

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

БЛИЗНЮК ВЛАДИСЛАВА ОЛЕКСАНДРІВНА

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій
канд. техн. наук, доцент
_____ О. В. Зелінська
« ____ » _____ 20__ р.

**РОЗРОБКА ВЕБПЛАТФОРМИ ПРОДАЖУ ТОВАРІВ АНІМАЦІЙНОЇ
ТЕМАТИКИ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:
Т. В. Січко, доцент кафедри
інформаційних технологій,
к. т. н., доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною
шкалою)
Голова ЕК: _____

Вінниця - 2024

АНОТАЦІЯ

Близнюк В. О. Розробка вебплатформи продажу товарів анімаційної тематики. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній (бакалаврській) роботі досліджено та проаналізовано поняття вебплатформи, основні її принципи та методи створення. З використанням технологій PHP, HTML, CSS та фреймворків Laravel, Tailwind було розроблено веб платформу, мета якої – продаж товарів анімаційної тематики.

Ключові слова: веб платформа, розробка, веб-дизайн, база даних, PHP, HTML.

61 ст. 39 рис., 2 табл., 40 джерел.

ABSTRACT

Blyznyuk V. O. Development of a Web Platform for the Sale of Animation-Themed products. Specialty 122 "Computer Science", educational program "Computer Science". Vasyl Stus Donetsk National University, Vinnytsia, 2024.

In the qualifying (bachelor's) work, the process of the concept of a web platform, its main principles and methods of creation were researched and analyzed. With the use of PHP, HTML, CSS technologies and Laravel, Tailwind frameworks, a web platform was developed, the purpose of which is the sale of animation-themed products.

Keywords: web platform, development, web design, database, PHP, HTML.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Розгляд та аналіз останніх досліджень.....	7
1.2 Аналіз аналогічних продуктів.....	11
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-САЙТУ	23
2.1 Функціональні та нефункціональні вимоги	23
2.2 Проектування бази даних	27
2.3 Реалізація бази даних	29
2.4 Модель роботи інтернет-магазину.....	35
РОЗДІЛ 3.РЕАЛІЗАЦІЯ РОЗРОБКИ ВЕБ-ПЛАТФОРМИ.....	37
3.1 Вибір технологій для розробки	37
3.2 Створення сторінки авторизації та реєстрації	43
3.3 Створення головної сторінки	45
3.4 Розробка сторінки перегляду товарів	47
3.5 Створення кошика	50
3.6 Тестування роботи розробленого продукту.....	51
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

PHP – Hypertext Preprocessor

БД – База даних

UI – User interface

SQL – Structured query language

СУБД – Система управління базою даних

ORM - Object-Relational Mapping

CSRF - Cross-Site Request Forgery

ВСТУП

Актуальність інтернет-магазинів у сучасному світі не може бути недооцінена. Із роками онлайн-торгівля стає все більш значущою та невід'ємною частиною повсякденного життя. Перш за все, інтернет надає можливість компаніям досягти клієнтів навіть за межами своєї країни. Це відкриває нові ринки та дозволяє залучити аудиторію, яка раніше була недосяжною.

Зручність для споживачів є ще однією важливою перевагою інтернет-магазинів. Клієнти можуть здійснювати покупки у будь-який час, знаходячись у зручному для них місці, що особливо цінно в умовах сучасного темпу життя.

З маркетингової точки зору, інтернет-магазини мають безліч можливостей для просування товарів і послуг. Вони можуть використовувати різноманітні інструменти цифрового маркетингу, такі як соціальні медіа, електронні розсилки та контент-маркетинг, щоб залучити увагу потенційних клієнтів.

Компанії постійно вдосконалюються, намагаючись забезпечити кращий сервіс для своїх клієнтів та оптимізувати процеси покупок.

Розвиток інтернет-магазинів на українському ринку також є очевидним. За останні роки спостерігається стрімкий ріст їх кількості та популярності серед споживачів. В Україні, як і в інших країнах, інтернет-торгівля стала невід'ємною складовою ринкових процесів, забезпечуючи підприємствам нові можливості для розвитку та збільшення обсягів продажів.

Отже, Інтернет-магазини стають не лише невід'ємною частиною сучасного бізнесу, а й надійним засобом розвитку та забезпеченням людей бажаними продуктами у будь-якій точці країни. Вони надають безліч переваг і зручностей, зокрема глобальний доступ, доступність цілодобово, розширений асортимент товарів та послуг, а також зручність порівняння цін і характеристик.

Об'єктом дослідження є процес розробки вебсторінки для інтернет-магазину анімаційної тематики, включаючи аспекти дизайну, функціональності, інтерактивності та користувацького досвіду.

Предмет дослідження: функціональні вимоги та вимоги до UI інтерфейсу вебплатформи.

Метою дослідження розробка вебплатформи інтернет-магазину анімаційної тематики з метою залучення цільової аудиторії та створення сприятливого користувацького досвіду.

Завдання дослідження полягає в:

- виборі інструментів реалізації проєкту;
- проєктуванні бази даних вебплатформи;
- встановленні основних вимог до UI дизайну;
- визначення функціональності та інтерактивності веб-сторінки інтернет-магазину.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Розгляд та аналіз останніх досліджень

Нещодавні дослідження та публікації, присвячені онлайн-шопінгу в Україні та світі, показують, що цей сегмент електронної комерції стрімко зростає. Дані досліджень свідчать про те, що інтернет-магазини продовжують набирати популярність серед споживачів як в Україні, так і в інших країнах.

Електронна комерція (e-commerce) - це процес купівлі та продажу товарів чи послуг через Інтернет. Цей термін включає в себе будь-яку форму комерційної діяльності, яка відбувається онлайн: від покупки товарів в інтернет-магазинах до електронних платежів, аукціонів, бронювання квитків та бронювання готелів через вебсайти [30].

Класифікація інтернет-магазинів є важливою для розуміння їхньої специфіки та особливостей. За типом товарів інтернет-магазини можуть бути загального вжитку або спеціалізованими. Товари загального вжитку охоплюють широкий асортимент товарів, які призначені для різних категорій споживачів, таких як електроніка, одяг, косметика тощо. Спеціалізовані магазини спеціалізуються на конкретному виді товарів або послуг, таких як книги, автозапчастини, харчові продукти або меблі.

За способом продажу інтернет-магазини можуть бути розділені на B2C (Business-to-Consumer) та B2B (Business-to-Business). B2C магазини продають товари безпосередньо кінцевим споживачам, тоді як B2B магазини постачають товари або послуги іншим компаніям або бізнесам.

За моделлю бізнесу інтернет-магазини можуть бути прямими або маркетплейсами. Прямі магазини володіють товаром або виробляють його самостійно та продають кінцевому споживачеві. Маркетплейси, навпаки, є платформами, які дозволяють різним продавцям розміщувати свої товари для продажу через одну спільну платформу.

За цільовою аудиторією інтернет-магазини можуть бути спрямовані на масові ринки або на певні сегменти ринку. Масові ринки орієнтовані на широке коло споживачів, тоді як спеціалізовані магазини можуть зосередитися на певних групах клієнтів або специфічних товарах.

Основні переваги електронної комерції включають доступність 24/7, можливість глобального ринку, зручність та широкий вибір товарів та послуг. Електронна комерція також дозволяє підприємствам ефективно використовувати дані про клієнтів для персоналізованих пропозицій та покращення обслуговування клієнтів.

Цей сегмент бізнесу постійно розвивається завдяки технологічним інноваціям, включаючи штучний інтелект, розширену реальність та блокчейн. Електронна комерція стала невід'ємною частиною сучасного бізнесу і продовжує зростати як важливий сегмент глобальної економіки.

Зокрема, розвиток електронної комерції та зростання обсягів онлайн-покупок. Що стосується України, де ринок електронної комерції розвивається досить стрімко, увага зосереджена на розумінні специфіки звичок, уподобань та ставлення українських споживачів до різних аспектів електронної комерції в Інтернеті.

Також зростає інтерес до розвитку електронних платежів та логістичної інфраструктури для забезпечення більш зручного та ефективного досвіду онлайн-покупок. При проведенні електронних платежів важливим аспектом є безпека. Сучасні платіжні системи використовують різноманітні технології для захисту даних користувачів, такі як SSL (Secure Socket Layer) для шифрування даних під час передачі, двофакторна автентифікація, токенизація (заміна конфіденційних даних спеціальними токенами) та інші методи. Ці заходи спрямовані на захист від шахрайства та несанкціонованого доступу до фінансової інформації. Зростає тенденція розвитку електронної комерції, новітніх технологій та інновацій, які впливають на роботу інтернет-магазинів [30].

Особлива увага приділяється впливу соціальних мереж, мобільних додатків, штучного інтелекту та інших сучасних технологій на процеси електронної комерції. Загалом, останні дослідження та публікації відображають стійкий рух і динаміку розвитку інтернет-магазинів як в Україні, так і в усьому світі, і свідчать про те, що цей сегмент електронної комерції залишається одним із найперспективніших і найбільш швидкозростаючих.

У сфері електронної комерції, одним з важливих аспектів є необхідність детальної оцінки продукту з боку клієнта, особливо коли йдеться про товари, які можна оцінити за допомогою дотику або перевірки на собі, наприклад, одяг. Це важливо, оскільки вірогідність успішної та задовільної покупки залежить від того, наскільки клієнт може отримати реальне уявлення про продукт перед придбанням.

У таких випадках, важливо, щоб інтернет-магазин надавав можливість докладного огляду продукту, включаючи детальні фотографії з різних кутів, можливість збільшення зображень для більшої чіткості, а також відгуки та описи, які допоможуть клієнтові зрозуміти, що він отримає.

Деякі інтернет-магазини також пропонують варіанти віртуальної примірки або 360-градусний перегляд товарів, що дозволяє клієнтам краще оцінити продукт перед покупкою. Такі функції допомагають зменшити ризик невдачі з покупкою через відсутність можливості особистого огляду товару.

Отже, враховуючи нюанс електронної комерції, важливо, щоб інтернет-магазини надавали клієнтам достатньо інформації та можливостей для детальної оцінки продукту, забезпечуючи таким чином задоволення від покупки та підвищуючи рівень довіри споживачів.

В інтернеті існує безліч магазинів, які пропонують різноманітні товари, але серед них особливе місце займають ті, які спеціалізуються на анімаційній тематиці. Аніме, манга, та інші анімаційні продукти є об'єктом прагнень для колекціонерів, любителів та шанувальників цього жанру. Ця культура останніми роками набирає все більших обертів серед української молоді, але стаціонарних магазинів ще не так багато. Враховуючи цей попит, розглядається створення

онлайн-магазину, який надаватиме можливість легкого доступу до цих унікальних продуктів. В даному контексті, важливо розглянути ключові аспекти успішності такого магазину, щоб забезпечити якісний та задовільний досвід для клієнтів.

За останні роки спостерігається зростаюча тенденція до популяризації японської манги, зокрема серед молоді. Манга та японські комікси займають важливе місце у сучасній культурі, особливо серед молоді в Україні. Манга, як унікальна форма мистецтва, пропонує читачам захоплюючі сюжети, які часто відрізняються глибиною та різноманітним тем. Вона одночасно відноситься і до образотворчого мистецтва, і до літератури. Завдяки цьому, японські комікси здобули значну популярність серед українських підлітків та молодих людей, які знаходять у них джерело натхнення, розваги та пізнання нових культурних аспектів.

Роль манги у формуванні смаків та уподобань молоді є значною. Вона впливає на їхній стиль, захоплення, а також сприяє розвитку творчих здібностей. Манга часто стає поштовхом до вивчення японської мови, культури та традицій, що розширює світогляд і культурний горизонт молодих українців. У той же час, манга є важливою частиною спільнот фанатів, де молодь може знайти однодумців, обмінюватися думками та обговорювати улюблені твори.

Таким чином, культура манги та японських коміксів стає все більш інтегрованою у повсякденне життя сучасної молоді в Україні, відіграючи значну роль у їхньому культурному та особистісному розвитку.

Так, з великим зростанням онлайн-торгівлі та ростом популярності товарів японської анімації створення веб-платформи для онлайн-магазину є доволі актуальною ідеєю. Стаціонарних магазинів з подібним асортиментом ще доволі мало, оскільки ця культура тільки розвивається, тому онлайн-магазин дозволить залучити аудиторію з усієї країни, а можливо й з інших країн, що забезпечить доступ до бажаних товарів для будь-кого.

У бакалаврській роботі описано процес розробки веб платформи B2C маркетплейсу зі спеціалізованим типом товарів, що спрямоване на певну цільову категорію.

1.2 Аналіз аналогічних продуктів

Інтернет-магазини можуть бути різних типів, залежно від специфіки продуктів, які вони продають, та характеру бізнесу. Серед них можна виділити такі, як маркетплейс та інтернет-магазин.

Маркетплейс і інтернет-магазин - це два різних типи електронних торгових платформ, які пропонують товари та послуги через Інтернет. Вони мають свої особливості і функціонал, але основна різниця між ними полягає в тому, як вони організовані та яким чином товари представлені користувачам.

Маркетплейс - це онлайн-платформа, де кілька незалежних продавців продають свої товари або послуги. На маркетплейсі можуть бути представлені товари різних брендів та постачальників, які реалізують свою продукцію через цю платформу. Користувач може порівнювати ціни, вибирати товари від різних продавців та робити покупки на одному сайті.

Інтернет-магазин - це онлайн-торгова платформа, яка належить одній компанії або бренду, і продає товари або послуги лише цієї компанії. Товари, представлені в інтернет-магазині, можуть бути тільки від одного постачальника або виготовлятися самою компанією, що і буде реалізовано у дипломній роботі. Для проведення аналізу першого аналогу продукту взято інтернет-магазин “Malopus”. Це магазин товарів анімаційної тематики, який пропонує такі товари, як манга, комікси, мерч тощо.



Рисунок 1.1 - Логотип інтернет-магазину аналогу №1

При вході на сайт можна побачити головну сторінку, на якій розміщений баннер, що містить різні пропозиції до купівлі різного жанру манги, мерчу, надає інформацію про нові надходження та перепродажу, що ми можемо побачити на рис. 1.2. Також можна відмітити кнопку зворотнього відгуку.

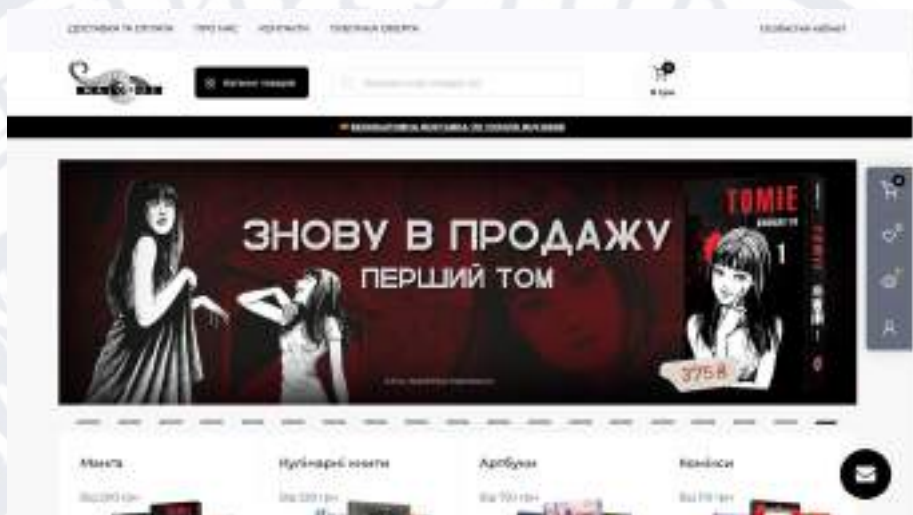


Рисунок 1.2 - Головна сторінка сайту-аналогу

При першому візиті на сайт можна одразу помітити монохромний дизайн сторінки, який доволі лаконічно поєднується з кольоровими обкладинками баннерів та манг. Інтерфейс націлений на користувача, тому дуже простий і інтуїтивно зрозумілий, що є невід'ємною частиною будь-якого інтерфейсу.

На головній сторінці сайту розміщений розділ "Новинки", який привертає увагу користувачів до останніх надходжень в асортименті магазину. Цей розділ містить широкий вибір різноманітних товарів, які включають у себе мангу, артбуки та комікси. Кожен продукт у цьому розділі супроводжується інформацією про його тип, назву та ціну, щоб користувачі могли швидко оцінити його.

Крім того, поруч із кожним товаром розміщена кнопка швидкого додавання до кошика, що дозволяє зручно та швидко придбати обраний товар безпосередньо з головної сторінки сайту. Цей розділ створений для того, щоб забезпечити користувачам максимальну зручність та швидкість в процесі вибору та покупки новинок, що надійшли до асортименту магазину (рис. 1.3).

На сайті бокове меню є важливим елементом навігації, оскільки воно містить такі корисні функції, як кошик, уподобані товари та кнопка профілю.

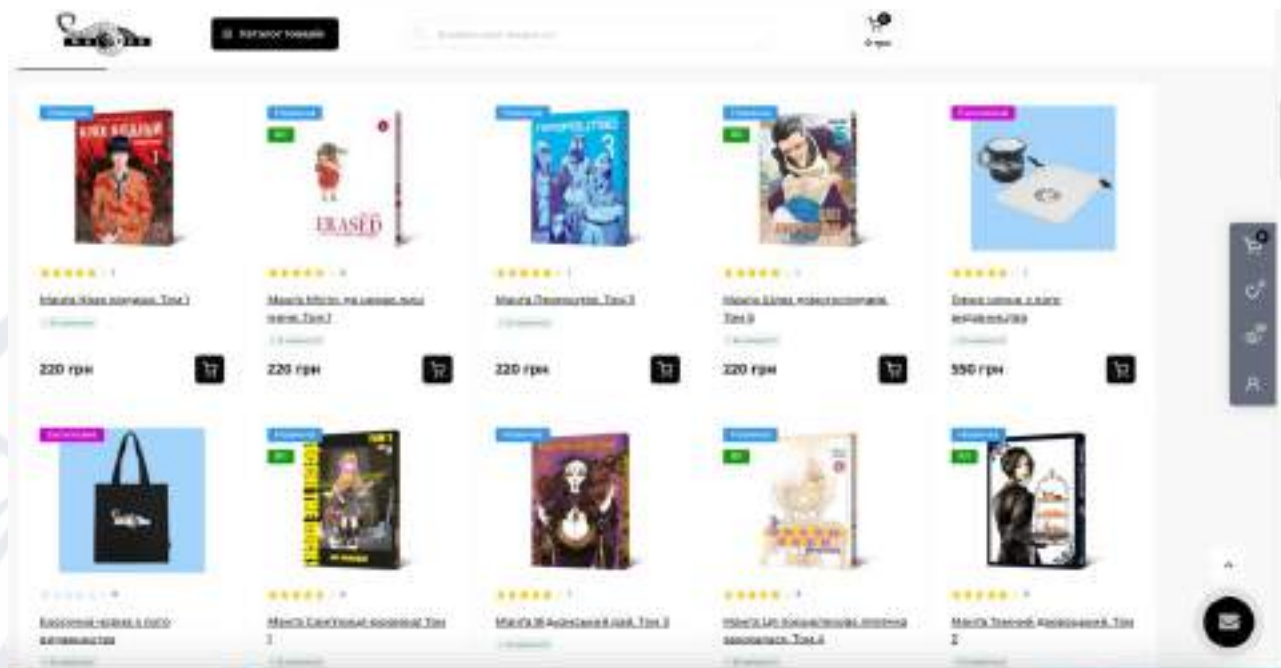


Рисунок 1.3 - Список товарів

Ця стратегічно розташована панель забезпечує зручний доступ до основних функцій сайту навіть під час перегляду інших сторінок. Користувачам не потрібно витрачати час на пошук цих опцій, оскільки вони завжди легко доступні. Такий підхід до розташування функціоналу сприяє покращенню загального досвіду користувача та робить процес онлайн-шопінгу більш зручним і ефективним.

Також на цьому сайті присутнє Dropdown меню. Dropdown меню — це елемент інтерфейсу користувача на веб-сайті або у додатку, який представляє собою список опцій або посилань, що розкривається, коли користувач натискає на певну кнопку або наводить на неї курсор. Dropdown меню дозволяє зберігати простір на сторінці, організовуючи велику кількість опцій у компактній формі, яка розкривається лише за потреби. Це покращує зручність навігації та робить інтерфейс більш чистим і структурованим.

Dropdown меню є одним з ключових елементів сучасного веб-дизайну, що значно підвищує зручність та ефективність навігації на сайті. Цей елемент інтерфейсу дозволяє користувачам легко знаходити потрібні розділи та функції,

зберігаючи при цьому простір на сторінці і забезпечуючи чистий, організований вигляд.

Dropdown меню забезпечує зручність для користувачів, дозволяючи їм швидко та легко отримати доступ до великої кількості опцій без перевантаження інтерфейсу. Це особливо важливо для сайтів з великою кількістю категорій або підкатегорій, таких як інтернет-магазини. Користувач може просто навести курсор на основне меню і побачити всі доступні підкатегорії, що значно спрощує процес навігації.

Ефективне використання простору є ще однією перевагою dropdown меню. Веб-дизайнери можуть використовувати цей елемент для розміщення великої кількості посилань і опцій в компактній формі. Це допомагає зберегти мінімалістичний дизайн сторінки, зосередивши увагу користувачів на основному контенті, без відволікання на надмірну кількість елементів.

Таке меню також сприяє організації та структурованості контенту. За допомогою цього елемента можна логічно групувати схожі опції та функції, роблячи інтерфейс більш інтуїтивно зрозумілим. Це дозволяє користувачам легко знаходити потрібну інформацію або функціональність, що підвищує загальну ефективність використання сайту.

Реалізація dropdown меню може бути виконана різними способами, включаючи використання HTML, CSS та JavaScript. Основна структура меню створюється за допомогою HTML, стилізація та приховування/відображення опцій здійснюється за допомогою CSS, а для додавання інтерактивності часто використовується JavaScript. Використання сучасних фреймворків, таких як Bootstrap, може значно спростити процес створення dropdown меню, забезпечуючи готові рішення та стилі [26].

Загалом, dropdown меню є важливим інструментом для покращення користувацького досвіду та ефективного управління контентом на сайті. Його правильна реалізація може значно підвищити зручність навігації, зберігаючи при цьому естетичний вигляд і функціональність вебсторінки.

Як можна побачити на рис. 1.4, у меню присутні усі категорії товарів, які наявні у магазині.



Рисунок 1.4 – Dropdown меню

У панелі навігації сайту наявна додаткова функція пошуку, яка робить процес знаходження конкретних товарів ще зручнішим. Це дозволяє ефективно використовувати час та мінімізувати зусилля при пошуку конкретного товару серед широкого асортименту, що пропонується на сайті. Така можливість пошуку створює додаткову зручність для користувачів, допомагаючи їм швидко знаходити те, що вони шукають, і роблячи процес покупок онлайн ще приємнішим і ефективнішим.

Можливість передзамовлення продуктів, які ще не вийшли на ринок, є однією з найпривабливіших особливостей для багатьох покупців. Це дозволяє їм отримати доступ до найновіших релізів за спеціальною ціною, а також отримати бажаний товар одразу після його випуску. Це може бути особливо актуально для популярних продуктів, які, ймовірно, швидко розпродаються.

З переваг передзамовлення продуктів можна виділити наступні пункти:

- Гарантований доступ: Передзамовлення гарантує, що ви отримаєте товар, коли він буде випущений, навіть якщо він швидко розпродається.
- Спеціальні ціни: Багато продавців пропонують знижки або інші спеціальні пропозиції для тих, хто робить передзамовлення.

- Ранній доступ: У деяких випадках ви можете отримати ранній доступ до продукту, якщо зробите передзамовлення.
- Ексклюзивні бонуси: Деякі продавці пропонують ексклюзивні бонуси для тих, хто робить передзамовлення, наприклад, додаткові аксесуари або цифрові матеріали.

Перейшовши на сторінку безпосередньо певного товару (рис 1.5) можна побачити всю основну інформацію про товар загальний: опис манги, серію, дату виходу, автора. Доступна функція вибору кількості екземплярів для замовлення і власне сама кнопка додання до кошику.



Рисунок 1.5 - Інтерфейс сторінки продукту

Сторінка оформлення замовлення (рис. 1.6) є ключовою частиною будь-якого інтернет-магазину. Її інтуїтивний інтерфейс допомагає користувачам легко та швидко заповнити необхідну інформацію для замовлення. На цій сторінці можна знайти такі елементи, як ім'я, номер телефону, спосіб оплати та доставки.

Рисунок 1.6 - Сторінка оформлення замовлення

В якості другого аналогу продукту було обрано інтернет-магазин коміксів та аніме-товарів YoroKobi (рис 1.7). Даний інтернет-магазин пропонує дещо більший перелік товарів – від манг та коміксів до тематичного посуду.



Рисунок 1.7 - Логотип інтернет-магазину аналогу №2

Одразу користувача зустрічає головна сторінка (рис. 1.8), котра оформлена у яскравих помаранчевих кольорах, що створює відповідну атмосферу для інтернет-магазину аніме-культури. Дизайн сторінки орієнтований на зручність користувача та швидкий доступ до основних розділів сайту. Приваблює відсутність реклами.

Верхнє меню навігації містить основні пункти для швидкого переходу: "Головна", "Товари", "Новинки", "Доставка", "Повернення та обмін".

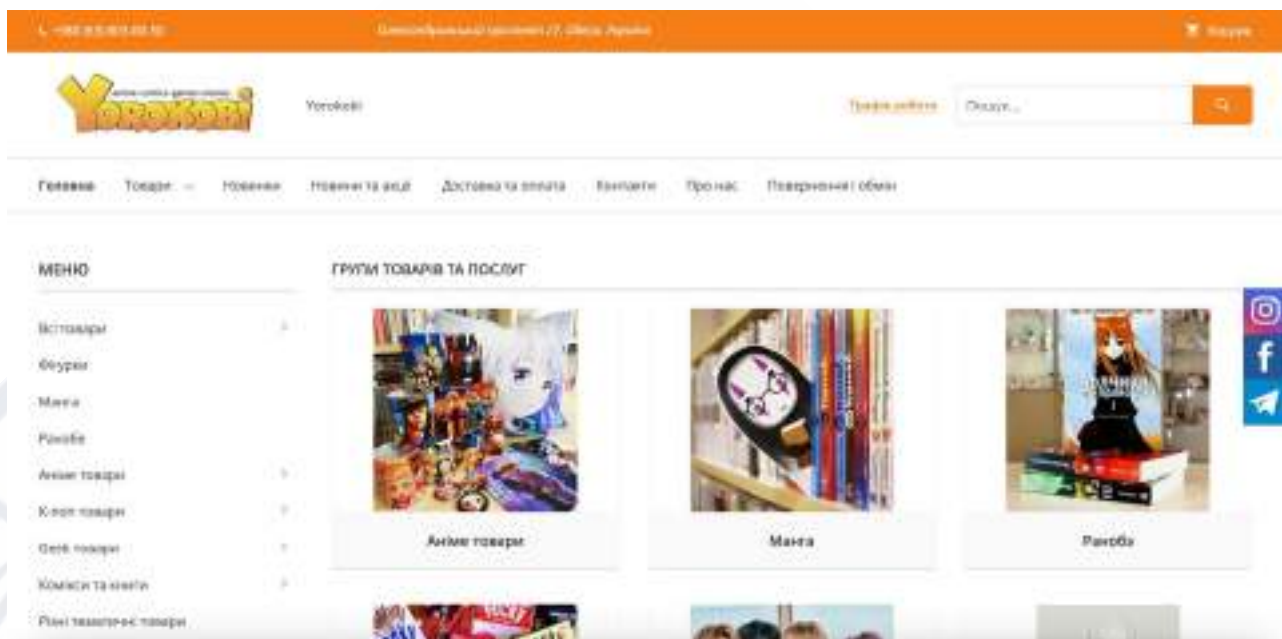


Рисунок 1.8 – Головна сторінка інтернет-магазину аналогу

Ліворуч розташоване меню з категоріями товарів, що дозволяє користувачам легко орієнтуватися серед різноманіття продукції. Категорії включають такі розділи, як манга, фігурки, ранобе, різні тематичні товари та інші товари, пов'язані з аніме. Це меню полегшує пошук і вибір потрібних товарів, роблячи навігацію інтуїтивно зрозумілою.

Праворуч знаходяться три кнопки з посиланнями на соціальні мережі магазину, такі як Facebook, Instagram та Twitter. Це дозволяє користувачам легко підписатися на новини магазину та дізнаватися про нові надходження та акції.

Соціальні мережі стають важливим каналом комунікації, де магазин може взаємодіяти зі своїми клієнтами.

У правому верхньому кутку сторінки розташована стрічка пошуку товарів, яка дозволяє користувачам швидко знаходити потрібні продукти за ключовими словами або назвами. Поруч із нею знаходиться кнопка кошика, яка відображає кількість доданих товарів і надає швидкий доступ до списку покупок. Це спрощує процес навігації та оформлення замовлення, забезпечуючи зручність і ефективність для користувачів.

При перегляді товарів певної категорії наявний детальний фільтр товарів (рис.1.9), щоб забезпечити швидкий та ефективний пошук конкретних бажаних

товарів. Користувач може вказати бажаний діапазон цін, наявність чи відсутність акції на продукти, тип товару та інше.

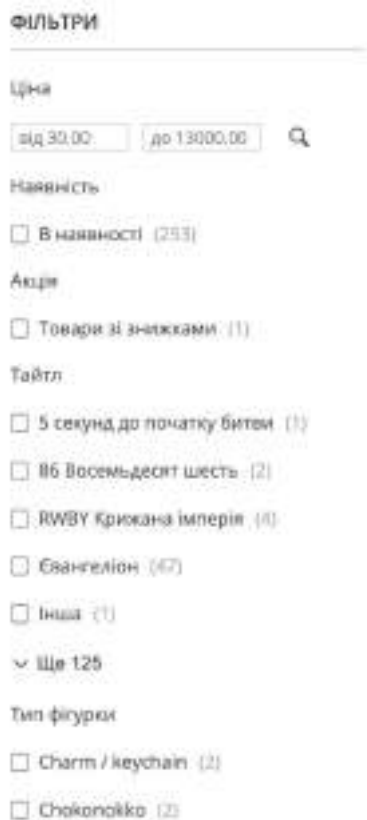


Рисунок 1.9 – Фільтр товарів

При натисканні на бажаний товар користувач переходить на сторінку з його описом (рис. 1.10). Там він може ознайомитися з ціною та наявністю товару, а також з умовами оплати, доставки та повернення товару, та, найголовніше, купити.



Рисунок 1.10 - Сторінка перегляду окремого товару

Також на сайті присутня функція зручного перегляду товару. Під час перегляду товарів на веб-сайті інтернет-магазину часто виникає потреба детальніше розглянути обкладинку, щоб краще оцінити товар. Ця функція дає змогу користувачам зручно та без зайвих зусиль це зробити.

Користувач наводить курсор миші на обкладинку товару, який його зацікавив та замість простого збільшення зображення, на екрані з'являється збільшена копія обкладинки, не затуляючи інші елементи сторінки, що дозволяє користувачеві детально розглянути товар, не втрачаючи контексту сторінки.

Завдяки цій функції користувачу можна роздивитися дрібні деталі товару, наприклад, візерунки або текстуру, та краще оцінити колір та деталі товару, а також переконатися, що товар відповідає їхнім очікуванням.

Прогортавши до низу користувач має змогу ознайомитись з характеристиками товару (рис. 1.11), такими як видавництво, кількість томів, матеріал, обкладинка, тайтл, тип поліграфічного паперу та мову, на яку манга перекладена.

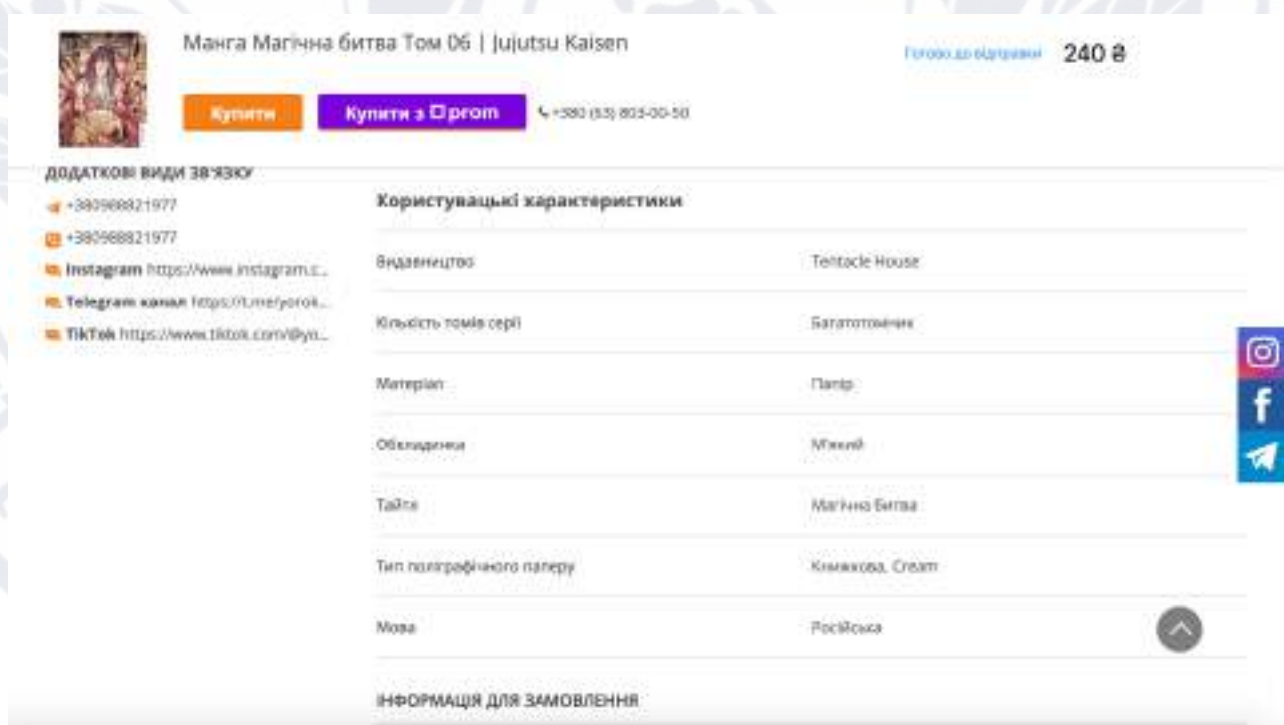


Рисунок 1.11 – Характеристики товару

Після натискання на кнопку «купити» товар додається до кошика. До нього можна додати безліч товарів і потім переглянути це у вигляді зручного списку (рис. 1.12).

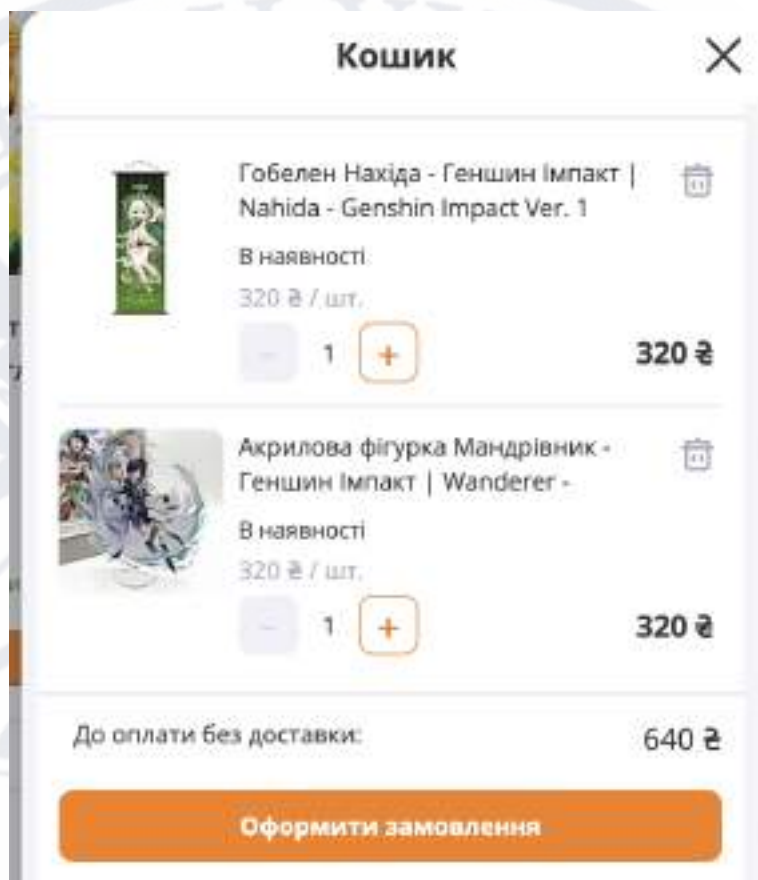


Рисунок 1.12 - Кошик аналогу інтернет-магазину №2

Отже, розглянувши аналоги продуктів потрібної тематики, можна виділити кілька ключових моментів, за якими можна оцінити веб-сайти, а саме:

- Актуальність дизайну
- Наявність реклами
- Зручність навігації
- Зворотній зв'язок
- Функція пошуку товарів

Підсумування усіх основних критеріїв веб платформ у табл. 1.1:

Табл. 1.1 – Підсумування аналізу критерій

Додаток Критерії	Аналог №1	Аналог №2	Веб-додаток бакалаврського проекту
Актуальність дизайну	+	-	+
Зручність навігації	+	+	+
Зворотній зв'язок	+	-	+
Функція пошуку товарів	+	+	+
Функція збільшення фото	-	+	-

РОЗДІЛ 2

ПРОЕКТУВАННЯ ВЕБ-САЙТУ

2.1 Функціональні та нефункціональні вимоги

Розробка веб-сайту для інтернет-магазину манги передбачає врахування різноманітних функціональних і нефункціональних вимог, щоб забезпечити зручну та ефективну роботу платформи для користувачів. Важливо забезпечити інтуїтивний інтерфейс, який дозволить легко знаходити потрібні товари, а також надавати детальну інформацію про кожен продукт. Функціональні вимоги - це можливості та поведінка, які повинен забезпечувати веб-сайт, щоб відповідати потребам і очікуванням його користувачів і зацікавлених сторін. Вони описують, що веб-сайт повинен робити, а не як він повинен це робити [31].

Система реєстрації та авторизації є одним із найважливіших компонентів будь-якого сучасного вебсайту, що забезпечує безпеку, персоналізацію та контроль доступу для користувачів. Вона складається з декількох ключових елементів і процесів, які забезпечують надійний захист даних і зручність використання.

Реєстрація користувачів є першим кроком для тих, хто хоче створити обліковий запис на вашому сайті. Вона зазвичай включає заповнення форми з основною інформацією, такою як ім'я користувача, адреса електронної пошти та пароль. Ім'я користувача є унікальним ідентифікатором, який буде використовуватися для входу в систему. Адреса електронної пошти використовується для підтвердження реєстрації та відновлення паролю, а пароль забезпечує захищений спосіб входу. Після заповнення форми користувачеві може бути надіслано підтвердження електронною поштою для верифікації облікового запису, що додає додатковий рівень безпеки.

Процес авторизації дозволяє зареєстрованим користувачам входити в свої облікові записи. Користувач вводить свої облікові дані, такі як ім'я користувача або електронну пошту і пароль. Система перевіряє ці дані на відповідність тим, що зберігаються в базі даних. Якщо облікові дані вірні, користувач отримує

доступ до свого облікового запису. Важливим аспектом є захист цих даних за допомогою сучасних методів шифрування, таких як хешування паролів, щоб гарантувати їхню безпеку.

Одним із важливих компонентів системи авторизації є функція відновлення паролю. У разі, якщо користувач забув свій пароль, він може скористатися функцією відновлення, яка зазвичай включає відправлення на електронну пошту користувача посилання для скидання паролю. Це посилання дозволяє користувачеві створити новий пароль, зберігаючи при цьому безпеку його облікового запису.

Додаткові функції, такі як двофакторна аутентифікація (2FA), можуть значно підвищити рівень безпеки. Двофакторна аутентифікація додає ще один шар захисту, вимагаючи від користувача не лише ввести пароль, але й підтвердити свою особу за допомогою додаткового методу, такого як код, надісланий на мобільний телефон.

Загалом, система реєстрації та авторизації є необхідним компонентом для забезпечення безпеки, зручності та персоналізації на вашому сайті. Вона не лише захищає дані користувачів, але й дозволяє забезпечити їм персоналізований досвід, надаючи доступ до спеціальних функцій та налаштувань.

Пошук і фільтрація товарів також важливі функціональні елементи, що забезпечують зручність користувачам у пошуку конкретних продуктів або відображенні товарів за певними критеріями, такими як автор, жанр, серія тощо. Детальний перегляд товарів з описом та зображеннями дозволить користувачам краще ознайомитися з продукцією перед покупкою. Це може включати опис сюжету, рейтинги, відгуки користувачів, а також зображення обкладинок та внутрішніх сторінок манги.

Кошик для зберігання товарів і оформлення замовлення є необхідною функцією, що дозволяє користувачам збирати товари перед оплатою та забезпечує зручний процес покупки. Оплата та вибір способу доставки - це ключові функціональні можливості, які забезпечують користувачам зручний спосіб оплати за товари та вибір способу доставки, який найбільше їм підходить.

Отже, з функціональних вимог (рис. 2.1) можна виділити наступні пункти:

- авторизація та реєстрація;
- пошук і фільтрація;
- детальний перегляд;
- кошик;
- оплата;
- забезпечення вибору способу доставки.



Рисунок 2.1 - Функціональні вимоги до сайту

Нефункціональні вимоги - це вимоги, які визначають, наскільки добре система працює, включаючи такі речі, як швидкість, безпека, надійність, цілісність даних тощо [6].

З нефункціональних вимог слід звернути увагу на швидкість та ефективність сайту. Сайт повинен завантажуватися швидко і працювати без затримок навіть при великому навантаженні користувачів.

Зручний інтерфейс для користувачів передбачає легкий та зрозумілий дизайн сайту, логічну навігацію та інтуїтивно зрозумілі елементи керування.

Безпека даних є ще однією важливою нефункціональною вимогою, що передбачає захист персональної інформації користувачів та безпечність платіжних операцій.

Сумісність та адаптивність до різних пристроїв і розмірів екранів забезпечує зручний доступ до сайту з будь-якого пристрою, такого як комп'ютер, смартфон або планшет.

Доступність для всіх категорій користувачів означає, що сайт повинен бути доступним для людей з різними потребами та обмеженнями, включаючи людей з обмеженими можливостями.

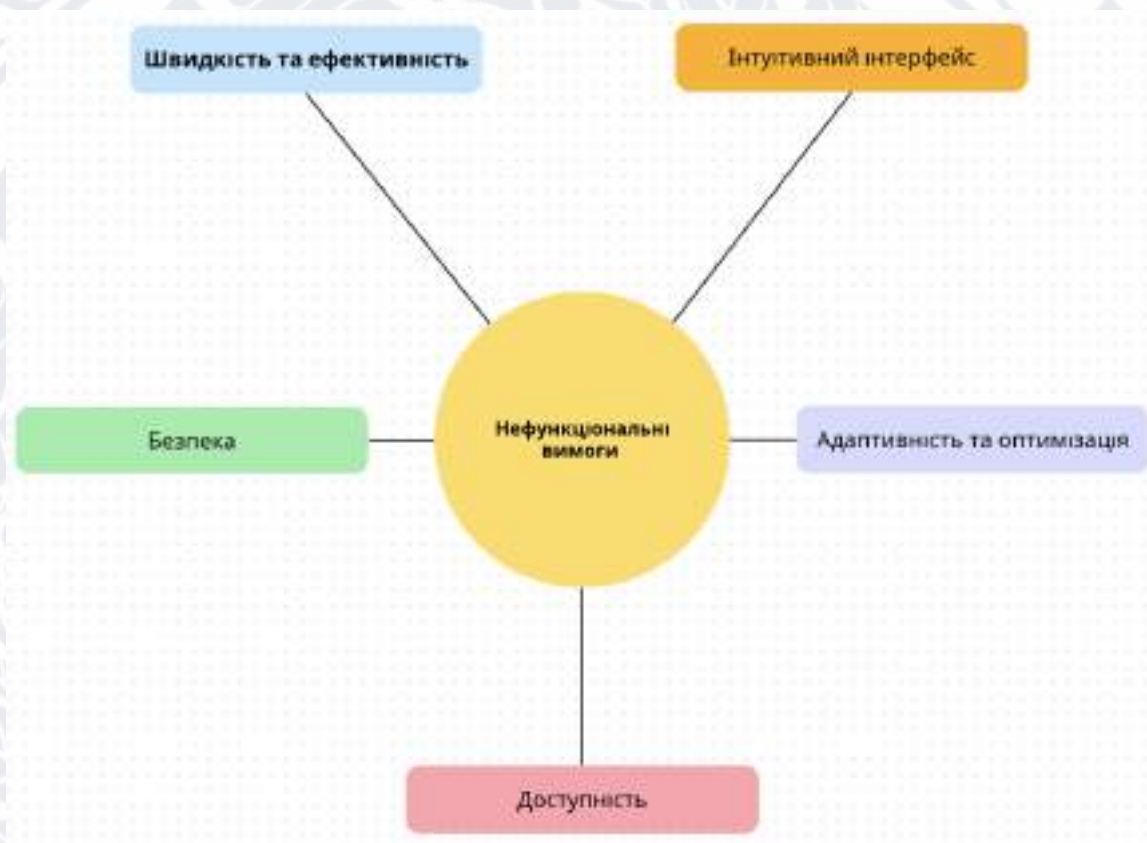


Рисунок 2.2 - Нefункціональні вимоги до сайту

Отже, з нефункціональних вимог можна виділити:

- забезпечення безпеки даних;
- швидкість і ефективність;
- інтуїтивно зрозумілий інтерфейс;
- адаптивність та оптимізація;
- доступність.

2.2 Проектування бази даних

База даних відіграє ключову роль у функціонуванні веб-сайту для інтернет-магазину манги. Вона служить основою для зберігання, організації та управління великим обсягом інформації, що необхідна для ефективної роботи сайту.

База даних - це організована колекція даних, яка структурована таким чином, щоб полегшити доступ, управління і оновлення інформації. Дані у базі даних зберігаються у вигляді таблиць, які складаються з рядків і стовпців. Кожен рядок представляє окремий запис, а кожен стовпець - окреме поле з певним типом даних [9].

Серед найбільш поширених типів баз даних можна виділити реляційні та нереляційні бази даних. Реляційні бази даних, такі як MySQL, PostgreSQL і Oracle, зберігають дані в таблицях і використовують SQL (Structured Query Language) для управління даними та запитів. Вони забезпечують високу рівень структурованості і підтримують складні запити та транзакції.

Нереляційні бази даних, або NoSQL бази даних, такі як MongoDB, Cassandra і Redis, використовують різні моделі зберігання даних, включаючи документи, графи, стовпці та ключ-значення. Вони забезпечують гнучкість у зберіганні даних і часто використовуються для роботи з великими обсягами неструктурованих даних, де важлива швидкість і масштабованість.

Бази даних надають різноманітні функції для ефективного управління даними. Серед них:

- зберігання та організація даних: Забезпечують структуроване зберігання даних, що полегшує їхній доступ і управління;
- запити та маніпуляція даними: Дозволяють користувачам отримувати необхідні дані за допомогою запитів, а також додавати, оновлювати та видаляти записи;
- індексація: Підвищують швидкість пошуку і доступу до даних шляхом створення індексів для ключових полів;
- забезпечення цілісності даних: Підтримують цілісність і узгодженість даних через використання обмежень та транзакцій;

- безпека: Забезпечують захист даних за допомогою контролю доступу, шифрування та інших механізмів безпеки;
- резервне копіювання та відновлення: Дозволяють створювати резервні копії даних та відновлювати їх у разі втрати або пошкодження.

У веб-розробці бази даних використовуються для зберігання та управління всіма типами даних, від облікових записів користувачів і транзакцій до вмісту сайтів та журналів дій. Сервери баз даних взаємодіють з веб-серверами через спеціальні інтерфейси або API, забезпечуючи швидкий доступ до необхідної інформації.

Для роботи з базами даних у веб-розробці часто використовуються ORM системи, які дозволяють розробникам працювати з базами даних за допомогою об'єктно-орієнтованих мов програмування, таких як Python або JavaScript. Це спрощує процес розробки, зменшуючи необхідність написання складних SQL-запитів.

Отже, можна виділити два основні аспекти, які демонструють важливість бази даних, що включають наступне:

1. Зберігання інформації про товари. База даних зберігає детальну інформацію про всі товари, що представлені в інтернет-магазині. Це включає назву товару, його опис, ціну, наявність на складі, зображення, жанр, автора та інші атрибути. Завдяки цьому користувачі можуть легко знайти та ознайомитися з будь-яким продуктом, доступним на сайті.
2. Управління користувачами. База даних містить інформацію про зареєстрованих користувачів, включаючи їхні облікові дані, історію покупок, уподобання та відгуки. Це дозволяє створювати персоналізований досвід для кожного користувача, пропонуючи їм товари, які можуть бути цікаві саме їм, на основі попередніх покупок та переглядів.

Для розробки запропонованої інформаційної системи була розроблена схема бази даних для ефективного та зручного продажу. Інформація про базу даних представлена в табл. 2.1.

Табл. 2.1 – Елементи бази даних

Назва таблиці	Опис
products	Містить перелік товарів
product_genres	Містить відповідності між товаром і їх жанром
users	Містить інформацію о користувачі
password_reset_tokens	Таблиця, що містить токени для відновлення пароля
genres	Містить перелік жанрів

2.3 Реалізація бази даних

Реалізація бази даних є одним з ключових етапів розробки інформаційної системи, що забезпечує зберігання, доступ та управління даними. Цей процес включає в себе проектування структури бази даних, вибір відповідної системи управління базами даних, створення таблиць, визначення ключових елементів і налаштування зв'язків між ними. Правильна реалізація бази даних є критично важливою для забезпечення надійності, ефективності та масштабованості системи, а також для гарантування цілісності і безпеки даних.

У кожній таблиці бази даних є ключовий елемент, який відіграє критичну роль у забезпеченні унікальності та зв'язності даних. Ці ключові елементи називаються ключами, і вони є фундаментальними для структури і функціонування реляційних баз даних. Є декілька видів ключових елементів: первинний, зовнішній та унікальний ключі [39].

Первинний ключ (Primary Key) — це унікальний ідентифікатор кожного запису в таблиці. Він гарантує, що кожен рядок у таблиці має унікальне значення для поля або комбінації полів, які виступають як первинний ключ. Первинні ключі використовуються для забезпечення цілісності даних і уникнення дублювання записів. Наприклад, у таблиці "Користувачі" первинним ключем може бути унікальний ідентифікаційний номер користувача (ID).

Зовнішній ключ (Foreign Key) — це поле або набір полів у таблиці, яке посиляється на первинний ключ у іншій таблиці. Зовнішні ключі забезпечують зв'язок між таблицями, створюючи реляційні зв'язки. Вони гарантують цілісність даних, забезпечуючи, що значення зовнішнього ключа відповідають існуючим значенням первинного ключа в пов'язаній таблиці. Наприклад, у таблиці "Замовлення" зовнішній ключ може посилатися на ID користувача з таблиці "Користувачі", що дозволяє зв'язувати замовлення з конкретними користувачами.

Унікальний ключ (Unique Key) — це поле або набір полів, яке забезпечує унікальність значень у цьому полі або наборі полів у таблиці. Унікальні ключі дозволяють уникнути дублювання даних в певних стовпцях, але на відміну від первинного ключа, таблиця може мати кілька унікальних ключів. Наприклад, у таблиці "Користувачі" поле "Електронна пошта" може бути унікальним ключем, щоб гарантувати, що кожна електронна адреса реєструється лише один раз.

Також існують індекси. Індокси - це додаткові структури даних, які прискорюють пошук і доступ до даних у таблиці. Індокси створюються на основі одного або кількох полів і значно підвищують продуктивність запитів до бази даних. Хоча індекси не є ключами, вони часто використовуються у поєднанні з ключами для оптимізації запитів. Наприклад, індекс на полі "Прізвище" у таблиці "Користувачі" дозволяє швидше виконувати пошук користувачів за прізвищем.

Отже, у всіх ключових елементів є важливе значення. Вони забезпечують структурованість і взаємозв'язок даних, що є основою реляційної моделі, а також гарантують цілісність та унікальність даних, дозволяють створювати складні

зв'язки між таблицями і оптимізувати продуктивність системи. Розуміння і правильне використання ключових елементів є критично важливими для ефективного проєктування і управління базами даних.

Для представлення архітектури реляційної бази даних, що відповідає вимогам додатку, було створено схему. Рис. 2.3 ілюструє таблиці даних з окремими стовпчиками та зв'язки між ними.

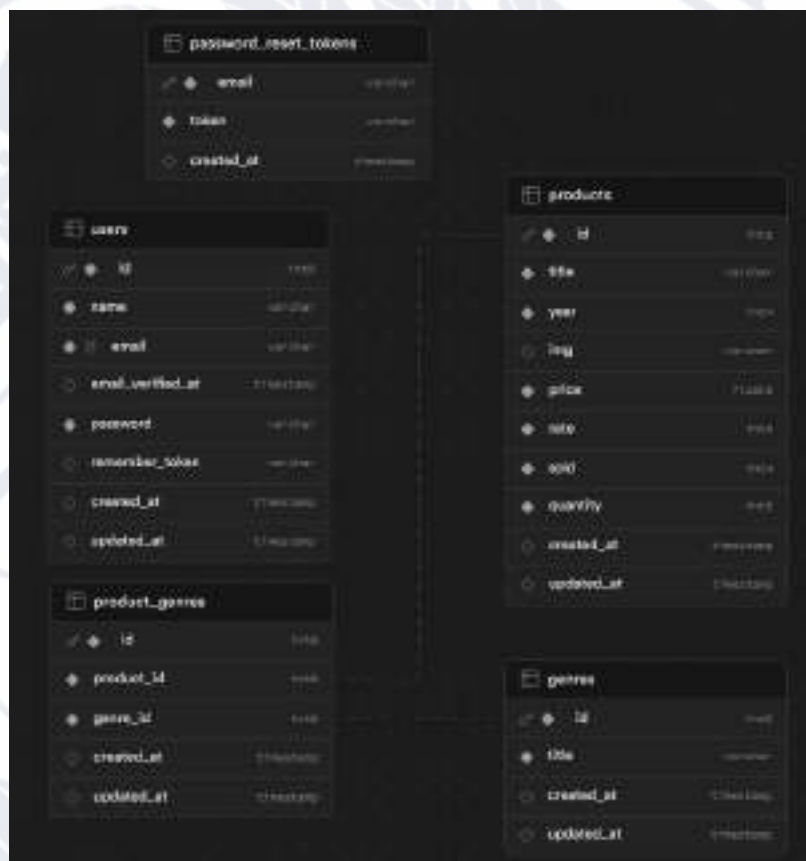


Рисунок 2.3 – Схема бази-даних

Для створення таблиці бази даних за допомогою PHP часто використовуються ORM системи, такі як Eloquent в Laravel. ORM - це програмна технологія, яка дозволяє взаємодіяти з об'єктно-орієнтованим кодом (написаним на мовах програмування, які підтримують об'єктно-орієнтовану парадигму, таких як Python, Java, або PHP) і реляційними базами даних. ORM виконує завдання відображення об'єктів програмного коду на записи бази даних і навпаки, забезпечуючи зручний спосіб роботи з даними без прямої роботи з SQL-запитами. ORM дозволяє створювати, отримувати, оновлювати та видаляти записи в базі даних, використовуючи методи об'єктів програмного коду, що

відповідають моделям даних. Також ORM може перетворювати вирази, що відображають об'єктні моделі, на відповідні SQL-запити, що дозволяє виконувати складні операції з базою даних без прямої роботи з SQL. До того ж ORM дозволяє визначати зв'язки між об'єктами (наприклад, один до одного, один до багатьох, багато до багатьох) і автоматично обробляти ці відносини при взаємодії з даними [38].

У структурі бази даних буде реалізовано кілька окремих міграцій, кожна з яких відповідатиме за створення певних таблиць з інформацією про товари, жанри та встановлення зв'язків між мангами та відповідними жанрами. Це дозволить створити систему, що забезпечує зручне та ефективне зберігання та управління даними про манги та їх класифікації за жанрами. Кожна міграція буде містити логіку створення конкретної таблиці з необхідними полями, а також визначення зв'язків між таблицями, які допоможуть у зручному та ефективному доступі до даних. Такий підхід сприятиме створенню добре організованої бази даних, що відповідає потребам проєкту і забезпечує простоту та ефективність у роботі з інформацією.

Клас міграції в Laravel — це спеціальний клас, який використовується для керування схемою бази даних. Він дозволяє створювати, змінювати та видаляти таблиці та стовпці в базі даних. Міграції забезпечують версіонування структури бази даних, що полегшує керування змінами та розгортанням цих змін на різних середовищах (наприклад, на локальному комп'ютері розробника, на тестовому сервері та на продакшн сервері) [2].

До переваг використання міграцій можна віднести полегшене керування змінами в структурі бази даних. Міграції дозволяють легко відслідковувати та керувати всіма змінами, що відбуваються в схемі бази даних, забезпечуючи чітке розуміння і контроль над процесом модифікації.

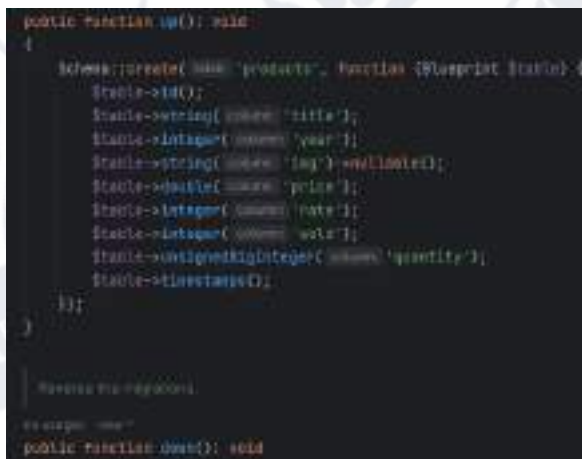
Ще однією важливою перевагою є синхронізація середовищ. Завдяки міграціям забезпечується однакова структура бази даних на всіх середовищах, будь то локальний комп'ютер розробника, тестовий сервер або продакшн сервер.

Це сприяє зменшенню помилок, пов'язаних із невідповідностями між середовищами.

Також варто зазначити інкрементальні зміни, які надають можливість поступового застосування змін. Це дозволяє зменшити ризик помилок, оскільки зміни можна впроваджувати крок за кроком, тестуючи кожен етап і гарантуючи стабільність бази даних протягом всього циклу розробки та розгортання.

У данному випадку використовується метод `Schema::create`, що є частиною Laravel, для створення таблиці `products`. Цей метод дозволяє легко визначати структуру таблиці за допомогою функції зворотного виклику, що приймає об'єкт `Blueprint`. В Laravel, `Blueprint` — це клас, який використовується для визначення структури таблиць в міграціях бази даних, що надає методи для створення та налаштування стовпців, індексів, зв'язків та інших аспектів таблиці. `Blueprint` дозволяє описувати структуру таблиці у зручному та читабельному форматі. Він робить процес створення та модифікації таблиць в Laravel інтуїтивно зрозумілим та зручним, дозволяючи розробникам зосередитися на логіці додатку, а не на деталях реалізації бази даних.

На рис. 2.4 зображено створення таблиці до бази даних на основі мови програмування PHP:



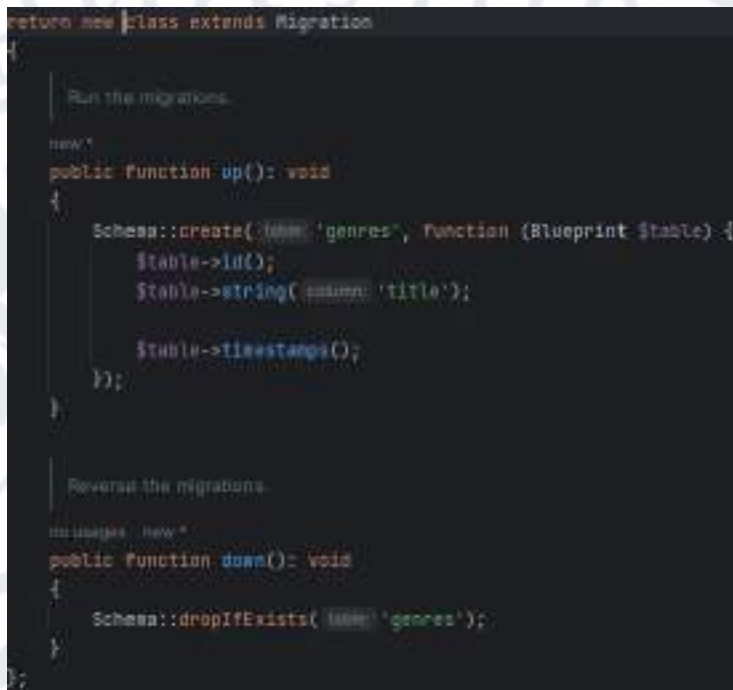
```
public function up() {
    Schema::create('products', function (Blueprint $table) {
        $table->id();
        $table->string('name', 'title');
        $table->integer('year');
        $table->string('img', 'wallpaper');
        $table->double('price');
        $table->integer('rate');
        $table->integer('sold');
        $table->unsignedInteger('quantity');
        $table->timestamps();
    });
}

// Reverse the migration
// example: -me-
public function down() {
}
```

Рисунок 2.4 - Лістинг створення міграції

У даній функції створюються стовпчики таблиці, яка міститиме всю інформацію про товар таку, як порядковий номер, назва, рік видання, обкладинка, ціна, кількість продажів та кількість наявного товару даного типу.

Далі створюється окремо таблиця для додавання жанрів та присвоюються їй кожному з них (рис. 2.5). Використовуючи фреймворк Laravel, цей код відповідає за створення міграції бази даних для таблиці, яка зберігатиме жанри для манг.



```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('genres', function (Blueprint $table) {
            $table->id();
            $table->string('title');

            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('genres');
    }
};
```

Рисунок 2.5 - Створення міграції для жанрів

Далі створюється міграція для створення таблиці `product_genres`, яка забезпечує зв'язок між продуктами та жанрами. Ця таблиця буде містити зовнішні ключі, що посилаються на таблиці `products` та `genres`, та автоматично зберігати час створення та оновлення записів.



```
public function up(): void
{
    Schema::create('product_genres', function (Blueprint $table) {
        $table->id();
        $table->foreignId('product_id')->references('id')->on('products');
        $table->foreignId('genre_id')->references('id')->on('genres');
        $table->timestamps();
    });
}
```

Рисунок 2.6 - Таблиця співвідношень жанрів

Ця міграція забезпечує можливість відстеження зв'язків між продуктами і жанрами, що зберігаються у відповідних таблицях `products` та `genres`, відповідно.

Таблиця товарів (рис. 2.7) складається з таких стовпців, як:

- ID – задає порядковий номер товару
- Title – містить назву товару
- Img – містить посилання на обкладинку
- Price – містить ціну товару
- Rate – оцінка товару від користувачів
- Sold – кількість проданих екземплярів
- Quantity – кількість наявних екземплярів
- Year – рік видання товару



ID	Title	Price	Rate	Sold	Quantity	Year
1	Book 1	10.00	4.5	100	100	2010
2	Book 2	15.00	4.2	80	80	2011
3	Book 3	12.00	4.8	120	120	2012
4	Book 4	18.00	4.1	60	60	2013
5	Book 5	9.00	4.6	150	150	2014
6	Book 6	11.00	4.3	90	90	2015
7	Book 7	14.00	4.4	70	70	2016
8	Book 8	16.00	4.0	50	50	2017
9	Book 9	13.00	4.7	110	110	2018
10	Book 10	17.00	4.2	65	65	2019

Рисунок 2.7 - Таблиця бази даних

2.4 Модель роботи інтернет-магазину

На рис. 2.8 зображена інформаційна модель роботи вебплатформи інтернет-магазину, у якій описані функціональні можливості для користувача та обмін і передача інформації, які відбуваються між користувачем, адміністратором та базою даних, в якій значна частина інформації, власне, і зберігається.

При вході на сайт, користувач проходить процедуру реєстрації, під час якої вводяться особисті дані. Ці дані надсилаються до бази даних і стають доступними для адміністратора.

Адміністратор має повний доступ до бази даних і може редагувати існуючі дані товарів, а також додавати нові товари до бази даних. Після додавання товарів вони стають доступними для перегляду як користувачам, так і самому адміністраторові.

Після авторизації на сайті користувач може переглядати список доступних товарів. Він має можливість додати обрані товари до кошика та оформити замовлення. Під час оформлення замовлення користувач надає дані для оплати та доставки.

Замовлення, оформлене користувачем, потребує підтвердження адміністратором перед виконанням. Після підтвердження адміністратором замовлення обробляється, і товари відправляються користувачеві.

Після отримання товару користувач може залишити відгук про придбаний товар. Однак цей відгук потребує підтвердження адміністратором перед тим, як він буде опублікований на сайті.

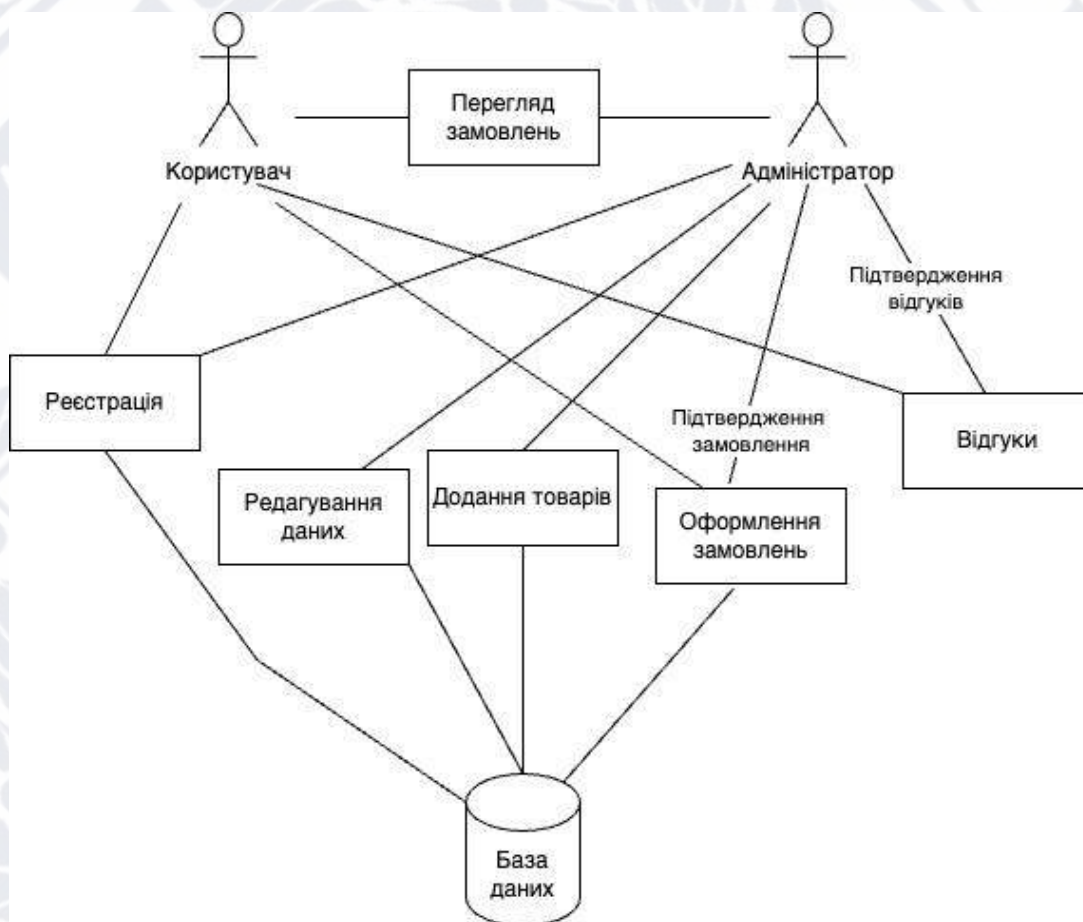


Рисунок 2.8 – Інформаційна модель роботи інтернет-магазину

РОЗДІЛ 3

РЕАЛІЗАЦІЯ РОЗРОБКИ ВЕБ-ПЛАТФОРМИ

3.1 Вибір технологій для розробки

Розробка сайту для інтернет-магазину вимагає вибору відповідних мов програмування та інструментів, які забезпечать ефективність, безпеку і функціональність системи. Існує ряд мов програмування та фреймворків, які широко використовуються для розробки веб-сайтів, зокрема інтернет-магазинів, такі як HTML, CSS, JavaScript, PHP та Python.

JavaScript є ключовою мовою для фронтенд розробки веб-додатків, включаючи інтернет-магазини. Його використання дозволяє створювати динамічні інтерфейси користувача та реагувати на дії користувача без перезавантаження сторінки. Багато фреймворків та бібліотек для JavaScript, таких як React, Vue.js та Angular, допомагають розробникам швидко створювати потужні веб-додатки. Наприклад, React, який розроблений компанією Facebook, забезпечує компонентну архітектуру та можливість повторного використання коду, що дозволяє ефективно організувати та розвивати складні інтерфейси. Vue.js, з іншого боку, пропонує простоту використання та невеликий розмір, що робить його чудовим вибором для початківців та проєктів з обмеженими ресурсами. Незалежно від вибору фреймворку, JavaScript забезпечує розширені можливості для створення інтерактивних інтерфейсів, які забезпечують зручне та приємне користувацьке досвід. Отже, JavaScript, як мова програмування, забезпечує динамічність та інтерактивність веб-сайту та використовується для створення взаємодії з користувачем, анімації, валідації форм та інших функціональних можливостей [27].

Python також є важливою мовою програмування для веб-розробки, особливо завдяки своїй простоті та читабельності коду. Використання Python, зокрема з фреймворками Django або Flask, може бути вигідним вибором для створення інтернет-магазину. Django надає повноцінний стек розробки, включаючи адміністративний панель, аутентифікацію, маршрутизацію та

безпеку, що дозволяє швидко створювати функціональний веб-додаток. Flask, з іншого боку, є легким та гнучким фреймворком, який дозволяє розробникам зосередитися на специфічних аспектах свого проєкту та швидко розгортати веб-додатки. Таким чином, Python є ще одним потужним інструментом для створення інтернет-магазинів, який варто врахувати при розгляді варіантів розробки вашого проєкту [23].

Наступним слід зазначити HTML. Хоч це і не мова програмування, але HTML є невід'ємною базою при розробці будь якого веб сайту. HTML - це основна мова розмітки веб-сторінок, що визначає структуру та зміст вмісту сторінки. У HTML використовуються теги, які вказують браузеру, як відображати кожен елемент на сторінці. Наприклад, тег `<h1>` визначає заголовок першого рівня, `<p>` - абзац тексту, `<ul>` та `<li>` - нумерований список та його елементи. Також, HTML дозволяє вбудовувати зображення, відео, аудіо та інші мультимедійні елементи у веб-сторінку [20].

CSS - це мова стилів, яка використовується для визначення зовнішнього вигляду веб-сторінок, включаючи кольори, шрифти, розміри, розташування елементів та інші стилістичні аспекти. CSS дозволяє розробникам змінювати вигляд інтерфейсу, щоб зробити його більш привабливим та зручним для користувачів. Наприклад, за допомогою CSS можна задати колір тла сторінки, ширину та висоту блоків, відступи між елементами, анімаційні ефекти та багато іншого.

Разом HTML та CSS утворюють основну складову веб-дизайну. HTML відповідає за структуру та зміст сторінки, в той час як CSS визначає її зовнішній вигляд та стилістику. Ці дві мови доповнюють одна одну, дозволяючи розробникам створювати якісний та естетично виглядаючий веб-контент.

Щодо серверної частини, PHP є однією з найпопулярніших мов програмування для веб-розробки. Використання PHP в поєднанні з фреймворком Laravel дозволяє ефективно створювати інтернет-магазини з різноманітною функціональністю, такою як керування товарами, користувачами, замовленнями та іншими аспектами бізнесу. Laravel забезпечує структурованість, безпеку та

простоту розробки завдяки своїм вбудованим інструментам та готовим компонентам.

По-перше, PHP має широке поширення і підтримується величезною спільнотою розробників, що означає, що для будь-якої проблеми чи питання завжди можна знайти рішення в документації. Багато популярних веб-сайтів і систем управління контентом, такі як WordPress, використовують PHP, підтверджуючи його надійність і ефективність.

По-друге, ця мова вирізняється своєю простотою у вивченні та використанні. Його синтаксис досить легкий у розумінні навіть для початківців, що дозволяє швидко розпочати розробку веб-додатку. Зокрема, PHP був створений спеціально для веб-розробки, тому він має вбудовані функції для роботи з HTTP-запитами, сесіями, базами даних і багато чого іншого, що спрощує створення динамічних веб-сторінок.

Важливо зазначити, що PHP має широку екосистему фреймворків, таких як Laravel, Symfony та CodeIgniter, які надають структуру для коду, спрощують розробку та допомагають дотримуватись найкращих практик. Ці фреймворки дозволяють розробникам швидко створювати масштабовані та підтримувані додатки.

Окремо слід розглянути такий PHP фреймворк, як Laravel. Laravel - це відкритий фреймворк для веб-розробки, написаний на мові програмування PHP. Він став одним з найпопулярніших інструментів для створення сучасних веб-додатків завдяки своїй потужності, гнучкості та простоті використання. Фреймворк пропонує широкий спектр функцій та інструментів, які спрощують процес розробки, забезпечуючи при цьому високий рівень безпеки та ефективності [34].

Однією з ключових особливостей Laravel є його елегантний та зрозумілий синтаксис, який дозволяє розробникам швидко створювати функціональний код без необґрунтованої складності. Фреймворк пропонує багато готових компонентів, які дозволяють виконувати такі завдання, як маршрутизація, аутентифікація, кешування та робота з базою даних без зайвого зусилля.

Крім того, Laravel забезпечує високий рівень безпеки, включаючи вбудовані засоби захисту від таких загроз, як вразливості SQL-ін'єкцій. SQL-ін'єкція - це вид атаки на базу даних, яка полягає у використанні зловмисного SQL-коду для недозвеного взаємодії з базою даних через веб-додаток. Зловмисник вставляє SQL-код у введені користувачем дані, такі як поля форм або параметри URL-адреси, з метою отримання неправомірного доступу до даних або зміни їх.

SQL-ін'єкції можуть бути використані для викрадення чутливої інформації, такої як паролі, кредитні картки або особисті дані, а також для виконання різних дій в базі даних, таких як видалення, оновлення або додавання даних. Вони можуть бути особливо небезпечними в тих випадках, коли веб-додаток не належним чином валідує введені дані або не застосовує параметризовані запити до бази даних.

Також Laravel забезпечує захист від CSRF-атак. CSRF - це тип атаки на веб-додатки, при якій зловмисник використовує авторизовану сесію користувача для виконання небажаних дій на веб-сайті без його відома або згоди. Атака полягає в тому, що зловмисник намагається перехопити або підробити запити від автентифікованого користувача, які надсилаються на цільовий веб-сайт.

Для здійснення атаки CSRF, зловмисник може використовувати різні механізми, такі як відправка спеціально підготовлених посилань або форм через електронні листи, соціальні мережі або інші канали комунікації. Якщо жертва, яка має активну сесію на цільовому веб-сайті, відкриє посилання або відправить форму, атака може бути успішною, і зловмисник може виконати небажані дії в ім'я цієї жертви.

Це робить фреймворк ідеальним вибором для розробки надійних та безпечних веб-додатків будь-якого рівня складності.

Завдяки активній спільноті розробників і постійному оновленню, Laravel залишається одним з найбільш перспективних та сучасних інструментів для веб-розробки. Він пропонує розширені можливості, готові рішення та зручні

інструменти, які дозволяють розробникам швидко створювати потужні та ефективні веб-додатки.

Також слід згадати про Tailwind CSS. Tailwind CSS - це сучасний, високопродуктивний фреймворк CSS, який дозволяє швидко та ефективно розробляти стильові компоненти для веб-додатків. Його особливість полягає в тому, що він пропонує набір готових класів, які описують стилізацію елементів, замість традиційного підходу з визначенням стилів в CSS-файлах.[14]

У Tailwind CSS кожен клас представляє собою конкретний стилевий атрибут, наприклад, розмір тексту, колір фону, відступи, рамки та інші. Це дозволяє розробникам швидко та просто застосовувати стилі до елементів, не потребуючи написання власного CSS-коду або розширення стилів для кожного елемента окремо.

Крім того, Tailwind CSS пропонує концепцію атомарного дизайну, де кожен стилевий клас є незалежним та має мінімальний вплив на інші елементи веб-сторінки. Це сприяє швидкості розробки та підтримці коду, а також забезпечує консистентність у вигляді та стилізації елементів.

Завдяки своїй простоті використання та широким можливостям налаштування, Tailwind CSS став популярним інструментом для розробки сучасних веб-додатків, який використовується багатьма веб-розробниками по всьому світу.

Отже, веб-платформа буде розроблятися на основі мови програмування PHP та фреймворку Laravel. Для створення структури веб-сторінки було обрано мову розмітки HTML, яка дозволяє створювати різні елементи на сторінці, такі як абзаци, заголовки, форми тощо.

Також буде використано CSS, що є мовою стилів, яка дозволяє налаштувати зовнішній вигляд елементів сторінки та задати їх бажані параметри типу кольору, розміру, рамки тощо.

Для швидкого і ефективного створення веб-інтерфесу буде застосовано фреймворк Tailwind CSS, який містить в собі авторські класи з стилями.

Застосовуватиметься вбудований шаблонний двигун Blade для PHP-фреймворка Laravel. Blade - це вбудований шаблонний мова для фреймворка Laravel, яка дозволяє легко та ефективно вбудовувати PHP-код безпосередньо в HTML-шаблони. Blade надає зручний та зрозумілий синтаксис для виконання різноманітних завдань, таких як відображення даних з бази даних, управління умовами та циклами, включення підшаблонів та інше.

Однією з основних переваг Blade є можливість використання директив, які спрощують роботу з HTML-формами, маршрутами, перекладами та іншими аспектами веб-розробки. Наприклад, директива `@csrf` генерує приховане поле CSRF для захисту від атак CSRF, а директива `@foreach` дозволяє зручно перебирати масиви та колекції даних.

Крім того, Blade підтримує наслідування шаблонів та включення різних секцій з інших файлів, що дозволяє створювати легкі та зрозумілі HTML-шаблони для веб-сторінок. Благодаря всім цим особливостям, Blade став потужним інструментом для розробки динамічних та масштабованих веб-додатків з використанням фреймворка Laravel.

Для бази даних використана PostgreSQL, яка є потужною системою управління реляційними базами даних (СУБД). PostgreSQL є відкритою, багатофункціональною, об'єктно-реляційною базою даних, яка користується великою популярністю серед розробників завдяки своїм надійним можливостям і розширюваності.

PostgreSQL підтримує стандарт SQL та має багато розширених функцій, таких як складні запити, тригери, функції та зовнішні ключі. Вона також надає підтримку для збереження великих обсягів даних і забезпечує високу продуктивність навіть у великих системах. PostgreSQL дозволяє легко управляти складними операціями з даними та підтримує різноманітні типи даних, включаючи JSON, що робить її гнучким вибором для сучасних веб-додатків.[36]

Ще однією перевагою PostgreSQL є її розширюваність. Розробники можуть створювати власні функції, типи даних, операції та індекси, що дозволяє адаптувати базу даних під специфічні потреби проекту. PostgreSQL також

забезпечує високу безпеку даних завдяки багаторівневій системі контролю доступу і підтримці шифрування.

Завдяки своїй надійності, продуктивності та багатофункціональності, PostgreSQL є ідеальним вибором для створення баз даних у веб-платформах. Вона добре інтегрується з Laravel, що дозволяє легко управляти базами даних, виконувати міграції, і здійснювати складні операції з даними через ORM (Object-Relational Mapping) систему Eloquent.

3.2 Створення сторінки авторизації та реєстрації

Перш за все, потрібно створити систему реєстрації та авторизації користувачів. У представленому коді (рис. 3.1). визначаються маршрути для реєстрації нових користувачів, аутентифікації, а також відновлення паролю. Всі ці маршрути групуються під загальним середовищем guest, яке вказує, що до них можуть звертатися тільки неавторизовані користувачі.

Для реєстрації нових користувачів визначаються два маршрути: один для відображення форми реєстрації, а інший для обробки поданих даних цієї форми та створення нового користувача. Аналогічно, для входу в систему також існують два маршрути: один для відображення форми входу, а інший для обробки даних і аутентифікації користувача.

```
Route::middleware('guest')->group(function () {
    Route::get('/register', [RegisteredUserController::class, 'create'])
        ->name('register');

    Route::post('/register', [RegisteredUserController::class, 'store']);

    Route::get('/login', [AuthenticatedSessionController::class, 'create'])
        ->name('login');

    Route::post('/login', [AuthenticatedSessionController::class, 'store']);

    Route::get('/forgot-password', [PasswordResetLinkController::class, 'create'])
        ->name('password.request');

    Route::post('/forgot-password', [PasswordResetLinkController::class, 'store'])
        ->name('password.email');

    Route::get('/reset-password/{token}', [NewPasswordController::class, 'create'])
        ->name('password.reset');

    Route::post('/reset-password', [NewPasswordController::class, 'store'])
        ->name('password.store');
});
```

Рисунок 3.1 - Створення маршрутів

Крім того, передбачені маршрути для відновлення паролю. Перший маршрут дозволяє користувачам запитувати скидання паролю, відображаючи форму для введення електронної пошти. Після цього, за допомогою другого маршруту, система відправляє електронний лист із посиланням для скидання паролю. Коли користувач переходить за цим посиланням, третій маршрут показує форму для введення нового паролю. Останній маршрут обробляє введені дані та оновлює пароль користувача.

Таким чином, ці маршрути забезпечують повну функціональність для управління реєстрацією, авторизацією та відновленням паролів у системі, створюючи зручний і безпечний процес для користувачів.

За допомогою розширення до CSS під назвою Tailwind та вбудованого у Laravel шаблонного механізму Blade створюється front-end складова веб-платформи. Tailwind CSS дозволяє ефективно стилізувати інтерфейси завдяки зручним утилітам класів, які можна легко застосовувати безпосередньо у HTML. Це прискорює розробку і зменшує необхідність написання власних CSS-правил.

Шаблонний механізм Blade, вбудований у Laravel, забезпечує зручний спосіб створення динамічних веб-сторінок. Він дозволяє використовувати шаблонні файли з розширенням `.blade.php`, які можуть містити як HTML-код, так і вбудовані вирази PHP для рендерингу даних. Blade також підтримує компоненти, включення інших шаблонів, цикли, умовні вирази та секції для зручного і структурованого коду.

На рис. 3.2 зображено остаточний варіант сторінки авторизації:



Рисунок 3.2 – Сторінка авторизації

На рис. 3.3 зображена сторінка реєстрації.

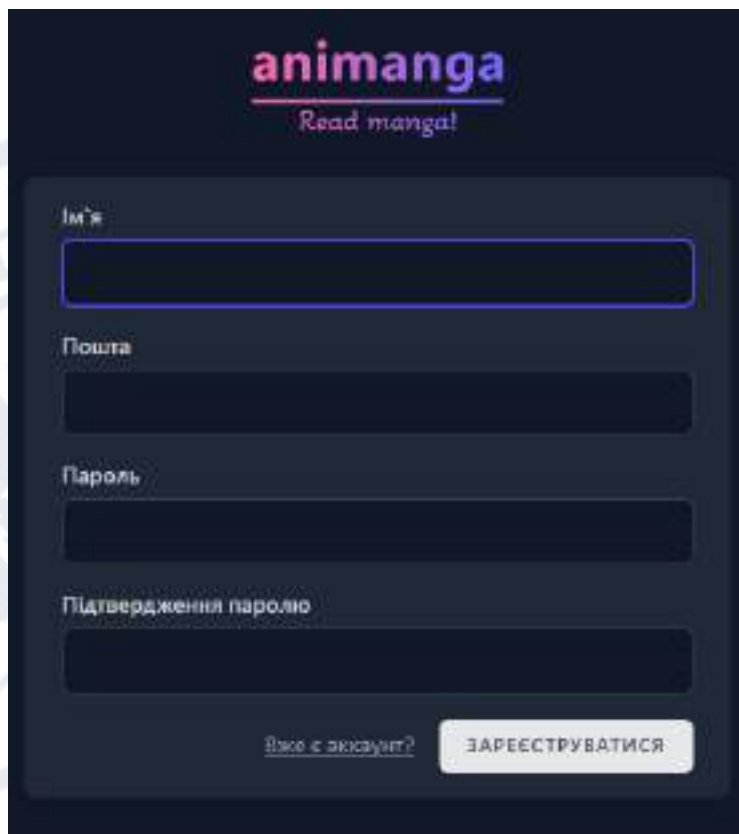


Рисунок 3.3 - Сторінка реєстрації

Користувач вводить своє ім'я, пошту та задає пароль для свого акаунту, після чого повторює заданий пароль, щоб запевнитись у його правильному написанні.

3.3 Створення головної сторінки

Обличчям сайту та тим, що формує перше враження, є головна сторінка. Вона повинна бути привабливою, інформативною та зручною для користувачів. Головна сторінка має відображати основні категорії товарів, такі як манга та тематичний мерч, щоб відвідувач одразу міг зорієнтуватися у асортименті. Важливо, щоб дизайн був сучасним і візуально привабливим, в решті решт читабельним.

Отже, при візиті на сайт користувач потрапляє на головну сторінку (рис. 3.4). Всі товари чітко розділені на дві основні категорії: манга та тематичний мерч, користувач одразу може ознайомитися з коротким описом обох категорій,

представлених у магазині. Це дозволяє користувачеві швидко орієнтуватися на сайті та знайти те, що його цікавить.

На головній сторінці також повинні бути представлені новинки, популярні товари та акції, щоб привернути увагу користувачів і заохотити їх до покупок. Інтуїтивно зрозуміла навігація, пошукова стрічка та легкий доступ до кошика, уподобаних товарів і профілю користувача – всі ці елементи сприяють позитивному користувацькому досвіду та ефективній роботі інтернет-магазину.



Рисунок 3.4 – Головна сторінка

Після короткого опису кожної з категорій наявне посилання "Перейти до перегляду товару", яке перекидає користувача на сторінку зі списком товарів відповідної категорії. Це посилання забезпечує зручний перехід до більш

детального перегляду товарів, що дозволяє користувачеві швидко знайти саме те, що його цікавить.

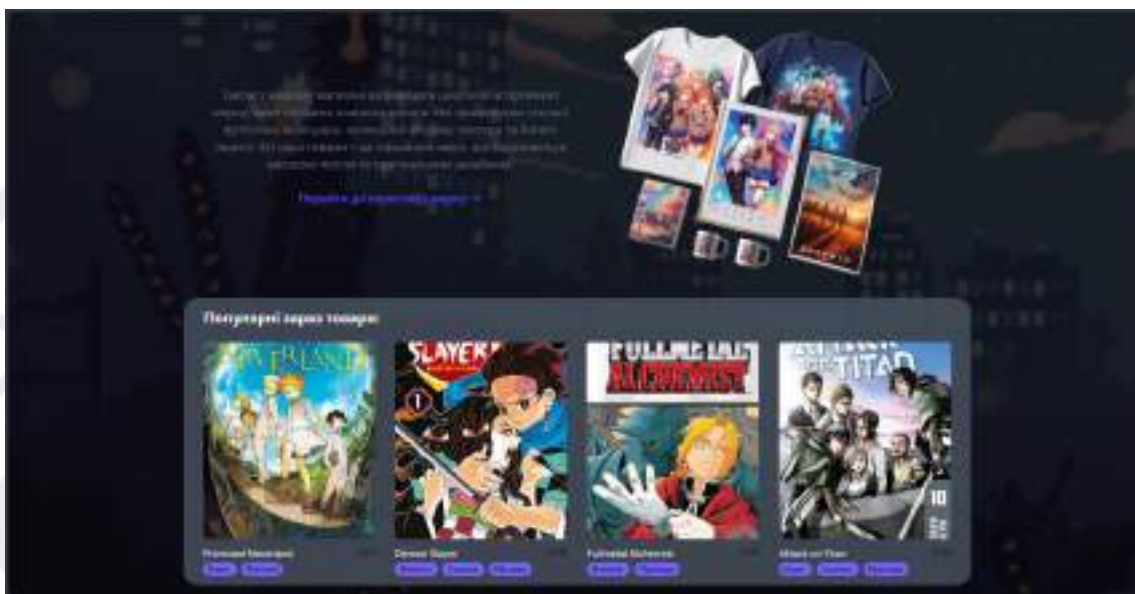


Рисунок 3.5 – Список запропонованих товарів на головній сторінці

Трішки прогорнувши вниз, користувач бачить список запропонованих популярних товарів (рис. 3.5), які могли б зацікавити його. Цей розділ спеціально розроблений для того, щоб привернути увагу до найпопулярніших і найцікавіших товарів, що можуть відповідати вподобанням користувача. Кожен товар у цьому списку супроводжується обкладинкою товару, назвою, ціною, та жанрами, до яких належить кожен товар, що робить процес вибору швидким та зручним.

Також на сайті представлений унікальний дизайн з анімованими фонами, які задають певну атмосферу для користувача. Ці анімовані фони не лише додають візуальної привабливості, але й створюють відповідний настрій, що занурює відвідувачів у світ манги та аніме. Такий підхід допомагає зробити перебування на сайті більш приємним та інтерактивним, що сприяє позитивному користувацькому досвіду та утриманню уваги клієнтів.

3.4 Розробка сторінки перегляду товарів

Сторінка зі списком товарів має виглядати таким чином, щоб забезпечити користувачу зручний доступ до всієї необхідної інформації про товари та

можливість швидко знайти потрібний продукт. Основні елементи, які присутні на сторінці, включають назви товарів, ціни, зображення (рис. 3.6 та рис. 3.7).

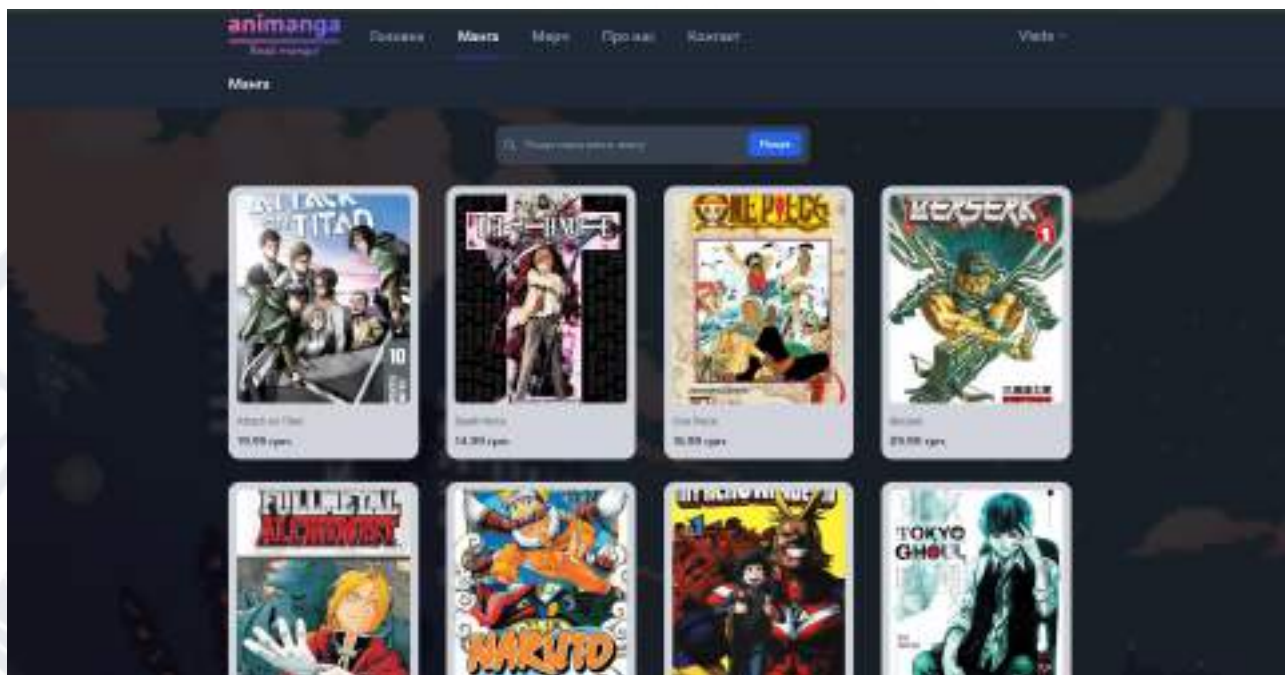


Рисунок 3.6 - Сторінка перегляду товарів категорії «Манга»

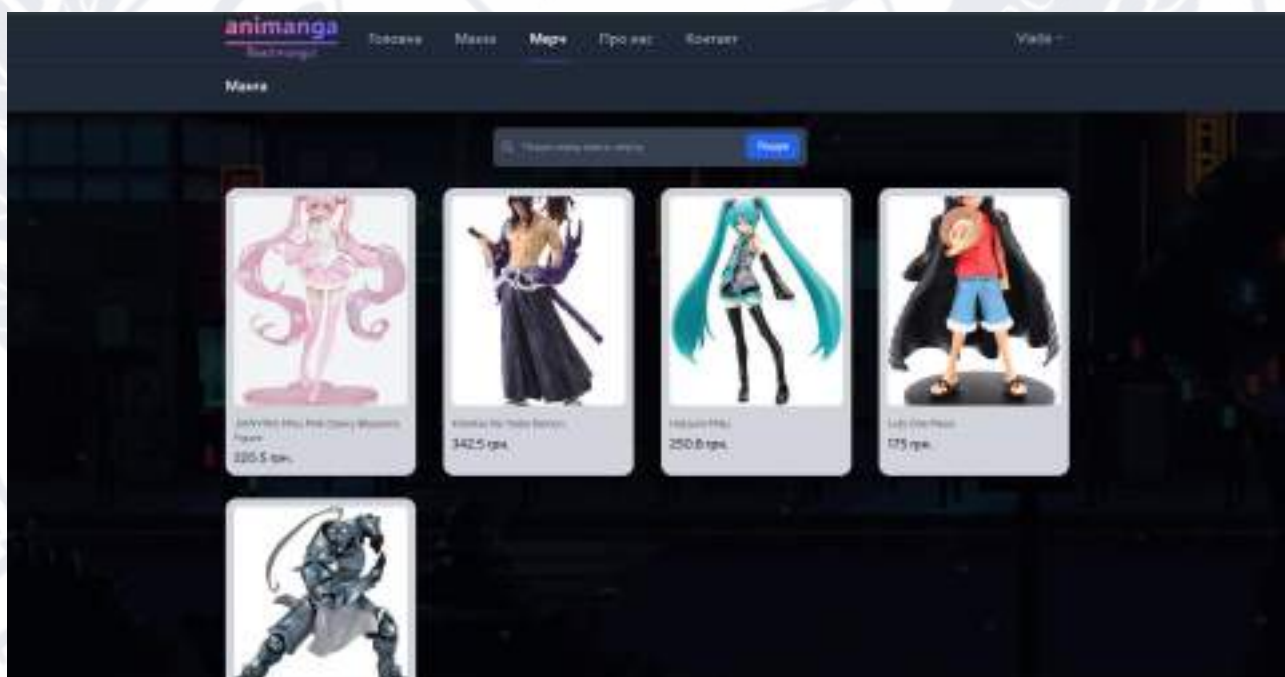


Рисунок 3.7 - Сторінка перегляду товарів категорії «Мерч»

У верхній частині сторінки розташоване головне меню, яке містить кнопки з посиланнями на різні розділи сайту: "Головна", "Про магазин" і "Контакти". Додатково, у правому верхньому кутку розміщене контекстне меню з інструментами для користувача, включаючи доступ до особистого профілю, кошика з покупками та кнопки для виходу з облікового запису.

Сторінка з окремим товаром призначена для детального представлення конкретного продукту, надання користувачеві всієї необхідної інформації та забезпечення можливості здійснити покупку. Основні елементи, які мають бути присутні на цій сторінці, включають зображення товару, назву, опис, ціну, автора, вибір кількості екземплярів, кнопку для додавання до кошика та можливість залишити відгук.

На початку сторінки розташоване велике зображення товару, що дозволяє детально роздивитися продукт. Поруч із зображенням знаходиться назва товару, його оцінка, перелік жанрів, рік видання, автор, перекладач та видавництво.

Трохи нижче наявна функція вибору кількості екземплярів та кнопка, що додає товар до кошику.

На рис. 3.8 можна побачити зовнішній вигляд даної сторінки з окремим товаром:

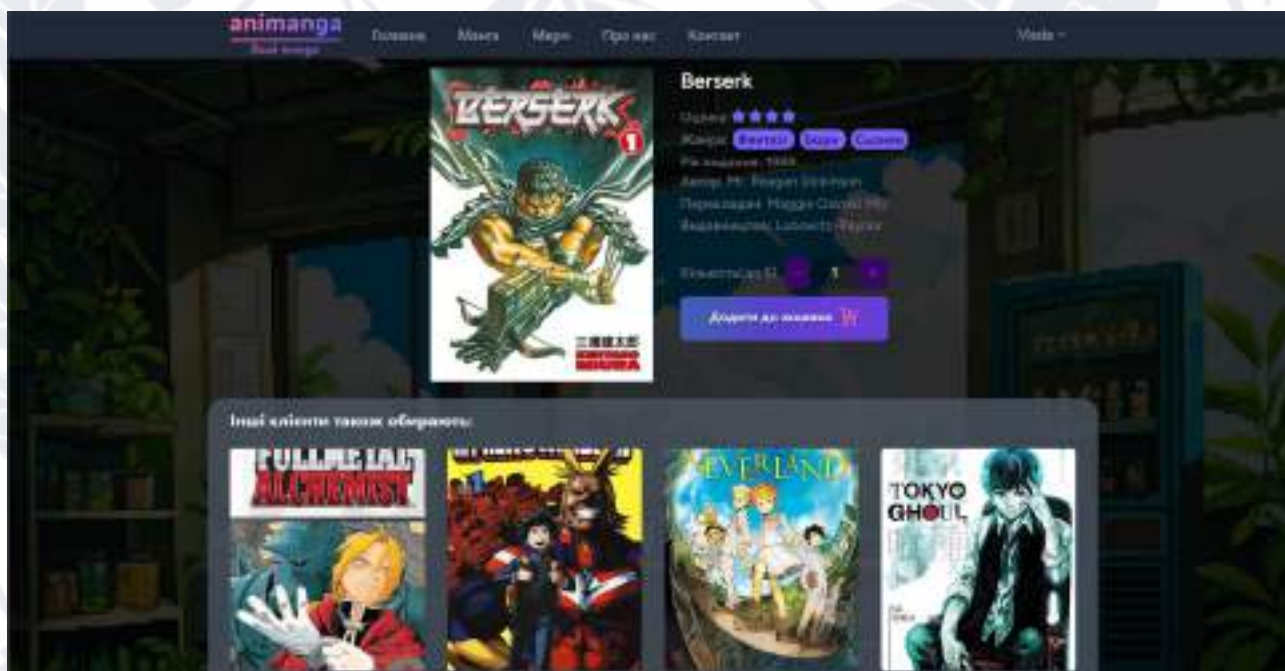


Рисунок 3.8 - Сторінка перегляду окремого товару

Під товаром пропонується перелік схожих товарів, які користуються попитом серед інших покупців. Це допомагає користувачу знайти популярні та перевірені продукти.

3.5 Створення кошика

Одним із ключових елементів будь-якого онлайн-магазину є сторінка з кошиком покупок. Ця сторінка відіграє вирішальну роль у процесі покупки, забезпечуючи зручний і зрозумілий спосіб перегляду та управління обраними товарами перед оформленням замовлення.

Сторінка з кошиком містить всі додані користувачем товари і висвітлює загальну суму обраних товарів (рис. 3.9).

Біля кожного доданого товару вказана ціна за один екземпляр, кількість доданих екземплярів. Якщо користувач помилково додав товар або змінив свій вибір, він може прибрати обраний продукт за допомогою кнопки «видалити з кошика».

У правій частині екрану відображається інформація про загальну кількість речей та тип обраної доставки. Також користувач може перевірити загальну суму товарів у кошику і швидко перейти до оплати.

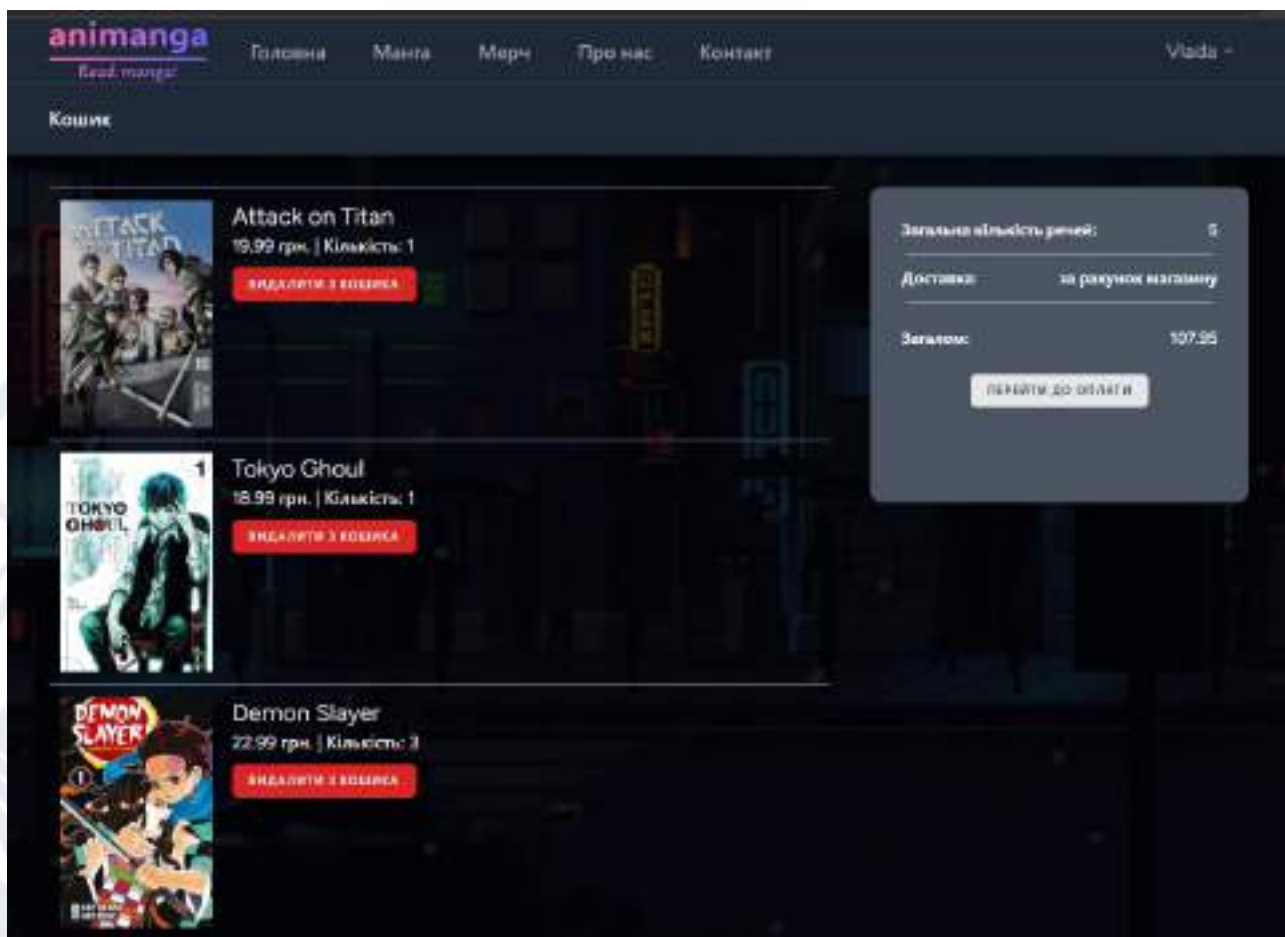


Рисунок 3.9 – Сторінка кошику

3.6 Тестування роботи розробленого продукту

Тестування веб-додатку перед його запуском є критично важливим етапом у розробці, який не можна ігнорувати. Воно забезпечує виявлення та усунення помилок, що можуть негативно вплинути на функціональність, безпеку та загальне враження користувачів від продукту. Завдяки ретельному тестуванню можна впевнитися, що додаток працює стабільно та відповідає всім вимогам і очікуванням.

Тестування веб-додатку перед запуском є невід'ємною частиною успішної розробки, яка допомагає забезпечити високу якість, безпеку та задоволеність користувачів. Тестування допомагає знайти та виправити технічні помилки, які можуть виникнути під час розробки. Це включає баги в коді, проблеми з сумісністю браузерів, помилки у верстці тощо. Окрім цього, за допомогою тестування можна перевірити, наскільки зручний та інтуїтивно зрозумілий

інтерфейс веб-додатку. Це дозволяє виявити потенційні проблеми з навігацією або дизайном, які можуть заважати користувачам.

В першу чергу слід перевірити коректність форми авторизації. При введенні невірного логіну або паролю користувачу має виводити повідомлення про помилку, щоб він розумів у чому проблема, яка не дає доступу до його акаунту.

При введенні невірного логіну (або паролю) виводиться повідомлення червоним шрифтом (рис. 3.10), яке повідомляє користувачу про помилку у введених ним даних.



Рисунок 3.10 – Перевірка роботи форми авторизації

Не менш важливо перевірити роботу форми реєстрації (рис. 3.11). Коли користувач вводить електронну пошту, яка вже прив'язана до іншого акаунту, з'являється повідомлення: «Цей e-mail вже використовується». Це забезпечує уникнення дублювання акаунтів і збереження унікальності кожного користувача.

Коли користувач вводить різні паролі або допускає помилку, з'являється повідомлення про те, що паролі не збігаються. Це допомагає користувачеві зрозуміти, що потрібно виправити, щоб успішно завершити реєстрацію.

Для забезпечення безпеки користувача, пароль має містити не менше 8 символів. Якщо користувач задає занадто короткий пароль, сайт не дозволяє

продовжити реєстрацію і попереджає про цю умову. Таким чином, форма реєстрації забезпечує безпеку і надійність акаунтів користувачів, допомагаючи уникати помилок під час введення даних.

Рисунок 3.11 – Перевірка форми реєстрації

Користувач має також змогу змінювати задану раніше інформацію про себе і свою пошту (рис. 3.12). Зміна інформації про себе на сайті зазвичай включає кілька кроків, включаючи взаємодію з базою даних. Скрипт перевіряє та обробляє дані для подальшого збереження у базі даних. Наприклад, перевірка на введення коректних даних (наприклад, перевірка на правильний формат електронної пошти або пароля), валідація та очищення даних перед збереженням.

Після обробки даних формується запит до бази даних для оновлення відповідних записів. Наприклад, у випадку редагування інформації про користувача, скрипт може виконати запит до бази даних для оновлення відповідного рядка таблиці з новими даними.

База даних виконує запит, оновлює відповідні записи з новими даними. Після успішного оновлення бази даних користувачу повідомляється про успішне збереження змін. У випадку помилки, користувачу надається відповідне повідомлення про помилку та інструкції щодо подальших дій.

Зовнішній вигляд форми для зміни інформації про користувача зображений на рис. 3.12.

Рисунок 3.12 – Оновлення інформації про користувача

Функція зміни паролю (рис. 3.13) теж є доволі важливою для безпеки. Перш ніж дозволити користувачеві змінювати пароль, система повинна перевірити, чи він дійсно є автентифікованим користувачем. Це включає введення поточного пароля або перевірку наявності сесії авторизації.

Користувач повинен ввести новий пароль. Перед зміною пароля система встановлює деякі обмеження на пароль, вимагається як мінімум 8 символів та великі літери.

Після того як новий пароль був перевірений та захищений, він оновлюється в базі даних. Це відбувається за допомогою запиту до бази даних, який оновлює запис користувача з новим захешованим паролем.

Після успішної зміни паролю користувачу може бути надіслане повідомлення про це, підтвердження або підтвердження через електронну пошту або повідомлення в його особистому кабінеті.

У випадку помилки у полі “Нинішній пароль” висвітлюється повідомлення, яке повідомляє про некоректність введеного паролю. Далі скрипт перевіряє чи збігається новий пароль в двох полях для введення. У випадку неспівпадіння нового паролю сайт так само повідомить про це користувача.

Рисунок 3.13 – Зміна та відновлення паролю

У користувача також є можливість видалення акаунту, якщо він більше не бажає його мати (рис. 3.14). В описі користувача попереджається про те, що після видалення акаунту усі його дані буде видалено. Після натискання користувачем на кнопку «Видалити акаунт» система розпочинає процес видалення акаунта. Це включає видалення всієї особистої інформації користувача з бази даних, видалення асоційованих записів або даних, а також видалення всіх сесій та кешованих даних.

Рисунок 1.14 – Функція видалення акаунту

З метою безпеки та аудиту система зберігає запис про видалення акаунта разом із датою та часом видалення, а також іншу відповідну інформацію. Це може допомогти вирішити потенційні спірні питання щодо видалення акаунта у майбутньому.

Перевірка коректної роботи фільтру жанрів та строки пошуку товарів є важливим етапом тестування інтернет-магазину. Жанри (рис. 3.15), які вказані в описі кожного товару, є клікабельними, щоб користувач міг швидко та зручно переглянути інші товари з таким же жанром.

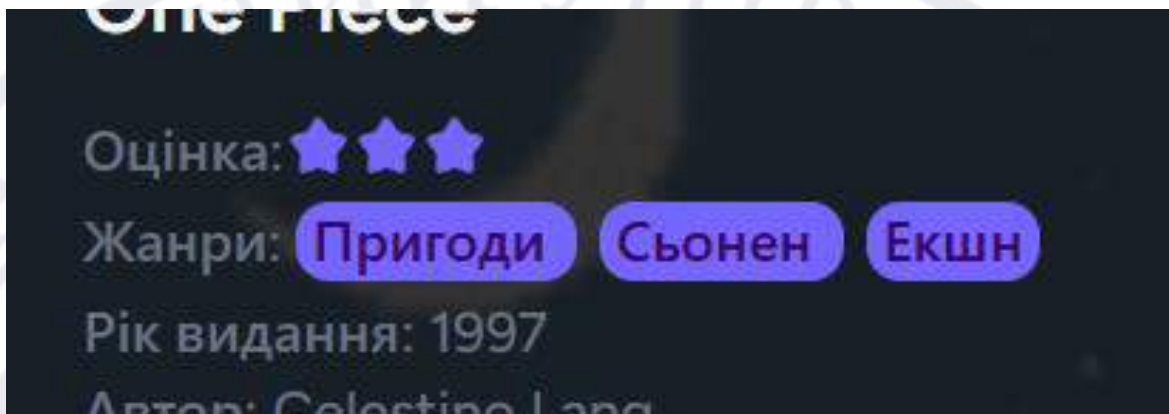


Рисунок 3.15 – Перелік жанрів

Після натискання на жанр користувач переадресований на відповідну сторінку (рис. 3.16), де відображається список усіх товарів з обраним жанром. Це дозволить клієнтам з легкістю знайти товари, які вони шукають, і забезпечить їм приємний досвід користування магазином.

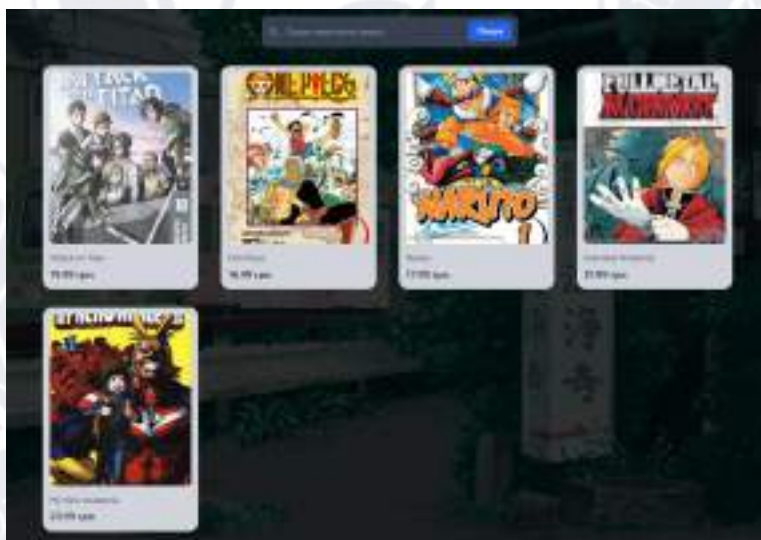


Рисунок 3.16 – Сторінка з відфільтрованими за жанром товарами

Такий функціонал не лише полегшує навігацію користувачів по сайту, але й збільшує ймовірність покупки, оскільки дозволяє швидко знайти товари, які відповідають їхнім інтересам та уподобанням. Тому важливо перевірити, що

фільтр жанрів та строки пошуку працюють бездоганно і забезпечують зручний та ефективний пошук товарів для користувачів.

Стрічка пошуку на сайті є важливим інструментом, який забезпечує користувачам можливість швидко знаходити необхідний контент. Коли користувач вводить ключові слова або фразу в поле пошуку, ця інформація передається на сервер для обробки. Сервер отримує запит і розпочинає його обробку, яка може включати попереднє очищення даних від зайвих символів і розбиття запиту на окремі ключові слова.

Після цього сервер виконує пошук відповідей у базі даних. Для цього використовуються різні алгоритми, такі як повнотекстовий пошук, пошук за ключовими словами та пошук за індексами. Важливо, щоб база даних була добре організована та індексована для швидкої обробки запитів.

Коли сервер знаходить відповідні результати, він сортує їх за релевантністю. Це може враховувати такі фактори, як частота використання ключових слів, популярність товару або дата додавання. Після сортування найрелевантніші результати виводяться на екран у вигляді списку або таблиці з посиланнями, кратким описом товарів, зображеннями та цінами.

Користувач може обрати один з результатів пошуку, натиснувши на посилання або кнопку, щоб перейти до сторінки з детальною інформацією про товар чи послугу. Якщо необхідно, користувач може змінити запит і повторити пошук для отримання додаткової інформації або інших результатів.

Таким чином, стрічка пошуку забезпечує ефективну взаємодію користувача з сайтом, полегшуючи процес знаходження потрібного контенту і підвищуючи загальну зручність користування сайтом.

Наприклад, нехай у стрічку пошуку буде введено одне ключове слово з назви бажаного товару. Після натискання кнопки «Пошук» користувачу висвітлюється список товарів, у назві яких наявне введене в пошукову строку слово.

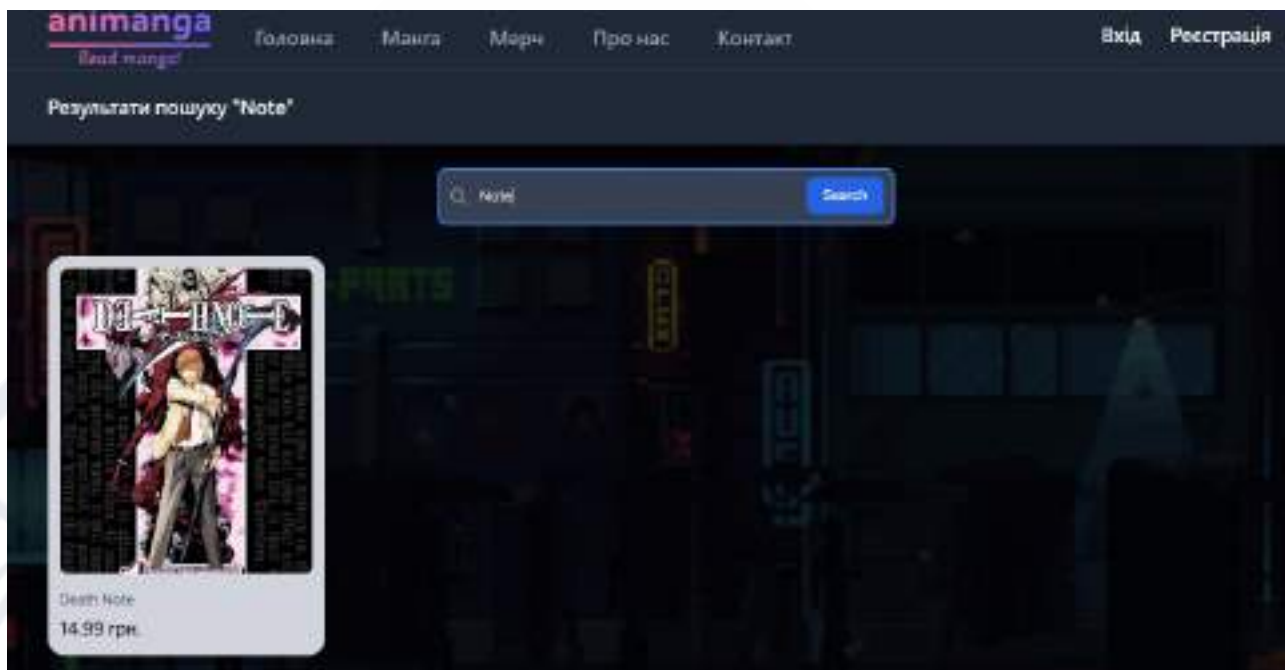


Рисунок 3.16 – Пошук товарів

Тепер слід перевірити, як працюватиме сайт для неавторизованого користувача. Це важливо для забезпечення доступності та функціональності ресурсу для всіх відвідувачів.

Всі основні функції сайту, такі як перегляд товарів, фільтрація за жанрами та використання стрічки пошуку, доступні без обов'язкової авторизації (можна переконатись переглянувши рис. 3.16). Неавторизований користувач має можливість ознайомитися з асортиментом товарів, прочитати описи та переглянути зображення.

Далі слід перевірити, як обробляються спроби неавторизованих користувачів здійснити дії, що вимагають авторизації, такі як додавання товарів до кошика, оформлення замовлення або написання відгуків. Сайт має коректно перенаправляти таких користувачів на сторінку входу або реєстрації, зберігаючи при цьому контекст запиту, щоб після успішної авторизації вони могли продовжити розпочату дію.

Також важливо перевірити повідомлення про помилки та підказки, які отримує неавторизований користувач під час взаємодії з сайтом. Вони повинні бути зрозумілими та допомагати користувачеві виконати необхідні дії для завершення процесу.

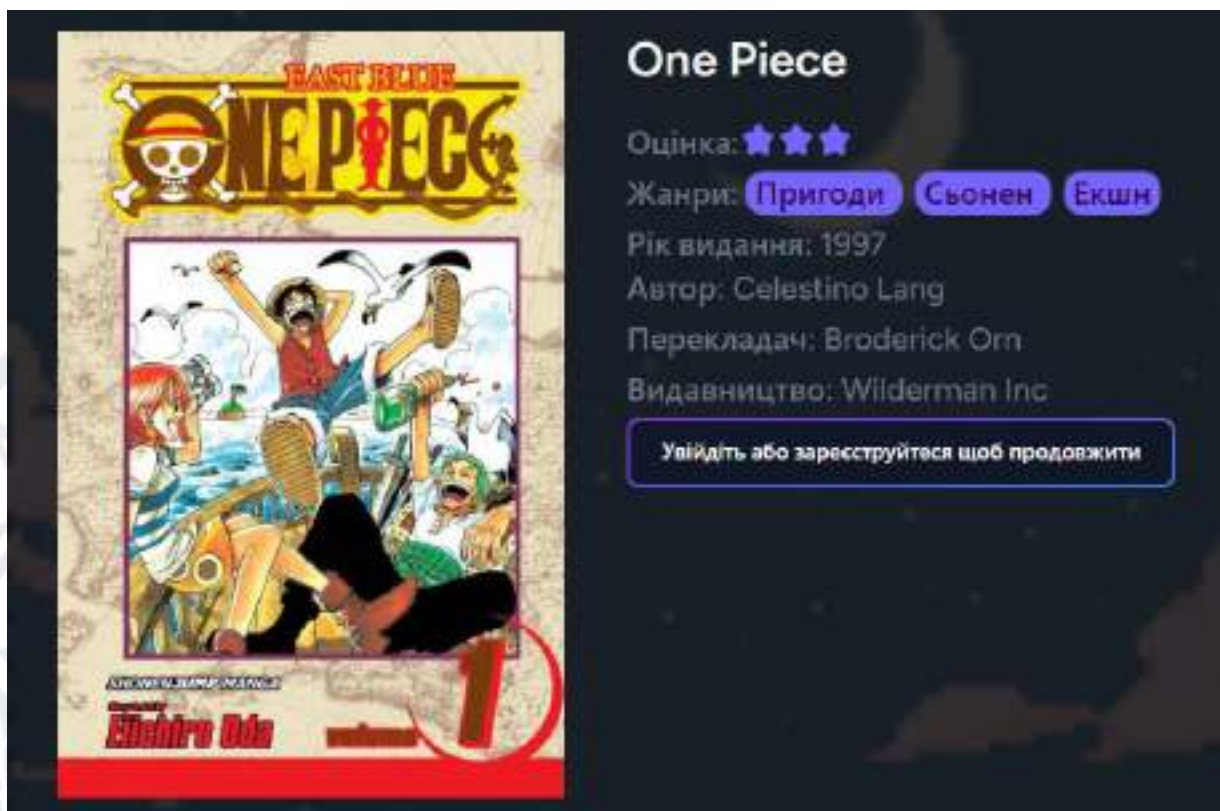


Рисунок 3.17 – Перевірка доступу до функції додавання до кошику товару для неавторизованого користувача

Неавторизованим користувачам замість кнопки «Додати до кошика» висвітлюється повідомлення «Увійдіть або зареєструйтеся щоб продовжити», натиснувши на яке, користувача перекине на сторінку авторизації. Також з будь-якої сторінки сайту доступні кнопки «Вхід» та «Реєстрація» (рис. 3.18) у правому верхньому кутку веб-сторінки, що дозволить швидко та зручно авторизуватись.

Вхід **Реєстрація**

Рисунок 3.18 – Швидко доступні функції «Вхід» та «Реєстрація»

Таким чином, забезпечення належної роботи сайту для неавторизованих користувачів є критичним кроком у забезпеченні доступності та зручності використання інтернет-магазину. Неавторизовані користувачі повинні мати можливість легко переглядати каталог товарів, використовувати функції пошуку та фільтрації, а також отримувати повну інформацію про товари без необхідності створювати акаунт. Це підвищує шанси на те, що відвідувачі сайту стануть постійними клієнтами, оскільки вони зможуть ознайомитися з асортиментом і

якістю обслуговування ще до реєстрації. Однак, дії, які вимагають авторизації, такі як додавання товарів до кошика, оформлення замовлення чи залишення відгуків, повинні бути чітко позначені, і користувачів слід ввічливо спрямовувати до форми входу або реєстрації для продовження цих дій. Це забезпечує баланс між зручністю для користувача та безпекою даних, що є ключовими аспектами успішної роботи інтернет-магазину.



ВИСНОВКИ

Метою проєкту є дослідження методів розробки вебплатформи інтернет-магазину, який дозволить продавати товари анімаційної тематики, який матиме інтуїтивно зрозумілий UI дизайн та відповідатиме усім функціональним вимогам. Було проведено дослідження аналогічних продуктів, задля поставлення стандартів до створеного проєкту та досягнення кращого результату, щоб мати змогу конкурувати з сайтами-аналогами.

За допомогою літературних джерел було досліджено методи та інструменти реалізації розробки вебплатформи і встановлено вимоги, які ставляться до розробленого продукту. Було задано і виконано усі поставлені на початку задачі, а саме:

- Обрано та проаналізовано інструменти розробки веб платформи;
- Визначено функціональні та нефункціональні вимоги;
- Розроблено структуру бази даних;
- Створено сучасний інтуїтивно зрозумілий та відповідаючий вимогам користувацький інтерфейс;
- Розроблено зручну та ефективну вебплатформу для продажу товарів.

Для реалізації проєкту з розробки вебплатформи інтернет-магазину було використано мову програмування PHP та фреймворк Laravel. Також досліджено та використано можливості мови текстової розмітки HTML. За допомогою літератури було досліджено усі відповідні темі терміни та методи використання обраних інструментів. Для проєктування бази даних було використано PostgreSQL.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A. Forbes The Joy of PHP Programming: A Beginner's Guide to Programming Interactive Web Applications with PHP and MySQL, 2015. 61 c.
2. Stauffer M. Laravel Up & Running A Framework for Building Modern PHP Apps, Boston, 2021. 50c.
3. M. MacDonald HTML5: The missing manual, Sebastopol, 2011. 109 c.
4. E. Meyer, E. Weyl CSS: The definitive guide, Boston, 2023. 210 c.
5. R. Nixon Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, Sebastopol, 2018. 95 c.
6. J. Duckett HTML&CSS: Design and build websites, Hoboken, 2011. 452 c.
7. L. Ullman PHP for the web, Berkeley, 2011. 49 c.
8. I. Ben-Gan Podstawy języka T-SQL: Microsoft SQL Server 2022 i Azure SQL Database, Warszawa, 2023. 78 c.
9. A. Beaulieu Learning SQL, California, 2016, 18 c.
10. I. Jansch PHP/Architect's Guide to Enterprise PHP Development, 2008.
11. Kalob Taulien, PHP Crash Course: Learn PHP in 90 Minutes, Packt Publishing, 2020, 140 c.
12. C. Windler, A. Daubois Clean Code in PHP, California, 2022, 98 c.
13. L. Beighley, M. Morrison Head First PHP & MySQL, 2008, 539 c.
14. J.Lengstorf, T. Blom Hansen, S. Prettyman PHP 8 for Absolute Beginners: Basic Website and Web Application Development, 2022, 254 c.
15. D. Bierer PHP 8 Programming Tips, Tricks and Best Practices, 2021, 151 c.
16. E. Robson, E. Freeman Head First HTML and CSS, 2012, 425 c.
17. M. Hartl and L. Donahoe Learn Enough HTML, CSS and Layout to Be Dangerous: An Introduction to Modern Website Creation and Templating Systems, 2022, 120 c.
18. J. Meloni, J. Kyrnin Sams Teach Yourself HTML, CSS, and JavaScript All in One, 2019, 143 c.
19. B. Frain, Responsive Web Design with HTML5 and CSS, 2020, 203 c.

20. P.McFedries HTML & CSS Essentials For Dummies, 2024, 50 c.
21. M. Holzschlag Spring Into HTML and CSS, 2005, 200 c.
22. B. Downey Think Python, 2024, 10 c.
23. C. Jolowicz Hypermodern Python Tooling, 2024, 56 c.
24. N. Morgan JavaScript Crash Course, 2024, 32 c.
25. A. Osmani Learning JavaScript Design Patterns, 2023, 18 c.
26. M. Frisbie Professional JavaScript for Web Developers, 2023, 62 c.
27. B.Traversy, F.Pop 50 Projects in 50 Days - HTML, CSS, and JavaScript, 2021, 42 c.
28. L. L. Svekis, M.van Putten, R. Percival JavaScript from Beginner to Professional, 2021, 210 c.
29. A. Hoffman Web Application Security, 2024, 66 c.
30. Н. Берг, М. Найтс Amazon і майбутнє електронної торгівлі, 2023, 14 c.
31. J. Yablonski Laws of UX, 2024, 21 c.
32. A. Conta The Art and Science of UX Design: A step-by-step guide to designing amazing user experiences, 2023, 152 c.
33. J. Levy UX Strategy, 2021, 37 c.
34. R. Butti High Performance with Laravel Octane, 2023, 12 c.
35. E. Mann PHP Cookbook, “O'Reilly Media, Inc.”, 2023, 247 c.
36. C. Tanimura SQL for Data Analysis, 2021, 182 c.
37. J. Harrington Relational Database Design and Implementation, California, 2016, 492 c.
38. E. Foster, S. Godbole Database Systems, 2014, 361 c.
39. S. Saha Bagui, R. Walsh Earp Database Design Using Entity-Relationship Diagrams, 2022, 231 c.
40. C. Horstmann Modern JavaScript for the Impatient, 2020, 18 c.

ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, _____

Повністю вказується ПІБ та статус (посада для працівників, освітня (освітньо-наукова) програма – для здобувачів вищої освіти)

що нижче підписалась/підписався, розуміючи та підтримуючи загально визнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»;

що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації;

що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволений спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів;

що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративною етикою Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

(дата)

(підпис)