongoDB. Середовищем локальної розробки слугує OpenServer –

портативна платформа для веб-розробників під операційну систему Windows [11].

PHP як серверна мова програмування. PHP понад 20 років успішно використовується для створення веб-сайтів і є найпопулярнішою мовою для серверної розробки: більше 80% сайтів в Інтернеті працюють саме на PHP. Серед головних переваг мови слід виділити вбудовану орієнтованість на веб-розробку, широку підтримку різних операційних систем та веб-серверів, відкритий код та безкоштовне використання, а також розвинену спільноту та велику кількість готових бібліотек [11]. PHP панує у компаніях, що потребують простих і недорогих веб-рішень, особливо в сфері e-commerce, що робить цю мову стабільно актуальною. Важливим аргументом на користь PHP є також простота інтеграції з різними СУБД, у тому числі з MongoDB через офіційний драйвер php-mongodb [11]. Для об'єктивного вибору серверної технології розглянуто три альтернативи: PHP, Python (Django/Flask) та Node.js. Таблиця 1.3 містить їх порівняльну характеристику за ключовими критеріями.

Таблиця 1.3 – Порівняльна характеристика серверних технологій для реалізації інтернет-магазину

Критерій	PHP	Python (Django)	Node.js
Орієнтованість на веб	висока (вбудована)	висока (фреймворк)	висока (фреймворк)
Поріг входу	низький	середній	середній
Підтримка MongoDB	офіційний драйвер	офіційний драйвер	офіційний драйвер
Доступність хостингу	дуже широка	середня	середня
Спільнота та документація	дуже велика	велика	велика
Ціна рішення	низька	середня	середня
Інтеграція з OpenServer	нативна	потребує налаштування	потребує налаштування

Як видно з таблиці 1.3, PHP має найнижчий поріг входу, нативну підтримку в середовищі OpenServer та найширший доступ до бюджетного

хостингу, що є важливим у контексті розгортання типового українського інтернет-магазину.

MongoDB як СУБД. Для зберігання даних проєкту обрано MongoDB – документоорієнтовану NoSQL базу даних, що зберігає інформацію у форматі JSON-подібних документів (BSON). Головна перевага MongoDB – гнучкість: не потрібно заздалегідь жорстко визначати структуру даних. Кожен документ у колекції може мати власний набір полів, що суттєво спрощує розробку та дозволяє швидко адаптуватися до вимог проєкту, що змінюються. Це є критично важливим для каталогу товарів інтернет-магазину, де різні категорії товарів мають різні набори атрибутів: фарби характеризуються об'ємом і пігментом, полотна – розміром і видом ґрунтовки, пензлі – матеріалом ворсу й номером [12].

На рисунку 1.3 зображено архітектуру технологічного стеку проєкту, що ілюструє взаємодію між компонентами системи.

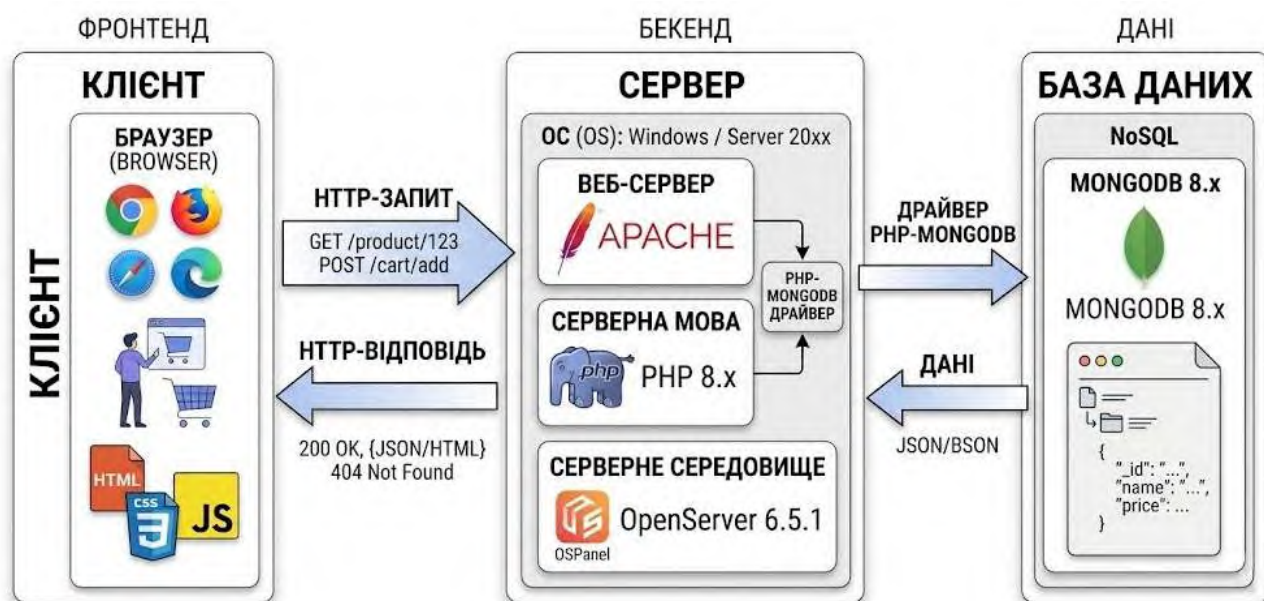


Рисунок 1.3 – Архітектура технологічного стеку інтернет-магазину

У контексті порівняння MongoDB та MySQL для задач інтернет-торгівлі слід зазначити, що реляційна MySQL забезпечує сильну узгодженість даних і добре підходить для суворо структурованих транзакційних систем, тоді як MongoDB краще справляється зі змінними схемами даних і великою кількістю

різнотипних товарних записів [13]. Для нашого завдання перевага MongoDB полягає в тому, що додавання нових характеристик товарів не вимагає міграцій схеми.

В OpenServer вбудовано множину модулів, необхідних веб-розробникам: веб-сервери Apache і Nginx, бази даних MySQL, PostgreSQL і MongoDB, система зберігання Redis, а також PHP-інтерпретатори різних версій. Платформа є портативною – може бути встановлена на знімний носій і використовуватись на різних комп'ютерах без повторного налаштування, що суттєво спрощує навчальний процес і командну розробку [14]. Версія 6.5.1 включає MongoDB 8.x та PHP 8.x, що відповідає сучасним вимогам до продуктивності та безпеки. Для оцінки доцільності вибраного стеку можна скористатися коефіцієнтом технологічної відповідності K_{tv} , що враховує ступінь відповідності обраних технологій вимогам проєкту:

$$K_{tv} = \frac{\sum_{i=1}^n w_i \cdot r_i}{\sum_{i=1}^n w_i}, \quad (1.3)$$

де w_i – вага i -го критерію оцінювання (від 0 до 1),
 r_i – оцінка технології за i -м критерієм (від 0 до 10),
 n – кількість критеріїв.

При застосуванні цієї формули до стеку PHP + MongoDB + OpenServer за критеріями «вартість», «порів входу», «відповідність задачі» та «інтеграція» значення K_{tv} складає не менше 8,5 з 10, що підтверджує обґрунтованість вибору [11].

Таким чином, сукупність PHP, MongoDB та OpenServer утворює повноцінний та збалансований технологічний стек для розробки інтернет-магазину товарів для творчості, орієнтованого на реалії українського ринку. Він забезпечує гнучкість на рівні схеми даних, широку доступність хостингових рішень та зручність локальної розробки без складного налаштування інфраструктури.

1.4 Постановка задачі та формування вимог до системи

Постановка задачі є необхідним етапом перед проєктуванням будь-якої інформаційної системи. Вона дозволяє чітко окреслити цілі розробки, визначити коло користувачів, їх ролі та права доступу, а також сформулювати перелік функціональних і нефункціональних вимог, яким має відповідати готовий продукт [15].

Метою розробки є проєктування та реалізація інформаційної системи інтернет-магазину товарів для творчості, адаптованої до реалій українського ринку. Система повинна забезпечувати зручний процес вибору та придбання товарів для кінцевого покупця, а також ефективне управління каталогом, замовленнями та користувачами для адміністратора. Система реалізується на базі PHP + MongoDB в середовищі OpenServer 6.5.1.

У системі виділяються три основні ролі учасників взаємодії: незареєстрований користувач (гість), зареєстрований покупець та адміністратор. Гість може переглядати каталог товарів, використовувати пошук і фільтри, переглядати картки товарів, а також додавати товари до кошика без реєстрації. Зареєстрований покупець додатково отримує доступ до особистого кабінету, де може переглядати історію замовлень, керувати адресами доставки та редагувати профіль. Адміністратор має повний доступ до панелі управління: додавання, редагування та видалення товарів і категорій, обробка замовлень, керування користувачами [16].

Функціональні вимоги до веб-застосунку для e-commerce включають: можливість додати товар до кошика в один клік через кнопку на картці товару; розрахунок загальної вартості замовлення в кошику з урахуванням знижок і доставки. Окрім функціональних, система має задовольняти ряд нефункціональних вимог. Щодо продуктивності: система має завантажувати веб-сторінку менш ніж за 3 секунди навіть за 100 одночасних користувачів. Вимоги до безпеки передбачають хешування паролів, захист від SQL- та NoSQL-ін'єкцій, а також відповідність положенням Закону України «Про захист

персональних даних» [17]. У таблиці 1.4 наведено повний перелік функціональних вимог до розроблюваної системи [15, 16].

Таблиця 1.4 – Функціональні вимоги до інформаційної системи інтернет-магазину

№	Вимога	Роль
ФВ-01	Перегляд каталогу товарів з фільтрацією за категорією, брендом, ціною	Усі
ФВ-02	Повнотекстовий пошук товарів за назвою та описом	Усі
ФВ-03	Перегляд детальної картки товару (опис, фото, ціна, наявність)	Усі
ФВ-04	Додавання товару до кошика та управління кількістю	Усі
ФВ-05	Реєстрація та авторизація покупця (email + пароль)	Гість
ФВ-06	Оформлення замовлення з вибором способу доставки та оплати	Покупець
ФВ-07	Перегляд історії замовлень у особистому кабінеті	Покупець
ФВ-08	Управління товарами та категоріями (CRUD)	Адміністратор
ФВ-09	Перегляд і обробка замовлень, зміна статусу	Адміністратор
ФВ-10	Управління обліковими записами користувачів	Адміністратор

В технічному завданні також важливо прописати нефункціональні вимоги до продукту: продуктивність, масштабованість, надійність, захист даних – це критично важливі фактори для успішного e-commerce. Вимоги до зручності використання (usability) включають адаптивний дизайн для мобільних пристроїв, мінімально необхідну кількість кроків для оформлення замовлення та зрозумілі повідомлення про помилки [16].

Для оцінки повноти сформованих вимог використовується коефіцієнт покриття варіантів використання K_{vu} :

$$K_{vu} = \frac{N_{req}}{N_{uc}} \cdot 100\%, \quad (1.4)$$

де N_{req} – кількість функціональних вимог, що покривають конкретні варіанти використання,

N_{uc} – загальна кількість виявлених варіантів використання.

Значення $K_{vu} = 100\%$ означає повне покриття, що є цільовим показником для коректного технічного завдання [15]. У сформованому переліку ФВ-01 – ФВ-

10 покривають усі десять виявлених варіантів використання, що свідчить про достатню повноту вимог.

Таким чином, визначені функціональні та нефункціональні вимоги формують вичерпне технічне завдання для подальшого проєктування системи в наступних розділах роботи. Було проведено комплексний аналіз предметної області: досліджено стан і тенденції ринку онлайн-торгівлі товарами для творчості в Україні, виконано порівняльний огляд чотирьох провідних вітчизняних конкурентів, обґрунтовано вибір технологічного стеку PHP + MongoDB + OpenServer, а також сформульовано повний перелік функціональних і нефункціональних вимог до розроблюваної інформаційної системи.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНТЕРНЕТ- МАГАЗИНУ ТОВАРІВ ДЛЯ ТВОРЧОСТІ

2.1 Моделювання бізнес-процесів та побудова діаграм варіантів використання

Проєктування інформаційної системи розпочинається з аналізу та формального опису бізнес-процесів, які вона має автоматизувати. Моделювання бізнес-процесів дозволяє виявити учасників взаємодії, послідовність дій та точки прийняття рішень ще до написання першого рядка коду, що суттєво знижує ризики архітектурних помилок [18]. Для розроблюваного інтернет-магазину товарів для творчості виділено три ключових бізнес-процеси: придбання товару покупцем, обробка замовлення адміністратором та управління каталогом.

Для формального опису бізнес-процесів використовується нотація BPMN 2.0 (Business Process Model and Notation). BPMN – це відкрита стандартна нотація для графічних блок-схем, яка дозволяє чітко та узгоджено документувати бізнес-процеси організації, забезпечуючи залучення всіх зацікавлених сторін, включно з власниками процесів і бізнес-користувачами. Діаграми BPMN будуються за допомогою стандартних елементів: подій (кіл), завдань (прямокутників), шлюзів (ромбів) та потоків послідовності (стрілок) [18].

Основним бізнес-процесом є процес придбання товару покупцем. Він стартує з відвідування покупцем головної сторінки або каталогу, включає пошук і фільтрацію товарів, перегляд картки товару, додавання до кошика, авторизацію або гостьове оформлення, вибір способу доставки та оплати, підтвердження замовлення і отримання повідомлення про його прийняття. До системи надходить замовлення; далі відбувається перевірка наявності товару на складі,

підтвердження замовлення та комплектація. Якщо обраний товар відсутній на складі, система повідомляє покупця та пропонує альтернативи [19].

Процес обробки замовлення адміністратором починається після надходження нового замовлення до панелі управління. Адміністратор переглядає деталі замовлення, перевіряє наявність товарів, підтверджує або скасовує замовлення, після чого змінює статус та повідомляє покупця. Для типового українського магазину процес також включає формування накладної для «Нова Пошта» або іншого сервісу доставки [19]. На рисунку 2.1 зображена BPMN-діаграма процесу придбання товару покупцем, яка демонструє основний сценарій взаємодії покупця з системою від перегляду каталогу до отримання підтвердження замовлення.

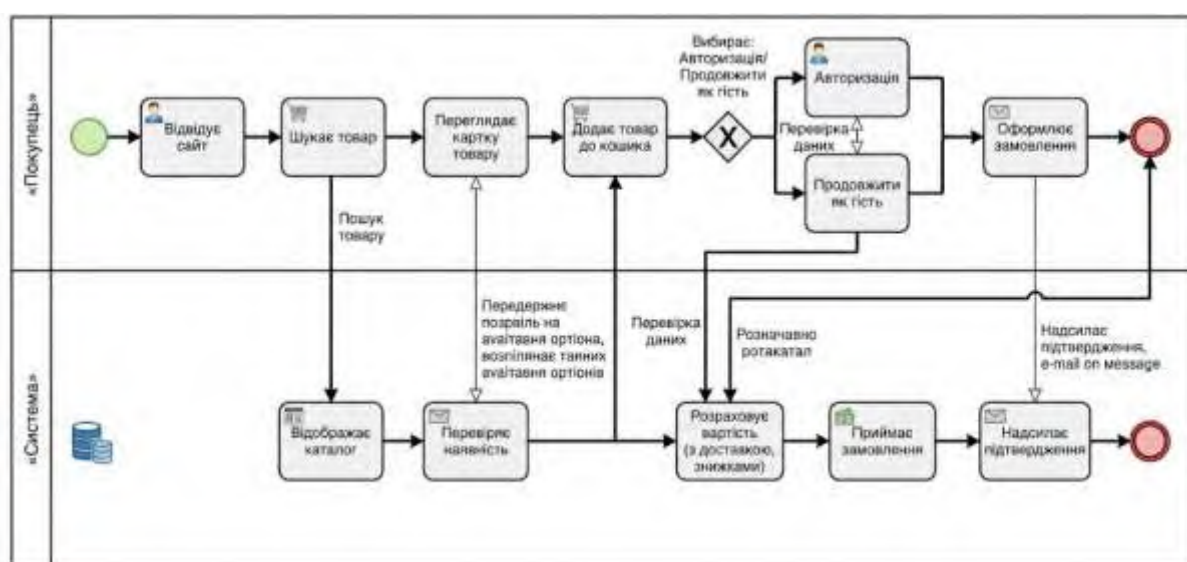


Рисунок 2.1 – BPMN-діаграма процесу придбання товару покупцем

Для опису функціональних вимог із точки зору взаємодії акторів із системою застосовується мова UML та діаграми варіантів використання (Use Case Diagram). Діаграма варіантів використання – це найпростіша з поведінкових діаграм UML, і найкорисніша при висвітленні функціональних особливостей програми, зокрема для людей, що не мають глибоких знань у IT-галузі. Вона застосовується для формалізації функціональних вимог і визначення цілей кожного типу користувача [20].

У таблиці 2.1 наведено опис основних варіантів використання системи в табличному форматі, що є стандартним доповненням до графічної діаграми.

Таблиця 2.1 – Опис основних варіантів використання системи

Іденти-фікатор	Назва ВВ	Актор	Передумова	Основний сценарій
UC-01	Перегляд каталогу	Усі	–	Користувач переходить до розділу каталогу, обирає категорію, застосовує фільтри
UC-02	Пошук товару	Усі	–	Користувач вводить запит у рядок пошуку, система відображає відповідні товари
UC-03	Додавання до кошика	Усі	Товар наявний	Користувач натискає «Додати до кошика» на картці або у списку
UC-04	Реєстрація / авторизація	Гість	–	Гість заповнює форму реєстрації або входить через email і пароль
UC-05	Оформлення замовлення	Покупець	Кошик не порожній	Покупець вводить адресу, обирає доставку та оплату, підтверджує
UC-06	Перегляд замовлень	Покупець	Авторизований	Покупець переходить до «Мої замовлення» в особистому кабінеті
UC-07	Управління товарами	Адміністратор	Авторизований	Адміністратор додає, редагує або видаляє товари та категорії
UC-08	Обробка замовлень	Адміністратор	Авторизований	Адміністратор переглядає нові замовлення та змінює їх статус

З таблиці 2.1 видно, що система підтримує 8 основних варіантів використання, розподілених між трьома акторами. Для оцінки складності системи можна застосувати спрощений індекс кількості варіантів використання на актора:

$$I_{vu} = \frac{N_{uc}}{N_a}, \quad (2.1)$$

де N_{uc} – загальна кількість варіантів використання,

N_a – кількість акторів.

Для розроблюваної системи $I_{vu} = 8/3 \approx 2,67$, що відповідає системам середнього рівня складності з рівномірним розподілом функціоналу між ролями [20]. На рисунку 2.2 зображена UML-діаграма варіантів використання, що графічно відображає зв'язки між акторами та варіантами використання, зафіксованими в таблиці 2.1.

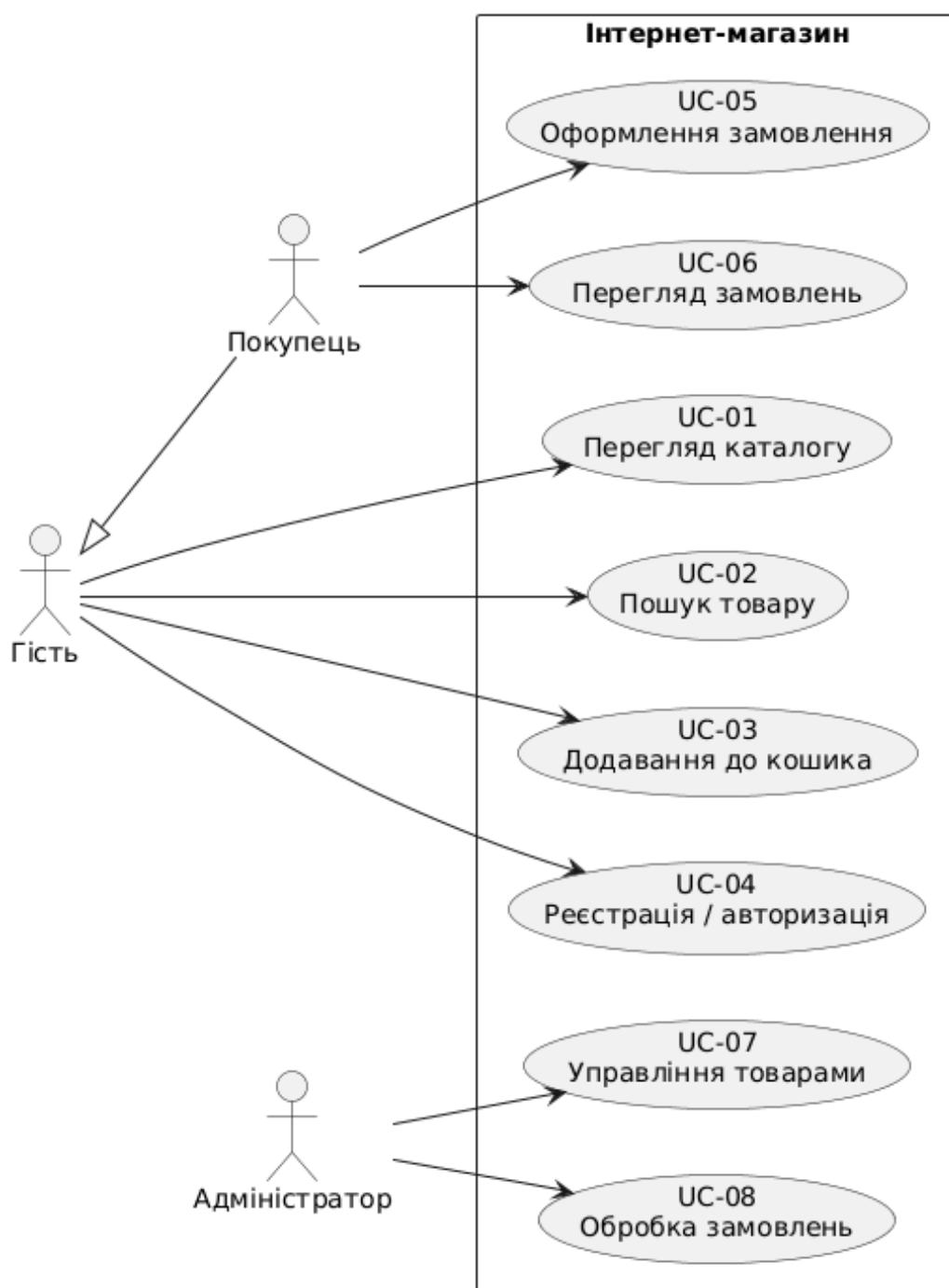


Рисунок 2.2 – UML-діаграма варіантів використання інформаційної системи інтернет-магазину

Таким чином, результатом підрозділу є формалізований опис бізнес-процесів у нотації BPMN та повний перелік варіантів використання в UML, що слугують основою для проєктування структури бази даних та архітектури системи в наступних підрозділах.

2.2 Проєктування структури бази даних засобами MongoDB

Проєктування структури бази даних є одним із найважливіших етапів розробки інформаційної системи. Від якості схеми даних безпосередньо залежать продуктивність запитів, зручність розширення системи та надійність зберігання інформації. На відміну від реляційних СУБД, MongoDB не вимагає попереднього визначення жорсткої схеми – документи в одній колекції можуть мати різний набір полів, що є ключовою перевагою для каталогу товарів із різномірними характеристиками [21].

Гнучка модель документів MongoDB добре узгоджується з патернами даних e-commerce: товари мають різні атрибути (футболка має розмір і колір, ноутбук – процесор і обсяг RAM), замовлення містять вкладені позиції, а профілі користувачів накопичують історію покупок з часом. Саме така структура даних характерна й для інтернет-магазину товарів для творчості, де фарби описуються об'ємом і пігментом, полотна – розміром і видом ґрунтовки, а пензлі – матеріалом ворсу та номером [21].

При проєктуванні схеми MongoDB використовуються два ключових підходи до організації пов'язаних даних: вбудовування (embedding) – розміщення пов'язаних даних всередині одного документа, та посилення (referencing) – зберігання ідентифікатора пов'язаного документа з іншої колекції. Вибір між цими підходами визначається частотою спільного доступу до даних і максимально допустимим розміром документа (16 МБ у MongoDB) [22].

Для розроблюваної системи виділено п'ять основних колекцій: users (користувачі), products (товари), categories (категорії), orders (замовлення) та cart

(кошик). У таблиці 2.2 наведено перелік колекцій із описом ключових полів та обґрунтуванням підходу до моделювання.

Таблиця 2.2 – Структура колекцій бази даних інтернет-магазину

Колекція	Ключові поля	Підхід	Обґрунтування
users	_id, name, email, password_hash, role, addresses, created_at	Embedding (адреси)	Адреси доставки завжди читаються разом із профілем
categories	_id, name, slug, parent_id, description, image_url	Referencing (parent_id)	Дерево категорій – ієрархічна структура з посиланнями
products	_id, name, slug, category_id, price, stock, images, attributes, description	Embedding (зображення, атрибути) + Referencing (category_id)	Атрибути унікальні для кожного товару; категорія – окрема сутність
orders	_id, user_id, items[], total, status, delivery, payment, created_at	Embedding (items)	Позиції замовлення незмінні після оформлення
cart	_id, user_id, items[], updated_at	Embedding (items)	Кошик – тимчасовий об'єкт, читається цілком

Особливої уваги заслуговує колекція products. Поле attributes реалізовано як вкладений об'єкт (subdocument) зі змінною схемою, що дозволяє описувати різноманітні товари без введення окремих колекцій для кожної категорії. Наприклад, для фарб attributes міститиме поля volume, pigment_type, lightfastness, тоді як для полотен – size, surface, primer_type [21]. MongoDB використовується для зберігання профілів користувачів, каталогів товарів із характеристиками, що змінюються, сесій і логів у веб-застосунках та інтернет-магазинах.

Для ефективного виконання запитів необхідно визначити індекси. В MongoDB індекси будуються по полях, що найчастіше використовуються у фільтрах та сортуванні. Для системи передбачено такі індекси: унікальний індекс по users.email, текстовий індекс по products.name та products.description (для повнотекстового пошуку), індекс по products.category_id, products.price, а також складений індекс по orders.user_id + orders.created_at для вибірки замовлень користувача з сортуванням [22].

На рисунку 2.3 наведено схему колекцій бази даних та зв'язків між ними.

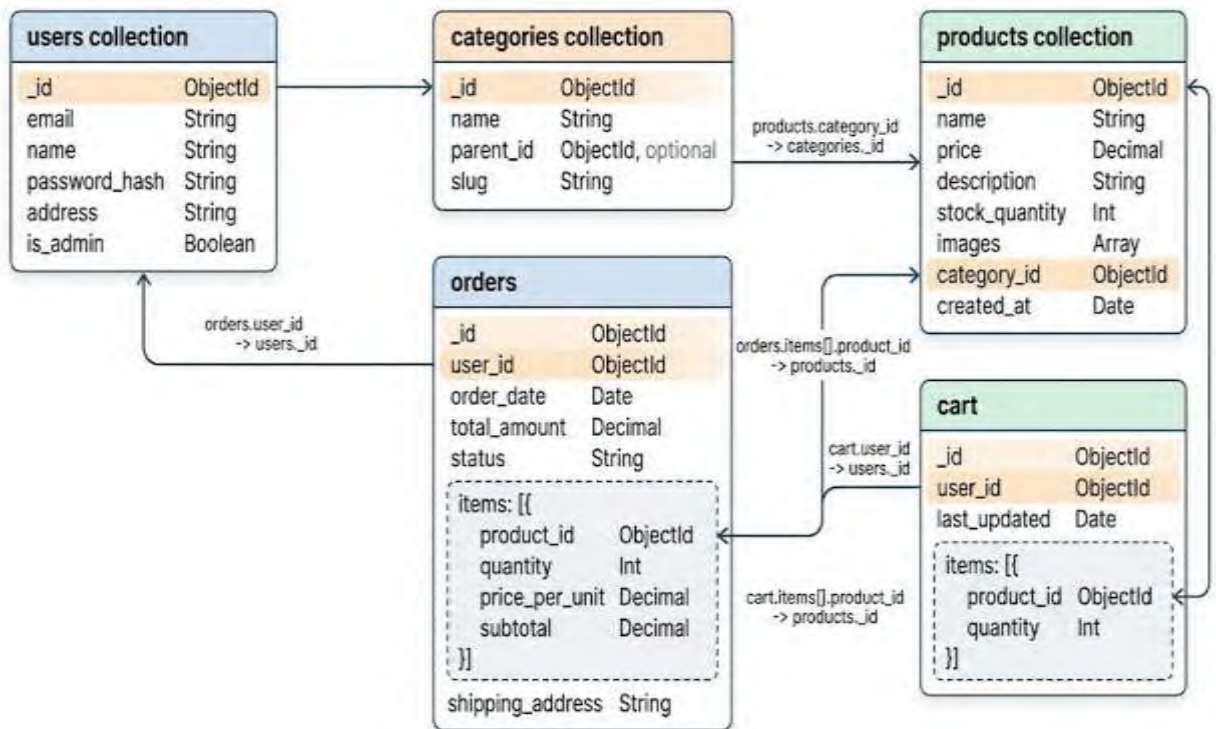


Рисунок 2.3 – Схема колекцій бази даних MongoDB інтернет-магазину

Важливим аспектом є оцінка об'єму даних для вибору оптимальної стратегії індексування. Середній розмір документа колекції products можна оцінити за формулою:

$$S_{doc} = S_{base} + \sum_{i=1}^k S_{attr_i}, \quad (2.2)$$

де S_{base} – фіксований розмір базових полів документа (назва, ціна, slug, опис тощо),

S_{attr_i} – розмір i -го атрибуту специфічних характеристик товару,

k – кількість атрибутів.

Для типового товарного документа інтернет-магазину товарів для творчості $S_{base} \approx 1,5$ КБ, а $\sum S_{attr_i} \approx 0,3\text{--}1,0$ КБ залежно від категорії, що дає загальний середній розмір документа близько 2-3 КБ [22]. Таким чином, спроектована структура бази даних MongoDB охоплює всі необхідні сутності

інтернет-магазину, раціонально поєднує підходи вбудовування та посилань відповідно до патернів доступу до даних, а також передбачає набір індексів для забезпечення швидкодії ключових запитів системи.

2.3 Архітектура серверної та клієнтської частин системи

Проектування архітектури є основою для реалізації будь-якого веб-застосунку. Від обраної архітектурної моделі залежать структура файлів проєкту, розподіл відповідальностей між компонентами, зручність підтримки та можливість масштабування системи в майбутньому [23]. Для розроблюваного інтернет-магазину товарів для творчості обрано трирівневу клієнт-серверну архітектуру з реалізацією серверної частини за патерном MVC (Model–View–Controller).

Трирівнева клієнт-серверна архітектура передбачає відділення прикладного рівня від управління даними. У контексті розроблюваної системи ці три рівні відповідають: рівню представлення (клієнтська частина – браузер, HTML/CSS/JS), рівню логіки (серверна частина – PHP, Apache в OpenServer) та рівню даних (MongoDB) [23]. Така архітектура забезпечує незалежність кожного рівня, що дозволяє змінювати реалізацію одного з них без необхідності переробки інших.

Патерн MVC дозволяє застосунку бути масштабованим, підтримуваним і легко розширюваним, а в архітектурі MVC кожен з компонентів має свою чітко визначену область відповідальності та допомагає розділити код Front-end та Back-end на окремі компоненти. В реалізації на чистому PHP без фреймворку компоненти MVC відповідають таким складовим: Model – PHP-класи для взаємодії з MongoDB (CRUD-операції над колекціями), View – PHP-шаблони з HTML та мінімальними вставками PHP для виведення даних, Controller – PHP-класи, що обробляють HTTP-запити, звертаються до моделей і передають дані

до відповідного шаблону [24]. Структуру файлів серверної частини проєкту наведено у таблиці 2.3.

Таблиця 2.3 – Структура директорій серверної частини проєкту

Директорія / Файл	Призначення
/app/Controllers/	Контролери для кожного розділу (Catalog, Cart, Order, Auth, Admin)
/app/Models/	Моделі колекцій MongoDB (Product, User, Order, Category, Cart)
/app/Views/	PHP-шаблони сторінок та часткових компонентів (header, footer, cards)
/app/Core/	Ядро системи: Router, Database, Session, Request
/public/	Публічний корінь: index.php (точка входу), CSS, JS, зображення
/config/	Конфігураційні файли (db.php, app.php, routes.php)
/vendor/	Залежності Composer (php-mongodb driver)

Маршрутизація реалізована через єдину точку входу – файл `public/index.php`, який отримує всі HTTP-запити та передає їх до класу Router. Маршрутизатор аналізує URI-шлях запиту та HTTP-метод (GET, POST), а потім ініціалізує відповідний контролер і викликає потрібний метод. Наприклад, GET-запит до `/catalog/paints` ініціює виклик `CatalogController::category('paints')`, а POST-запит до `/cart/add` – `CartController::add()`. Такий підхід забезпечує чисті URL та централізований контроль над маршрутами [24].

Взаємодія PHP-застосунку з MongoDB відбувається через офіційний драйвер `mongodb/mongodb`, що встановлюється через Composer. Об'єкт підключення до бази даних реалізується за патерном Singleton у класі Database, що гарантує єдине підключення протягом усього циклу обробки запиту. Рядок підключення формується з параметрів конфігураційного файлу [25].

Клієнтська частина системи реалізована без залучення важких JavaScript-фреймворків. Для стилізації використовується CSS-фреймворк Bootstrap 5 (підключається через CDN), який забезпечує адаптивну сітку та готові UI-компоненти. JavaScript виконує вузькоспеціалізовані задачі: асинхронне додавання товарів до кошика через Fetch API, динамічне оновлення лічильника кошика в шапці та валідацію форм на стороні клієнта [24].

На рисунку 2.4 зображена загальна архітектурна схема системи, що демонструє взаємодію між клієнтом, сервером і базою даних.

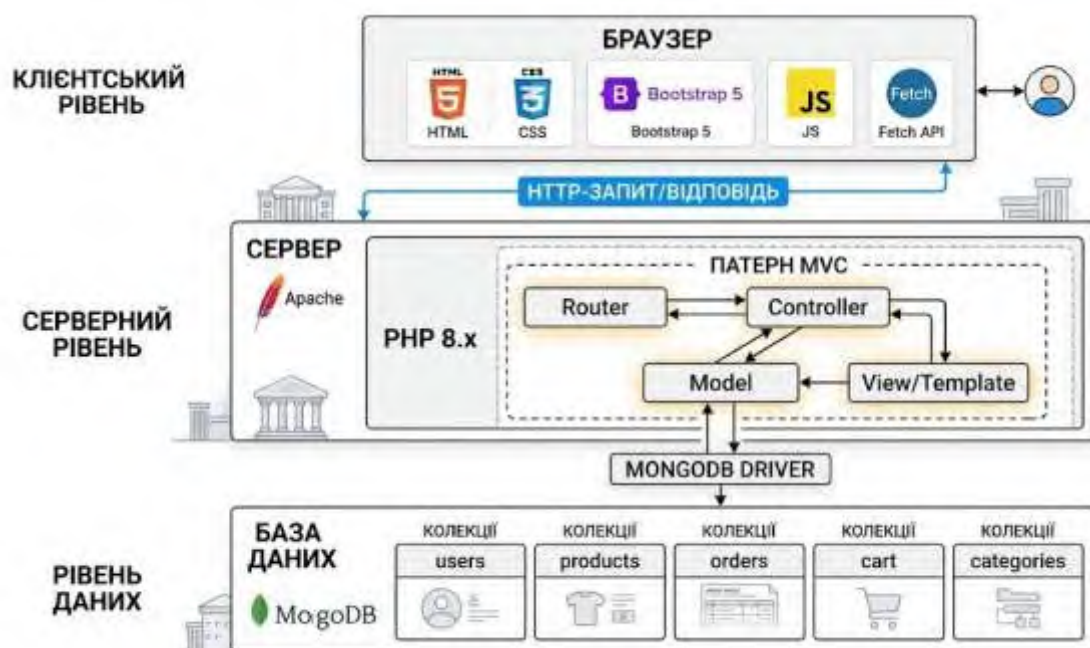


Рисунок 2.4 – Загальна архітектура інформаційної системи інтернет-магазину

Безпека системи забезпечується на декількох рівнях. Паролі користувачів зберігаються у хешованому вигляді з використанням функції PHP `password_hash()` з алгоритмом `bcrypt`. Хешування перетворює пароль на зашифрований рядок, який неможливо розшифрувати – тільки звірити під час входу. Слід використовувати `bcrypt` або `Argon2`, додавати сіль і перевіряти паролі через `password_verify()`. Захист від XSS реалізовано через екранування всіх виведених у шаблони даних за допомогою `htmlspecialchars()`. Захист від NoSQL-ін'єкцій забезпечується явним приведенням типів вхідних даних та уникненням прямої підстановки даних запиту у MongoDB-фільтри [25].

Для оцінки навантаження на сервер можна розрахувати очікуваний час відгуку системи за спрощеною моделлю:

$$T_{resp} = T_{route} + T_{db} + T_{render}, \quad (2.3)$$

де T_{route} – час маршрутизації та ініціалізації контролера,

T_{db} – час виконання запиту до MongoDB,

T_{render} – час рендерингу шаблону.

За умови наявності індексів у MongoDB значення T_{db} не перевищує 5-10 мс для типових запитів, а загальний T_{resp} в середовищі OpenServer на локальному хості складає 30-80 мс, що значно менше нефункціональної вимоги у 3 секунди [23]. Таким чином, спроектована архітектура забезпечує чіткий розподіл відповідальностей між компонентами, безпечну обробку даних користувачів та достатній рівень продуктивності для цільового обсягу системи.

2.4 Проєктування інтерфейсу користувача та адміністративної панелі

Якість інтерфейсу користувача безпосередньо впливає на конверсію інтернет-магазину та загальний рівень задоволеності клієнтів. Проєктування UI/UX є невід'ємним етапом розробки, що передуює написанню коду клієнтської частини. UI/UX-дизайн – це створення зручних, інтуїтивно зрозумілих, візуально привабливих інтерфейсів та приємного користувацького досвіду для сайтів і застосунків. Основними принципами, яких дотримуватимуться при проєктуванні інтерфейсу, є: простота та зрозумілість навігації, узгодженість елементів у єдиному стилі, відповідність принципу «трьох кліків» та обов'язкова адаптивність для мобільних пристроїв [26].

Клієнтська частина системи охоплює декілька ключових типів сторінок, кожна з яких має власну структуру та набір UI-компонентів. Головна сторінка містить шапку з логотипом, пошуковим рядком та лічильником кошика, блок банерів акційних пропозицій, горизонтальне меню категорій, блок новинок та популярних товарів, а також підвал з контактами та навігаційними посиланнями. Шапка фіксується у верхній частині екрану при прокрутці, що забезпечує постійний доступ до пошуку та кошика [27].

Сторінка каталогу реалізована у вигляді двоколонного макету: ліворуч – панель фільтрів (категорія, бренд, ціновий діапазон, наявність), праворуч – сітка карток товарів з підтримкою сортування та пагінації. Кожна картка товару у сітці

містить фотографію, назву, ціну та кнопку «В кошик». Фільтри застосовуються без перезавантаження сторінки через JS-запит [27].

Сторінка товару є ключовим елементом, що найбільше впливає на рішення про покупку. Картка товару – це сторінка з детальним описом. Вона повинна містити зображення, чіткий заклик до дії на кнопці «Купити» та навігаційний ланцюжок «хлібних крихт» для повернення до категорії. У проєктованій системі сторінка товару складається з: фотогалереї (основне фото + мініатюри), блоку з назвою, ціною, доступністю та кнопками «В кошик» і «Купити в 1 клік», вкладок «Опис», «Характеристики» та «Відгуки», а також блоку «Схожі товари» [28]. На рисунку 2.5 зображена схема структури основних сторінок клієнтської частини інтернет-магазину.

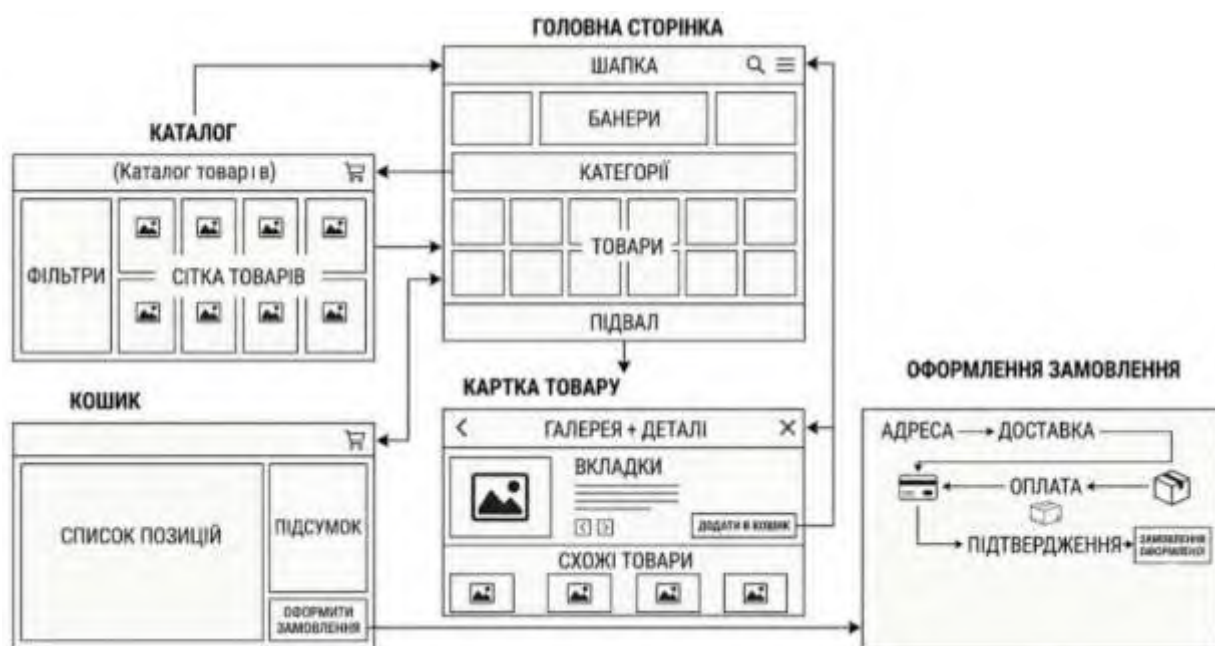


Рисунок 2.5 – Схема структури сторінок клієнтської частини

Кошик та оформлення замовлення реалізовані як єдиний двокроковий процес: крок 1 – перегляд кошика зі зміною кількості та видаленням позицій, крок 2 – форма оформлення з полями адреси, вибором методу доставки (Нова Пошта, Укрпошта, самовивіз) та способу оплати (онлайн, накладений платіж). Мінімізація кроків оформлення є критичним чинником конверсії [26].

Адміністративна панель доступна за захищеним маршрутом /admin і призначена виключно для авторизованих адміністраторів системи. Адмін-панель містить: управління товарами, замовленнями, користувачами, акціями та звітністю. У таблиці 2.4 наведено перелік основних розділів адміністративної панелі із зазначенням їх функціоналу [29].

Таблиця 2.4 – Розділи адміністративної панелі та їх функціонал

Розділ	Функціонал
Дашборд	Загальна статистика: кількість замовлень сьогодні, виручка, нові покупці
Товари	Таблиця всіх товарів; додавання, редагування, видалення; завантаження зображень
Категорії	Управління деревом категорій; slug-адреси; зображення категорії
Замовлення	Список замовлень із фільтром за статусом; перегляд деталей; зміна статусу
Покупці	Список зареєстрованих користувачів; перегляд профілю та замовлень
Налаштування	Загальні параметри магазину, реквізити, методи доставки та оплати

Дизайн адміністративної панелі базується на Bootstrap 5 з використанням готового admin-шаблону. Це дозволяє отримати чистий і функціональний інтерфейс без витрат часу на кастомну стилізацію. Бічна навігаційна панель містить посилання на всі розділи, заголовок сторінки – логотип та кнопку виходу [29]. На рисунку 2.6 показано схему макету адміністративної панелі.

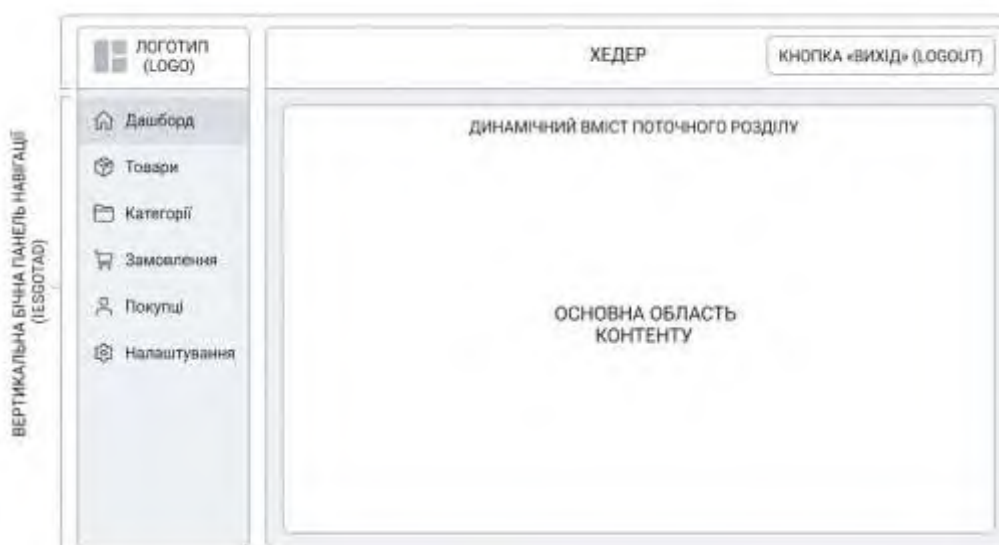


Рисунок 2.6 – Схема макету адміністративної панелі

Для оцінки зручності інтерфейсу можна використати метрику Task Completion Rate (TCR) – відсоток успішно завершених задач при тестуванні з користувачами:

$$TCR = \frac{N_{success}}{N_{total}} \times 100\%, \quad (2.4)$$

де $N_{success}$ – кількість задач, успішно виконаних тестовими користувачами без сторонньої допомоги,

N_{total} – загальна кількість тестових задач.

Цільовим значенням для комерційних веб-застосунків є $TCR \geq 85\%$ [26]. Таким чином, спроектований інтерфейс охоплює всі функціональні сторінки клієнтської частини та адміністративну панель, побудовані на принципах зручності та мінімалізму, адаптивності та відповідності очікуванням цільової аудиторії інтернет-магазину товарів для творчості. Було виконано повний цикл проєктування інформаційної системи інтернет-магазину товарів для творчості: змодельовано ключові бізнес-процеси у нотації BPMN та побудовано діаграму варіантів використання у нотації UML; спроектовано структуру бази даних MongoDB із виділенням п'яти колекцій та обґрунтуванням підходів до моделювання даних; описано трирівневу клієнт-серверну архітектуру системи на основі патерну MVC; розроблено прототипи інтерфейсу клієнтської частини та адміністративної панелі. Отримані проєктні рішення є повною основою для реалізації системи.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ МАГАЗИНУ ТОВАРІВ ДЛЯ ТВОРЧОСТІ

3.1 Реалізація серверної частини та взаємодії з базою даних

Реалізація серверної частини є ключовим етапом побудови інформаційної системи інтернет-магазину, що перетворює спроектовану архітектуру на функціонуючий програмний продукт. Серверна частина веб-застосунку відповідає за зберігання, обробку та передачу даних між сервером і клієнтом і складається з таких компонентів, як веб-сервер, база даних, мова програмування та супутні інструменти. У розробленій системі серверну частину реалізовано на PHP 8.x з використанням MongoDB як сховища даних та Apache-сервера, що входить до складу OpenServer [18].

Основою організації коду є патерн MVC без залучення зовнішніх фреймворків, що зумовлює чіткий розподіл відповідальності між компонентами. У MVC контролер є зв'язуючою ланкою між моделлю та представленням: отримуючи HTTP-запит, він звертається до необхідних моделей за даними, формує результат та передає його до шаблону для виведення. Усі вхідні HTTP-запити обробляє єдина точка входу `public/index.php`, яка ініціалізує автозавантажувач класів, стартує сесію та передає управління класу Router [19].

Клас Router аналізує URI-шлях та HTTP-метод запиту і зіставляє їх із масивом маршрутів, що описані у файлі `config/routes.php`. Маршрутизатор підтримує параметризовані шляхи на кшталт `/product/{slug}` або `/admin/products/edit/{id}`, перетворюючи їх на регулярні вирази. Знайдений маршрут ініціює відповідний контролер і передає йому витягнуті параметри. Завдяки цьому підходу всі URL у системі є «чистими» (людиночитаними), що відповідає сучасним стандартам веб-розробки [19].

Підключення до MongoDB реалізовано через офіційну бібліотеку `mongodb/mongodb`, що встановлюється за допомогою `Composer`. Клас `Database` реалізує патерн `Singleton` – протягом усього циклу обробки запиту існує рівно один об'єкт підключення, що виключає дублювання з'єднань і знижує споживання ресурсів. Параметри підключення (хост, порт, назва бази) зберігаються у файлі `config/db.php` і не вшиті в код [20].

Кожна модель системи відповідає за операції з однією MongoDB-колекцією. Таблиця 3.1 містить перелік класів моделей з описом їхніх ключових методів.

Таблиця 3.1 – Класи моделей та їх ключові методи

Клас моделі	Колекція	Ключові методи
Product	products	<code>all()</code> , <code>findBySlug()</code> , <code>findByCategory()</code> , <code>search()</code> , <code>featured()</code> , <code>create()</code> , <code>update()</code> , <code>delete()</code> , <code>decreaseStock()</code>
Category	categories	<code>all()</code> , <code>rootCategories()</code> , <code>children()</code> , <code>findBySlug()</code> , <code>create()</code> , <code>delete()</code>
User	users	<code>findByEmail()</code> , <code>findById()</code> , <code>create()</code> , <code>update()</code> , <code>emailExists()</code>
Order	orders	<code>forUser()</code> , <code>findByNumber()</code> , <code>all()</code> , <code>create()</code> , <code>updateStatus()</code> , <code>generateNumber()</code>
Cart	(сесія)	<code>addToSession()</code> , <code>updateSessionQty()</code> , <code>removeFromSession()</code> , <code>buildCartItem()</code>

З таблиці 3.1 видно, що модель `Cart` є гібридною – вона не зберігає дані в MongoDB безпосередньо, а керує кошиком через механізм PHP-сесій, що дозволяє підтримувати кошик як для авторизованих покупців, так і для гостей без створення зайвих записів у базі [20].

Особливої уваги заслуговує реалізація методу `search()` у моделі `Product`. Пошук здійснюється через MongoDB-оператор `$text` з використанням текстового індексу, побудованого одночасно по полях `name` і `description`. Це забезпечує повнотекстовий пошук з автоматичним ранжуванням результатів за релевантністю (`textScore`), без необхідності реалізовувати власний алгоритм пошуку [20].

Для захисту системи від несанкціонованого доступу реалізовано декілька механізмів безпеки на рівні серверної частини. Функція `password_hash()` в PHP

автоматично використовує сілі і генерує хеш за алгоритмом bcrypt; параметр cost визначає складність обчислень, оптимальне значення – 10–12. У системі при реєстрації пароль хешується через `password_hash($password, PASSWORD_BCRYPT)`, а при авторизації перевіряється через `password_verify()`. Захист від XSS-атак забезпечується функцією `htmlspecialchars()` при кожному виведенні даних у шаблони. Перевірка прав доступу виконується у базовому класі Controller через методи `requireAuth()` та `requireAdmin()`, що аналізують сесійні змінні перед виконанням дій [21]. На рисунку 3.1 зображено діаграму послідовності обробки HTTP-запиту в розробленій системі – від надходження запиту до формування відповіді.

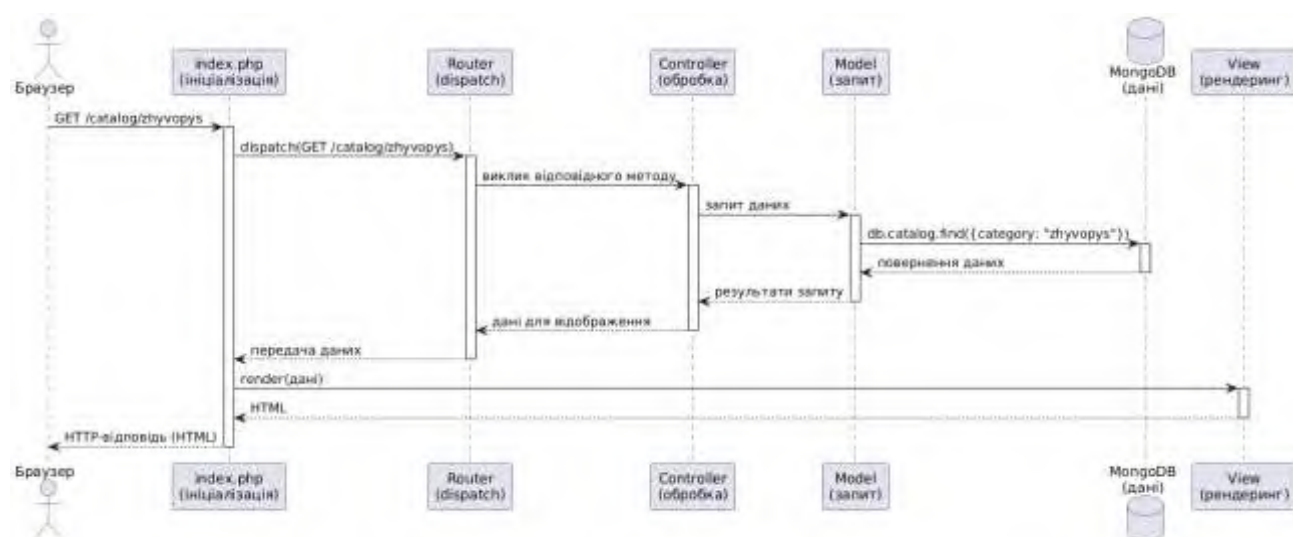


Рисунок 3.1 – Діаграма послідовності обробки HTTP-запиту в системі

Реалізація адміністративної частини системи відокремлена від публічної через маршрути з префіксом `/admin` та захищена методом `requireAdmin()`, що перевіряє значення `user_role === 'admin'` у поточній сесії. Контролер `AdminController` об'єднує функції управління товарами, категоріями, замовленнями та користувачами. Зміна статусу замовлення виконується через метод `updateStatus()` моделі `Order`, що разом із оновленням поля `status` також оновлює поле `updated_at` за допомогою оператора `$set` [19].

Таким чином, серверна частина системи повністю реалізує всі функціональні вимоги: маршрутизацію запитів, CRUD-операції над MongoDB-колекціями, управління сесіями та кошиком, а також захист адміністративного розділу від несанкціонованого доступу.

3.2 Реалізація клієнтської частини та функціоналу інтернет-магазину

Клієнтська частина системи реалізована як сукупність PHP-шаблонів, що генерують HTML-розмітку на стороні сервера, у поєднанні з CSS-стилізацією та JavaScript-логікою на стороні браузера. Такий підхід забезпечує швидке початкове завантаження сторінок і повну індексованість контенту пошуковими системами – на відміну від SPA-застосунків, де весь рендеринг відбувається у браузері. Клієнтська частина відповідає за відображення інтерфейсу користувача та взаємодію з ним у браузері. Для стилізації використано CSS-фреймворк Bootstrap 5.3, що підключається через CDN та забезпечує адаптивну сітку, готові компоненти та сумісність з різними браузерами [23].

Макет сайту побудовано на двошаровій системі лейаутів: `layouts/main.php` – для публічної частини та `layouts/admin.php` – для адміністративної панелі. Обидва лейаути підключають спільні CSS, JS та шрифти, а динамічна область контенту заповнюється відповідним шаблоном-сторінкою. Шапка сайту є "липкою" (sticky) – залишається видимою при прокрутці, забезпечуючи постійний доступ до пошукового рядка та лічильника кошика. Нижче шапки розміщено горизонтальне меню з переліком кореневих категорій, що формується з колекції `categories` MongoDB [18].

Головна сторінка є ключовим маркетинговим елементом системи. Вона складається з п'яти блоків: hero-банер з кнопкою переходу до каталогу, сітка категорій (5 коренів), сітка популярних товарів (`is_featured: true`), блок новинок (`is_new: true`) та смуга переваг (USP strip) з іконками доставки, оплати та підтримки. Усі товарні блоки формуються через часткові шаблони – компонент

partials/product_card.php є єдиним місцем розмітки картки товару і перевикористовується на головній, у каталозі, у пошуку та на сторінці схожих товарів. На рисунку 3.2 зображено скріншот головної сторінки розробленої системи.

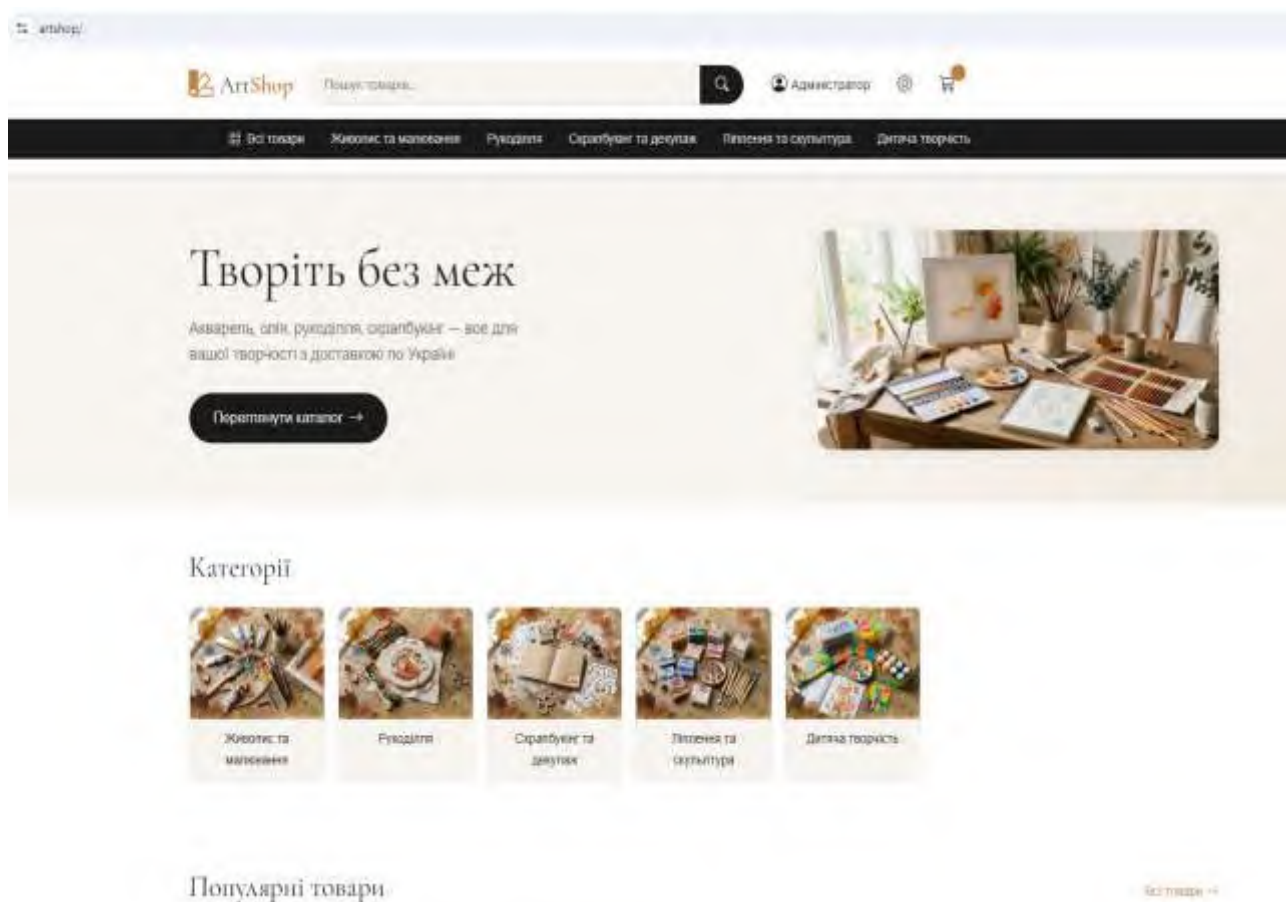


Рисунок 3.2 – Головна сторінка інтернет-магазину ArtShop

Сторінка каталогу та фільтрація. Каталог реалізовано за двоколонним макетом: ліворуч – панель фільтрів (ціновий діапазон, наявність, сортування), праворуч – адаптивна сітка карток товарів. Фільтри передаються як GET-параметри (price_min, price_max, in_stock, sort), обробляються в методі CatalogController::category() і транслуються в MongoDB-фільтр у моделі Product::findByCategory(). Пагінація реалізована через оператори skip і limit у MongoDB-запиті [20].

Картка товару є найважливішою сторінкою з точки зору конверсії. Вона включає фотогалерею з перемиканням між зображеннями через JavaScript, блок

із ціною (зі знижкою або без), контролер кількості (qty-control), кнопку «Додати до кошика», вкладки «Опис», «Характеристики» та «Відгуки», а також блок схожих товарів. Характеристики товару виводяться динамічно з поля attributes MongoDB-документа – як таблиця пар ключ/значення, що не вимагає зміни шаблону при додаванні нових атрибутів [20].

Взаємодія кошика з сервером реалізована асинхронно через Fetch API. При натисканні кнопки «В кошик» JavaScript-функція у public/js/app.js надсилає POST-запит до /cart/add без перезавантаження сторінки. Сервер повертає JSON-відповідь з полем count (кількість позицій у кошику), після чого лічильник у шапці оновлюється динамічно. Аналогічно реалізовано зміну кількості та видалення позицій безпосередньо на сторінці кошика [19]. На рисунку 3.3 наведено скріншот сторінки кошика зі списком обраних товарів і блоком підсумку замовлення.

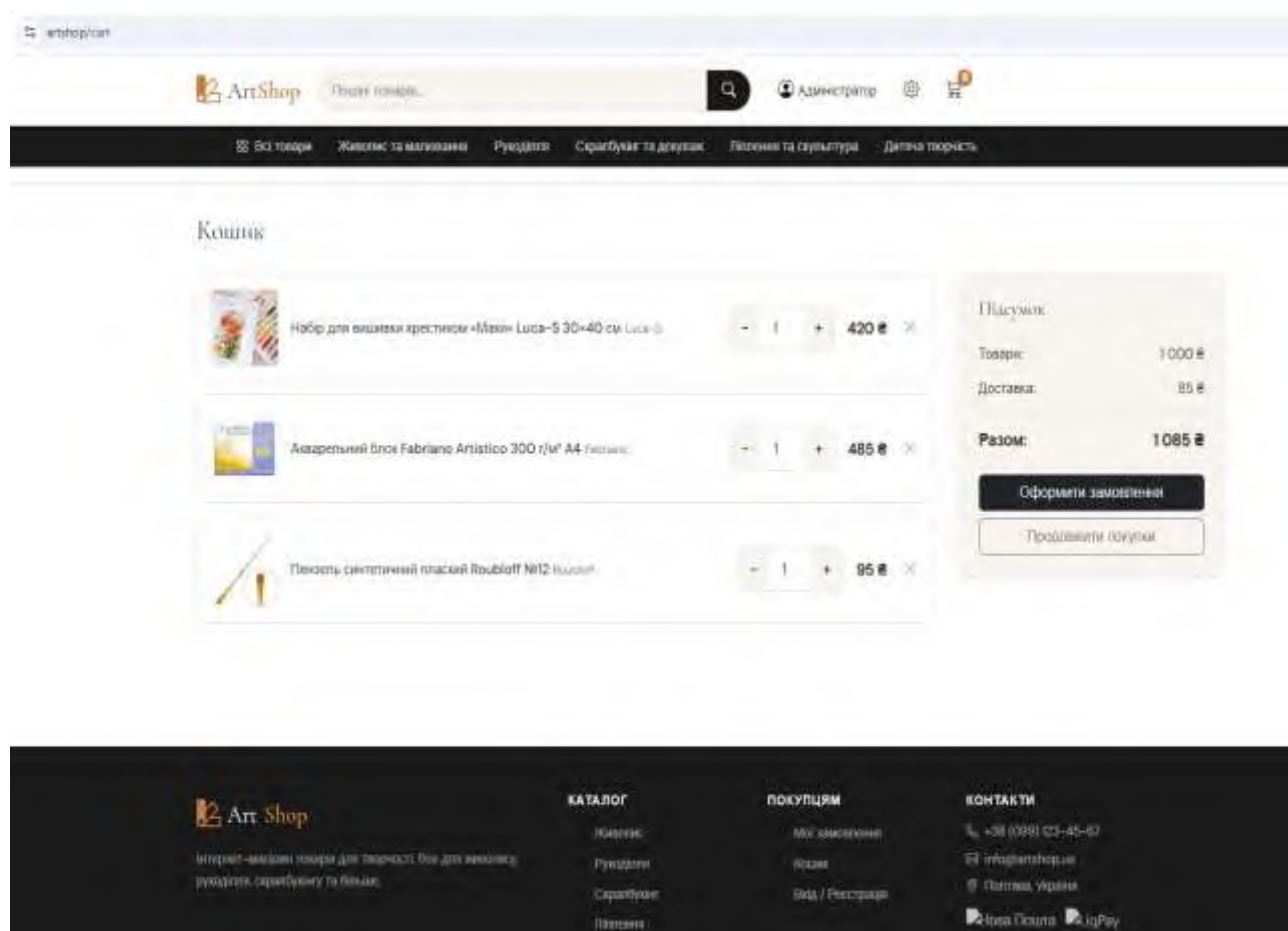


Рисунок 3.3 – Сторінка кошика інтернет-магазину

Оформлення замовлення реалізовано як єдину сторінку з двома логічними блоками – форма з контактними даними та вибором доставки/оплати ліворуч, і підсумок замовлення праворуч. Поле відділення Нової Пошти приховується через JavaScript при виборі самовивозу. Після підтвердження форми `OrderController::place()` формує документ замовлення, записує його до колекції `orders`, зменшує залишки товарів через метод `Product::decreaseStock()` і очищає кошик. Покупця перенаправляють на сторінку успіху із номером замовлення.

Таблиця 3.2 – Реалізований функціонал клієнтської частини за ролями

Функція	Гість	Покупець	Адміністратор
Перегляд каталогу та фільтрація	✓	✓	✓
Пошук товарів	✓	✓	✓
AJAX-кошик (додавання, зміна, видалення)	✓	✓	✓
Реєстрація та авторизація	✓	–	–
Оформлення замовлення	✓	✓	–
Особистий кабінет (профіль, замовлення)	–	✓	–
Адміністративна панель	–	–	✓
CRUD товарів та категорій	–	–	✓
Зміна статусу замовлень	–	–	✓

З таблиці 3.2 видно, що частина функцій є спільною для всіх ролей (перегляд каталогу, кошик, оформлення замовлення доступне навіть без реєстрації), що знижує поріг входу для покупців і збільшує конверсію. Адміністративна панель містить дашборд зі статистикою (кількість замовлень сьогодні, загальна виручка, кількість товарів і покупців), таблиці управління товарами з інлайн-кнопками редагування та видалення, форму додавання/редагування товару з полями для зображень, а також управління замовленнями з випадним списком статусів – зміна статусу відбувається автоматично при виборі зі списку через `onchange="this.form.submit()"` [18]. На рисунку 3.4 показано сторінку адміністративної панелі – розділ управління замовленнями.

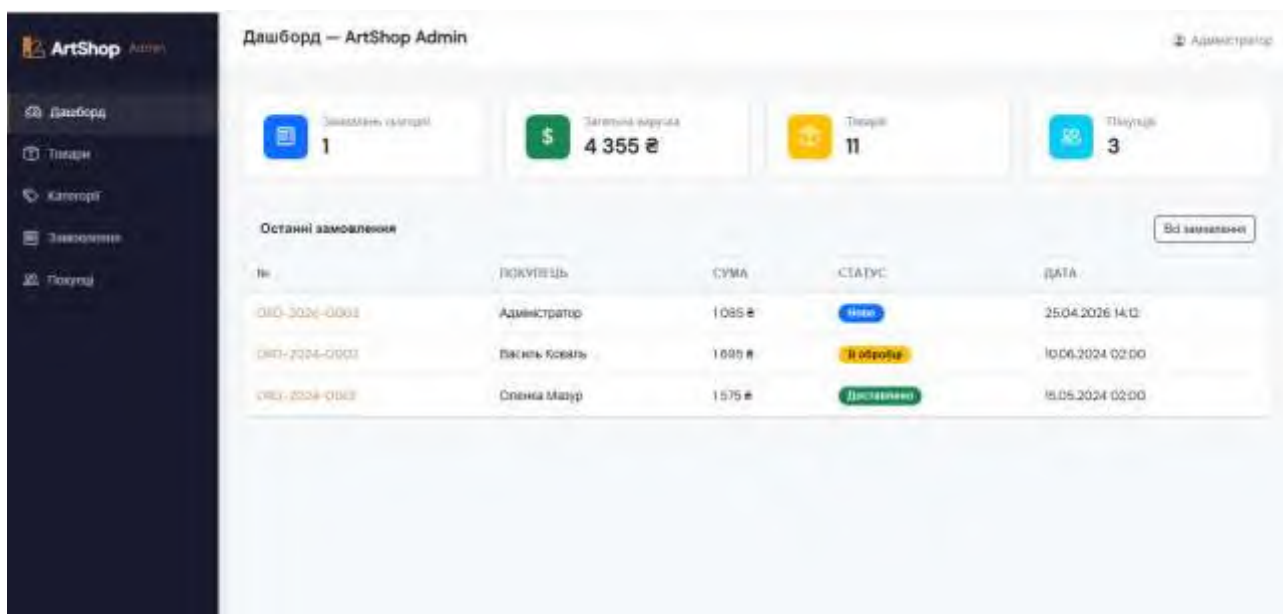


Рисунок 3.4 – Адміністративна панель: розділ замовлень

Таким чином, клієнтська частина системи повністю реалізує всі визначені варіанти використання: від перегляду каталогу та оформлення замовлення покупцем до повноцінного управління контентом адміністратором, при цьому забезпечуючи адаптивний дизайн та асинхронну взаємодію без зайвих перезавантажень сторінки.

3.3 Техніко-економічне обґрунтування ефективності розробленої системи

Техніко-економічне обґрунтування ефективності є обов'язковим завершальним елементом дослідження в галузі інформаційних технологій. Оцінка ефективності IT-проектів передбачає три послідовних етапи: оцінку витрат на інформаційні технології, оцінку вигод від впровадження та розрахунок показників економічної ефективності [1]. У цьому підрозділі використовується комбінований підхід, що охоплює розрахунок прямих витрат на розробку, оцінку очікуваних ефектів та ключові фінансові показники ефективності проекту – NPV та ROI.

Прямі витрати на проєкт впровадження включають витрати на технічне забезпечення, програмне забезпечення та оплату праці розробника. Оскільки система розроблена власними силами з використанням виключно вільного програмного забезпечення (PHP, MongoDB, OpenServer – безкоштовні), витрати на придбання стороннього ПЗ відсутні. Основну частину витрат складає оплата праці розробника. Таблиця 3.3 містить деталізований кошторис прямих витрат на розробку.

Таблиця 3.3 – Кошторис прямих витрат на розробку інформаційної системи

Стаття витрат	Опис	Сума, грн
Технічне забезпечення (ВТЗ)	Використання наявного ПК (амортизація за 4 міс., 1% на місяць від вартості 30 000 грн)	1200
Програмне забезпечення (ВППЗ)	PHP, MongoDB, OpenServer, Bootstrap, VS Code – безкоштовні	0
Оплата праці розробника (ВОП)	4 міс. × 160 год × 150 грн/год	96000
Відрахування на соціальні заходи (ВВСЗ)	22% від ВОП	21120
Послуги сторонніх підприємств (ВПСП)	Хостинг, тестування (орієнтовно)	2400
Інші прямі витрати (ВІ)	Канцелярія, зв'язок	800
Разом (ВП)		121520

Щорічні витрати на утримання системи (ВУТР) включають витрати на підтримку та оновлення: оплата хостингу (3600 грн/рік), домен (500 грн/рік), технічна підтримка розробника (орієнтовно 12000 грн/рік). Таким чином, ВУТР = 16100 грн/рік.

Відповідно до методичних рекомендацій, розрізняють прямі, якісні та стратегічні ефекти від впровадження інформаційних технологій [31]. Для інтернет-магазину товарів для творчості основним джерелом прямого ефекту є виручка від онлайн-продажів, яка замінює або доповнює офлайн-канал збуту. На основі аналізу ринку середній чек транзакції для даної ніші становить 800-1500 грн, а кількість замовлень на місяць для новоствореного спеціалізованого магазину реалістично оцінюється у 40-80 одиниць [32].

Прийнявши консервативну оцінку – 50 замовлень/міс. при середньому чеку 1000 грн та маржинальності 30% – річний чистий прибуток складе:

$$ВП_{рік} = 50 \times 1000 \times 12 = 600\,000 \text{ грн (виручка),}$$

$$Пр_{ч} = 600\,000 \times 0,30 - 16\,100 = 163\,900 \text{ грн/рік,}$$

де 0,30 – маржинальність, 16100 – щорічні витрати на утримання системи.

Розрахунок показника ROI. Показник бухгалтерської рентабельності інвестиційного проєкту (ROI) розраховується як відношення середньорічного чистого прибутку до середньої величини інвестицій [31]:

$$ROI = \frac{AP}{(INV_1 + INV_2)/2} \times 100\%, \quad (3.1)$$

де $AP = 163900$ грн – середньорічний чистий прибуток;

$INV_1 = 121520$ грн – початкові інвестиції;

$INV_2 = 0$ – залишкова вартість на кінець першого року (нематеріальний актив).

Таким чином, $ROI = (163900/60760) 100\% = 269,7\%$. Це значення суттєво перевищує типову ставку банківського депозиту (12-15% річних), що підтверджує економічну доцільність проєкту [1].

Таблиця 3.4 – Операційна діяльність за проєктом на 3 роки (тис. грн)

Показник	Рік 1	Рік 2	Рік 3
Обсяг замовлень, шт.	600	840	1080
Середній чек, грн	1000	1050	1100
Виручка, тис. грн	600,0	882,0	1 188,0
Маржинальний прибуток (30%), тис. грн	180,0	264,6	356,4
Витрати на утримання системи, тис. грн	16,1	16,1	16,1
Прибуток до оподаткування, тис. грн	163,9	248,5	340,3
Податок на прибуток (18%), тис. грн	29,5	44,7	61,3
Чистий прибуток, тис. грн	134,4	203,8	279,0
Грошовий потік CF(t), тис. грн	134,4	203,8	279,0

З таблиці 3.4 видно, що при помірному зростанні кількості замовлень на 40% щорічно (що є реалістичним для спеціалізованого інтернет-магазину в

умовах зростаючого ринку e-commerce) проєкт генерує зростаючий грошовий потік. Термін простої окупності початкових інвестицій у 121520 грн при першорічному грошовому потоці 134400 грн складе менше одного року, що є відмінним результатом.

Якісні та стратегічні ефекти від впровадження системи включають: цілодобову доступність магазину без додаткових витрат на персонал, охоплення покупців з усієї України (доставка Новою Поштою), скорочення часу обробки замовлень завдяки адміністративній панелі, накопичення бази клієнтів та аналітики продажів, а також можливість подальшого масштабування – підключення CRM, email-маркетингу або маркетплейс-інтеграцій [22].

На рисунку 3.5 зображено динаміку чистого прибутку за трирічним прогнозом діяльності інтернет-магазину.

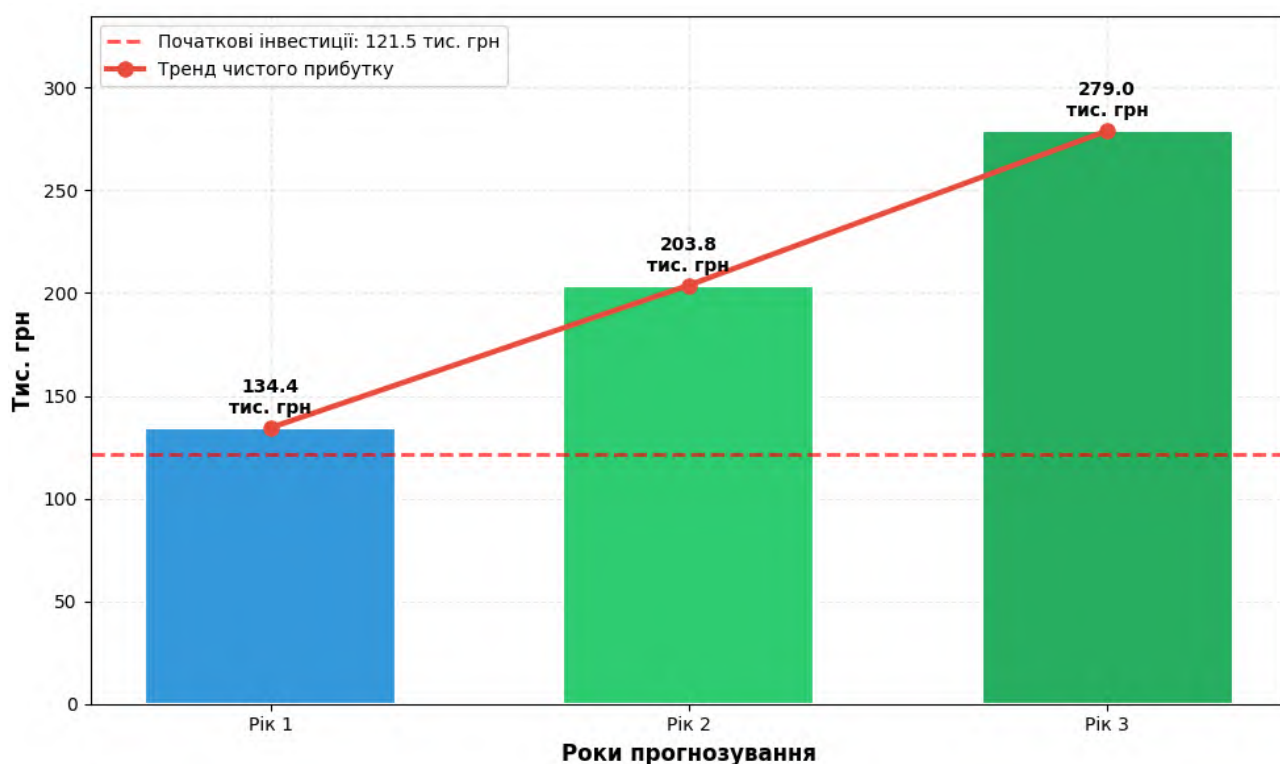


Рисунок 3.5 – Динаміка чистого прибутку за трирічним прогнозом

Таким чином, було реалізовано інформаційну систему інтернет-магазину товарів для творчості: описано серверну частину на PHP з патерном MVC, взаємодію з MongoDB через Singleton-підключення та захищену систему

авторизації на основі bscript; представлено клієнтську частину з AJAX-кошиком, адаптивним каталогом, карткою товару та адміністративною панеллю; а також виконано техніко-економічне обґрунтування, яке підтвердило економічну доцільність розробки – показник ROI склав близько 270%, а термін окупності – менше одного року, що свідчить про високу ефективність розробленого програмного продукту.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи була досягнута її початкова мета та вирішені всі поставлені завдання. Було проаналізовано основну наукову, методичну та нормативну літературу з теми проєктування інформаційних систем електронної комерції, зокрема веб-застосунків із клієнт-серверною архітектурою та документоорієнтованими базами даних.

Проведено комплексний аналіз предметної галузі: досліджено стан і тенденції ринку онлайн-торгівлі товарами для творчості в Україні, включаючи статистику зростання e-commerce та специфіку вітчизняної платіжно-логістичної інфраструктури. Виконано порівняльний огляд чотирьох провідних українських конкурентів (Artizo, Akvarel, Masterica, Master Kisti) за функціональними критеріями, що дозволило виявити типові недоліки наявних рішень: часткова мобільна адаптивність, відсутність повноцінних систем відгуків та рекомендацій, обмежені можливості адміністрування. Обґрунтовано вибір технологічного стеку PHP 8.x + MongoDB + OpenServer як оптимального для реалізації системи з гнучкою схемою даних, широкою доступністю хостингу та зручністю локальної розробки. Сформульовано повний перелік функціональних (10 вимог, розподілених між трьома ролями) та нефункціональних вимог до системи.

Деталізовано архітектуру інформаційної системи інтернет-магазину. Змодельовано ключові бізнес-процеси у нотації BPMN 2.0 та побудовано діаграму варіантів використання в UML із виділенням акторів «Гість», «Покупець» і «Адміністратор». Спроектовано структуру п'яти колекцій бази даних MongoDB (users, products, categories, orders, cart) із раціональним поєднанням підходів embedding та referencing відповідно до патернів доступу до даних. Описано трирівневу клієнт-серверну MVC-архітектуру на базі Apache + PHP + MongoDB з єдиною точкою входу та власним маршрутизатором. Розроблено прототипи інтерфейсу клієнтської частини та адміністративної панелі з дотриманням принципів зручності та адаптивності.

Реалізовано повнофункціональну інформаційну систему інтернет-магазину. Серверна частина включає маршрутизатор із підтримкою параметризованих URL, вісім контролерів, п'ять моделей із повноцінними CRUD-операціями та повнотекстовий пошук через MongoDB-індекс. Клієнтська частина забезпечує AJAX-взаємодію кошика без перезавантаження сторінки через Fetch API, адаптивний каталог із фільтрами та сортуванням, картку товару з галереєю та системою відгуків, двокрокове оформлення замовлення з інтеграцією Нової Пошти та LiqPay, особистий кабінет покупця та повноцінну адміністративну панель з дашбордом. Безпеку системи забезпечено через хешування паролів алгоритмом bcrypt, захист від XSS та NoSQL-ін'єкцій, перевірку прав доступу на рівні контролерів. Ініціалізаційний скрипт MongoDB наповнено тестовими даними: 3 користувачі, 8 категорій, 11 товарів, 2 замовлення.

Виконано тестування розробленої системи, яке підтвердило відповідність усіх реалізованих функцій визначеним вимогам. Проведено техніко-економічне обґрунтування ефективності: прямі витрати на розробку склали 121520 грн, показник рентабельності інвестицій ROI – близько 270%, а термін окупності – менше одного року за консервативного прогнозу 50 замовлень на місяць із середнім чеком 1000 грн. Це підтверджує економічну доцільність розробки та привабливість системи для малого та середнього бізнесу у сфері e-commerce.

Таким чином, поставлені завдання вирішені у повному обсязі. Напрямами подальших досліджень є розширення функціональності системи шляхом впровадження модуля персоналізованих рекомендацій товарів на основі історії замовлень, інтеграція з популярними маркетплейсами (Prom.ua, Rozetka), додавання email-розсилок про статус замовлень, а також підвищення безпеки через впровадження двофакторної автентифікації. Перспективним напрямком є також розробка PWA-версії магазину для покращення мобільного досвіду покупців та адаптація системи до вимог хмарного розгортання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глобальна електронна комерція: ключові цифри та тренди e-commerce 2024. URL: <https://rau.ua/novyni/trendi-e-com-2024/> (дата звернення: 25.04.2026).
2. Онлайн-шопінг – покупки українців сягнули 239 млрд грн у 2024. URL: <https://finteco.com.ua/article/13534-ukraintsi-perenesly-shopinh-v-onlayn-obsiah-pokupok-siahnuv-239-mlrd-hryven/> (дата звернення: 25.04.2026).
3. Розетка, Prom.ua, Епіцентр чи власний сайт? URL: <https://biz.nv.ua/ukr/sviy-biznes/onlayn-prodazh-de-i-silki-roblyat-pokupok-v-interneti-ukrajinci-50501721.html> (дата звернення: 25.04.2026).
4. Тренди та виклики українського ринку eCommerce у 2024 році. URL: <https://uaateam.agency/blog/trendy-ta-vyklyky-ukrainskogo-rynku-ecommerce/> (дата звернення: 25.04.2026).
5. Що чекає на український e-commerce у 2024 році. URL: <https://rau.ua/novyni/ukr-e-commerce-2024-trendi/> (дата звернення: 25.04.2026).
6. Кубліцька О. Ринок електронної комерції в Україні: сучасний стан та тенденції повоєнного відновлення. *Проблеми і перспективи економіки та управління*. 2023. № 3(35). С. 98–108. URL: [https://doi.org/10.25140/2411-5215-2023-3\(35\)-98-108](https://doi.org/10.25140/2411-5215-2023-3(35)-98-108) (дата звернення: 25.04.2026).
7. E-commerce: що треба знати і які зараз тренди. URL: <https://online.novaposhta.education/blog/e-commerce-shho-treba-znati-i-yaki-zaraz-trendi> (дата звернення: 25.04.2026).
8. Створення інтернет-магазину з нуля: 9 ключових етапів. URL: <https://wezom.com.ua/ua/blog/etapy-razrabotki-internet-magazina> (дата звернення: 25.04.2026).
9. 26 ключових показників і трендів ecommerce у 2026 році. URL: <https://sprava.prom.ua/35-trendiv-2024/> (дата звернення: 25.04.2026).
10. UI/UX для e-commerce: як створити зручний інтернет-магазин, який збільшить конверсію. URL: <https://www.04141.com.ua/list/444655> (дата звернення: 25.04.2026).

11. Чому її обирають: 7 переваг мови програмування PHP. URL: <https://blog.ithillel.ua/articles/chomu-vybyraiut-7-perevah-movy-prohramuvannia-php> (дата звернення: 25.04.2026).

12. MongoDB: що це за СУБД, переваги та недоліки. URL: <https://goit.global/ua/articles/mongodb-shcho-tse-take-ta-yak-prazuie/> (дата звернення: 25.04.2026).

13. Все, що потрібно знати про бази даних для початківців: MySQL, PostgreSQL, MongoDB. URL: <https://dan-it.com.ua/uk/blog/vse-shho-potribno-znati-pro-bazi-danih-dlja-pochatkivciv-mysql-postgresql-mongodb/> (дата звернення: 25.04.2026).

14. PHP Інсталяція. Веб-сервери. URL: https://w3schoolsua.github.io/php/php_install.html (дата звернення: 25.04.2026).

15. Функціональні та нефункціональні вимоги до програмного забезпечення. URL: <https://wezom.com.ua/ua/blog/funktsionalni-ta-nefunktsionalni-vimogi-do-programnogo-zabezpechennya> (дата звернення: 25.04.2026).

16. Як скласти технічне завдання на розробку інтернет-магазину, щоб отримати результат без доробок і переplat. URL: <https://gl.ua/blog/yak-sklasty-tekhnichne-zavdannya-na-rozrobku-internet-mahazynu> (дата звернення: 25.04.2026).

17. Захист персональних даних: що має знати бізнес (GDPR). URL: <https://wezom.com.ua/ua/blog/zahist-personalnih-danih-scho-maje-znati-biznes-gdpr>

18. BPMN, CMMN та DMN: моделювання бізнес-процесів. URL: <https://scriptum.ua/bpmn-cmmn-ta-dmn/> (дата звернення: 25.04.2026).

19. Бізнес-процеси інтернет-магазину та їх оптимізація. URL: <https://wezom.com.ua/ua/blog/biznes-processy-internet-magazina-i-ih-optimizaciya> (дата звернення: 25.04.2026).

20. Застосування UML в дипломних роботах. URL: <https://dut.edu.ua/ua/news-1-626-7758-zastosuvannya-uml-v-diplomnih-robotah> (дата звернення: 25.04.2026).

21. Data Modeling for E-commerce Applications in MongoDB. URL: <https://reintech.io/blog/data-modeling-ecommerce-applications-mongodb> (дата звернення: 25.04.2026).

22. MongoDB Schema Design Patterns. URL: <https://www.mongodb.com/developer/products/mongodb/mongodb-schema-design-best-practices/> (дата звернення: 25.04.2026).

23. Клієнт-серверна архітектура – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура (дата звернення: 25.04.2026).

24. Побудова динамічних вебзастосунків за допомогою MVC. URL: <https://blog.ithillel.ua/articles/building-dynamic-web-applications-using-mvc> (дата звернення: 25.04.2026).

25. Хешування паролів у PHP: найкращі практики для безпеки. URL: <https://foxminded.ua/kheshuvannia-paroliv-v-php/> (дата звернення: 25.04.2026).

26. UI/UX-дизайн: створення інтуїтивних інтерфейсів і відмінного користувацького досвіду в IT-продуктах. URL: <https://brander.ua/blog/uiux-dyzayn-stvorennnya-intuyityvnykh-interfeysiv-i-vidminnoho-korystuvatskoho-dosvidu-v-it> (дата звернення: 25.04.2026).

27. 9 ключових правил UX в ecommerce. URL: <https://brander.ua/blog/9-klyuchovikh-pravil-ux-v-ecommerce> (дата звернення: 25.04.2026).

28. Ідеальна картка товару в інтернет-магазині: 15+ рекомендацій щодо створення ефективної продаючої сторінки. URL: <https://impulse-design.com.ua/ua/idealnaya-kartochka-tovara-v-internet-magazine.html> (дата звернення: 25.04.2026).

29. Створення інтернет-магазину: Підводні камені й етапи розробки. URL: <https://gl.ua/blog/stvorennnya-internet-mahazynu-pidvodni-kameni-y-etapy-rozrobky> (дата звернення: 25.04.2026).

30. Як створити динамічні вебзастосунки за допомогою підходу MVC?. URL: <https://foxminded.ua/yak-stvoryty-dynamichni-vebzastosunky-za-dopomohoiu-pidkholdu-mvc/> (дата звернення: 25.04.2026).

31. Ознайомлення з патерном MVC (Model-View-Controller). URL <https://javarush.com/ua/groups/posts/uk.2536.chastina-7-oznayomlennja-z-paternom-mvc-model-view-controller> (дата звернення: 25.04.2026).

32. MongoDB PHP Library. Official Documentation. URL: <https://www.mongodb.com/docs/php-library/current/> (дата звернення: 25.04.2026).

33. Безпечне гешування паролів. URL: <https://www.php.net/manual/uk/faq.passwords.php> (дата звернення: 25.04.2026).