

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА**  
**ДИЗАЙНУ**

**ФАКУЛЬТЕТ МЕХАТРОНИКИ ТА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

***Кваліфікаційна робота***

на тему: Розроблення та оптимізація програмних засобів з підвищення  
комерційної ефективності інтернет-магазинів

Виконав: студент групи МгІТ-1-22  
спеціальності 122 Комп'ютерні науки  
освітньої програми Комп'ютерні  
науки

**Олег ДЕРЕВ'ЯНКО**

Керівник:

к.т.н., доц. Ганна КОРОГОД

Рецензент \_\_\_\_\_

к.т.н., доц. Тетяна АСТІСТОВА

Київ 2023

# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій

Кафедра комп'ютерні науки

Спеціальність 122 Комп'ютерні науки

Освітня програма Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри комп'ютерні науки**

\_\_\_\_\_ Володимир ЩЕРБАНЬ

«\_\_\_\_\_» \_\_\_\_\_ 2023\_року

## **З А В Д А Н Н Я**

НА КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТА

**Дерев'янку Олегу Олександровичу**

1. Тема роботи: Розроблення та оптимізація програмних засобів з підвищення комерційної ефективності інтернет-магазинів,  
науковий керівник роботи: Корогод Ганна Олександрівна, к.т.н., доц.,  
затверджені наказом закладу вищої освіти від 12.09.2023 року, № 210-уч.

2. Строк подання студентом роботи 14.11.2023р.

3. Вихідні дані до роботи:

Розробка кафедри комп'ютерних наук

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)

Вступ; РОЗДІЛ 1. Аналіз сучасного стану електронної комерції в Україні;  
РОЗДІЛ 2. Інформаційна модель даних та її аналіз; РОЗДІЛ 3. Вибір та застосування технологій по підвищенню комерційної ефективності веб-сайту. Додатки - програмні коди модулів системи.

## 5.Консультанти розділів кваліфікаційної роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдан ня видав	завдання прийняв
Вступ	Ганна КОРОГОД, к.т.н., доц.		
Розділ 1	Ганна КОРОГОД, к.т.н., доц.		
Розділ 2	Ганна КОРОГОД, к.т.н., доц.		
Розділ 3	Ганна КОРОГОД, к.т.н., доц.		
Виснов ки	Ганна КОРОГОД, к.т.н., доц.		

1. Дата видачі завдання: 08.2023р.

**КАЛЕНДАРНИЙ ПЛАН**

№ з / п	Назва етапів дипломної магістерської роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	30.08.2023	
2	Розділ 1 Аналіз сучасного стану електронної комерції в Україні	06.09.2023	
3	Розділ 2. Інформаційна модель даних та її аналіз	28.09.2023	
4	Розділ 3. Вибір та застосування технологій по підвищенню комерційної ефективності веб-сайту	21.10.2023	
5	Висновки	29.10.2023	
6	Оформлення кваліфікаційної магістерської роботи (чистовий варіант)	06.11.2023	
7	Здача кваліфікаційної магістерської роботи на кафедрі для рецензування	06.11.2023	
8	Перевірка кваліфікаційної магістерської роботи на наявність ознак плагіату		
9	Подання кваліфікаційної магістерської роботи на затвердження завідувачу кафедри		

Студент

\_\_\_\_\_

Олег ДЕРЕВ'ЯНКО

Науковий керівник роботи

\_\_\_\_\_

Ганна КОРОГОД

Директор НМЦУПФ

\_\_\_\_\_

Олена ГРИГОРЕВСЬКА

## АНОТАЦІЯ

### **Дерев'янку О.О. Розроблення та оптимізація програмних засобів з підвищення комерційної ефективності інтернет-магазинів**

Кваліфікаційна магістерська робота за спеціальністю 122 – «Комп'ютерні науки». – Київський національний університет технологій та дизайну, Київ, 2023 рік.

У даній кваліфікаційній магістерській роботі розглянуті актуальні питання оптимізації та підвищення комерційної ефективності інтернет-магазинів через вдосконалення їхніх веб-сайтів.

Досліджено основні аспекти методів оптимізації веб-сайтів та проведено порівняльний аналіз різних стратегій оптимізації. Розроблено інформаційну модель та відповідні рекомендації, які сприяють підвищенню конверсії та SEO-показників, забезпечують безпеку даних, швидкість завантаження та покращують користувацький досвід.

Реалізовано на прикладі розробленого веб-сайту запропоновану інформаційну модель, що, в цілому, призводить до підвищення комерційної ефективності інтернет-магазинів.

*Ключові слова: оптимізація веб-сайтів, комерційна ефективність сайту, конверсія, проектування веб-сайту, мобільна оптимізація, принципи веб-розробки, SEO-показники, користувацький досвід.*

## ABSTRACT

**OLEG DEREVYANKO.** Development and optimization of software tools to increase the commercial efficiency of online stores.

Qualifying master thesis in specialty 122 - "Computer science". - Kyiv National University of Technology and Design, Kyiv, 2023.

This master's thesis deals with the topical issues of optimizing and increasing the commercial efficiency of online stores through the improvement of their websites.

The main aspects of website optimization methods were studied and a comparative analysis of various optimization strategies was carried out. An information model and relevant recommendations have been developed, which contribute to increasing conversion and SEO-indicators, ensure data security, load speed and improve user experience.

The proposed information model was implemented on the example of the developed website, which, in general, leads to an increase in the commercial efficiency of online stores.

*Keywords: website optimization, online stores, commercial site efficiency, conversion, website design, mobile optimization, principles of web development, SEO indicators.*

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	8
ВСТУП .....	9
РОЗДІЛ 1 .....	12
АНАЛІЗ СУЧАСНОГО СТАНУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ В УКРАЇНІ	12
1.1. Зростання електронної комерції в Україні.....	12
1.1.1. Огляд стану ринку електронної комерції в Україні .....	12
1.1.2. Фактори, що впливають на зростання електронної комерції в Україні .....	13
1.1.3. Виклики та перешкоди для електронної комерції в Україні .....	14
1.2. Важливість веб-сайту для інтернет-магазину .....	16
1.3. Потенціал для підвищення комерційної ефективності .....	18
1.4. Поняття та принципи веб-розробки.....	20
1.5. Оптимізація веб-сайтів по їх комерціалізації .....	21
1.6. Постановка задач дослідження .....	25
Висновки до розділу .....	27
РОЗДІЛ 2 .....	28
ІНФОРМАЦІЙНА МОДЕЛЬ ДАНИХ ТА ЇЇ АНАЛІЗ .....	28
2.1. Технічні параметри при створенні інтернет-магазину.....	28
2.1.1. Функціонал при розробці веб-сайту.....	31
2.1.2. Дизайн та інтерфейс.....	33
2.1.3. Безпека інформації .....	36
2.2. Інформаційна модель системи.....	37
2.2.1. Структура бази даних.....	38
2.2.2. Об'єкти та їх взаємодія.....	48
Висновки до розділу .....	53
РОЗДІЛ 3 .....	54

ВИБІР ТА ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ПО ПІДВИЩЕННЮ КОМЕРЦІЙНОЇ ЕФЕКТИВНОСТІ ВЕБ-САЙТУ .....	54
3.1. Вибір технологічного стеку .....	54
3.1.1. Вибір фреймворку для веб-розробки .....	57
3.1.2. Вибір мови програмування .....	58
3.2. Проектування структури веб-сайту .....	60
3.2.1. Створення структури сторінок .....	62
3.2.2. Меню та навігація .....	68
3.3. Розробка функціоналу .....	71
3.3.1. Реєстрація та авторизація користувачів .....	72
3.3.2. Додавання товарів до кошика .....	77
3.3.3. Оформлення замовлення .....	82
3.4. Оптимізація веб-сайту по його комерціалізації .....	85
3.4.1. Пошукова оптимізація (SEO) .....	87
3.5 Тестування .....	89
3.5.1. Функціональне тестування .....	89
Висновки до розділу .....	95
ВИСНОВКИ .....	96
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	98
ДОДАТКИ .....	103
ДОДАТОК А .....	103
ДОДАТОК Б .....	109

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SEO – Search Engine Optimization (оптимізація для пошукових систем)

UI – User Interface (інтерфейс користувача)

UX – User Experience (взаємодія та досвід користувача)

ROI – Return on Investment (повернення інвестицій)

KPI – Key Performance Indicator (ключовий показник результативності)

CTR – Click-Through Rate (показник клікабельності)

CMS – Content Management System (система управління вмістом)

HTTPS – Hypertext Transfer Protocol Secure (безпечний протокол передачі гіпертексту)

UI/UX дизайн – Дизайн інтерфейсу та взаємодії з користувачем

SMM – Social Media Marketing (маркетинг у соціальних медіа)



## ВСТУП

У сучасному цифровому світі, який переповнений можливостями та доступом до інформації, ефективна онлайн-присутність стала необхідною для великої кількості підприємств. Споживачі вже давно перестали обмежуватися тільки офлайн покупками, адже Інтернет надав їм можливість швидко та зручно отримувати необхідні товари та послуги. Відтак, інтернет-магазини стали важливою складовою електронної комерції, яка набула надзвичайної популярності та поширення.

Проте, зростання популярності інтернет-магазинів призвело до зростання конкуренції в цій галузі. Тепер споживачі мають безліч варіантів для покупок онлайн, тож для інтернет-магазинів стало надзвичайно важливим не лише пропонувати високоякісні товари, але і забезпечувати найкращий користувацький досвід. Зокрема, інтернет-магазини стикаються з низкою викликів, серед яких:

1. Конкурентне середовище. Зараз існує велика кількість інтернет-магазинів, які конкурують між собою за увагу споживачів. Це створює необхідність для інтернет-магазинів бути інноваційними та ефективними в приверненні та утриманні клієнтів.

2. Змінні уявлення споживачів. Уявлення та очікування споживачів щодо онлайн-покупок постійно змінюються. Вони вимагають зручності, швидкості, безпеки та інноваційних рішень від інтернет-магазинів. Тому, підприємства повинні бути готові адаптуватися до цих змін, щоб задовольнити потреби своїх клієнтів.

У цьому контексті, актуальними слід вважати дослідження, які направлені на розробку та оптимізацію веб-сайтів для інтернет-магазинів, що дозволить забезпечити конкурентну перевагу та покращити їх комерційну ефективність.

**Метою** роботи є вдосконалення методів оптимізації веб-сайтів інтернет-магазинів по їх комерційної ефективності.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- проаналізувати існуючі методи оптимізації (розгляд і критичний аналіз попередніх підходів до оптимізації веб-сайтів інтернет-магазинів для ідентифікації недоліків та можливостей для вдосконалення);
- розробити інформаційну модель і відповідні рекомендації щодо покращення функціональності та дизайну веб-сайтів інтернет-магазинів з урахуванням сучасних тенденцій та очікувань клієнтів;
- застосувати розроблені рекомендації в реальні проекти інтернет-магазинів з метою оцінки їхнього впливу на комерційну ефективність;
- провести порівняльний аналіз отриманих результатів дослідження.

*Об'єктом дослідження* є процес оптимізації програмних засобів в розробці веб-сайтів.

*Предметом дослідження* є методи оптимізації веб-сайтів інтернет-магазинів по їх комерційної ефективності.

**Методи дослідження** включають аналіз, синтез, емпіричні спостереження, статистичний аналіз, моделювання та тестування. Інформаційна база дослідження складається з наукових джерел, статистичних даних, а також результатів попередніх досліджень у галузі електронної комерції. Технології для реалізації: мова програмування PHP з використанням бібліотеки JQuery та Ajax, для зберігання локальних та серверних даних, візуалізація додатку створена за допомогою каскадної таблиці стилів SCSS та фреймворку Bootstrap, гіпертекстового редактору HTML та каскадної таблиці стилів CSS.

**Наукова новизна** роботи полягає в розробці нових рекомендацій по оптимізації веб-сайтів інтернет-магазинів, які враховують сучасні тенденції у споживчій поведінці та технологічній динаміці. Результати цього

дослідження відрізняються від існуючих підходів і пропонують нові ідеї для вдосконалення веб-сайтів.

**Практична цінність** отриманих результатів полягає в комерційній ефективності застосування розроблених рекомендацій і методик.

**Апробація** результатів дослідження доповідались на міжнародній науково-практичній конференції та опубліковані у відповідних фахових виданнях.

Кваліфікаційна робота складається з вступу, розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 102 сторінок комп'ютерного тексту (без додатків).

## РОЗДІЛ 1

# АНАЛІЗ СУЧАСНОГО СТАНУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ В УКРАЇНІ

### 1.1. Зростання електронної комерції в Україні

У сучасному цифровому світі ефективна онлайн-присутність стала ключовою складовою успіху багатьох підприємств. Електронна комерція, або е-комерція, є однією з важливих галузей, яка активно розвивається в Україні. Даний розділ присвячений детальному аналізу динаміки та стану розвитку електронної комерції в Україні з огляду на ключові аспекти та фактори, що впливають на цей процес.

#### 1.1.1. Огляд стану ринку електронної комерції в Україні

Сучасний стан ринку електронної комерції в Україні характеризується наступними показниками:

1. Загальна динаміка зростання обсягів електронної комерції в країні: В останні роки спостерігається вражаючий ріст електронної комерції в Україні. За даними *Delta International Services*, обсяги е-комерції в країні зросли на 27% у 2022. Це свідчить про зростання популярності онлайн-покупок серед українських споживачів.

2. Зміни в споживчій поведінці українських споживачів, пов'язані з онлайн-шопінгом: За даними *Delta International Services* 44% українців зазначають, що вони регулярно роблять покупки в інтернеті. Це свідчить про значний вплив електронної комерції на споживчу поведінку.

3. Основні сектори та галузі, які найбільше розвиваються в електронній комерції.

1. Онлайн-торгівля продовольчими товарами. Західні регіональні супермаркети та магазини продуктів харчування все більше вкликають до себе споживачів через інтернет-платформи, що робить цей сектор одним із найшвидше зростаючих.

2. Одяг та мода. Багато онлайн-магазинів з одягом, взуттям та аксесуарами заробляють популярність серед українців, які шукають зручний і простий спосіб шопінгу.

3. Електронна та побутова техніка. Онлайн-продажі побутової та електронної техніки також стали дуже популярними, оскільки люди шукають оновлення та нові технології.

4. Ігри та розваги. Галузь онлайн-ігор та розваг, включаючи відеоігри та інтернет-гральні клуби, також демонструє зростання.

5. Маркетплейси та платформи. В Україні активно розвиваються маркетплейси та платформи, які дозволяють підприємцям продавати свої товари та послуги онлайн.

### **1.1.2. Фактори, що впливають на зростання електронної комерції в Україні**

Ключовими чинниками, які сприяють зростанню сфери електронної комерції в Україні, як відомо, є:

1. *Покращення доступності Інтернету та інфраструктури.* Останніми роками в Україні відбулися помітні зміни у покращенні якості та швидкості Інтернет-з'єднань. Це сприяло збільшенню кількості користувачів мережі та полегшило процес онлайн-покупок. За даними Всесвітнього банку, у 2021 році понад 43% населення України мали доступ до Інтернету.
2. *Зміни у регулюючому середовищі та законодавстві для електронної комерції.* Україна активно вдосконалює своє законодавство щодо

сфери електронної комерції та прав споживачів. Це сприяє створенню довіри серед як покупців, так і підприємців, що є важливим для зростання електронних торгівельних платформ. Наприклад, введення в дію Закону України "Про електронну комерцію" сприяє розвитку цього сегменту.

3. *Розширення спектру онлайн-платіжних систем та спрощення процедур оплати.* Завдяки збільшенню кількості доступних онлайн-платіжних систем та послуг, покупці мають більше можливостей для зручної оплати товарів і послуг. Це сприяє зростанню обсягу онлайн-транзакцій.
4. *Розвиток мобільної торгівлі.* Велика кількість осіб в Україні використовує смартфони для здійснення покупок. Розробка мобільних додатків та адаптація веб-сайтів до мобільних пристроїв сприяє збільшенню доступності та зручності онлайн-шопінгу.
5. *Розвиток онлайн-маркетплейсів та інтернет-платформ.* Все більше продавців обирають онлайн-маркетплейси та інтернет-платформи для представлення своїх товарів і послуг. Це робить вибір для покупців більш різноманітним і полегшує підприємцям ведення електронної комерції.
6. *Використання цифрових маркетингових стратегій.* Підприємства активно застосовують цифрові маркетингові методи, такі як соціальні мережі, інтернет-реклама та електронна пошта, для привертання уваги та залучення клієнтів до своїх онлайн-магазинів.

Ці чинники, разом з іншими, сприяють позитивному росту сфери електронної комерції в Україні, роблячи її більш доступною, зручною та популярною як для споживачів, так і для підприємців.

### **1.1.3. Виклики та перешкоди для електронної комерції в Україні**

Електронна комерція в Україні стикається з численними викликами, серед яких можна виділити такі:

1. *Питання безпеки онлайн-платежів та захисту особистих даних користувачів.* Збільшення обсягів онлайн-платежів призводить до зростання ризику кібератак та шахрайства. Споживачі та підприємства мають бути особливо обережними стосовно захисту фінансових даних і конфіденційності особистої інформації. Важливо впроваджувати технічні заходи безпеки для захисту платіжних даних.
2. *Низький рівень довіри споживачів до онлайн-торгівлі та електронних платіжних систем.* Споживачі можуть бути не впевнені у якості товарів, можливості обману, або недостатньою надійністю платіжних систем. Для подолання цього виклику важливо покращувати обслуговування, використовувати надійні платіжні системи та надавати гарантії на продукцію.
3. *Конкуренція та боротьба за ринкові позиції серед інтернет-магазинів.* Зростання популярності електронної комерції призвело до з'яви великої кількості інтернет-магазинів, що призводить до жорсткої конкуренції. Для досягнення успіху важливо розробляти ефективні маркетингові стратегії, покращувати обслуговування та пропонувати конкурентоспроможні ціни.
4. *Логістичні виклики.* Доставка товарів та обробка повернень можуть бути складними завданнями через велику територіальну протяжність та інфраструктурні особливості України. Покращення логістики та управління постачальницьким ланцюгом може допомогти зменшити ці труднощі.
5. *Податкова та митна політика.* Податкова та митна політика може бути складною для електронних підприємств, і зміни у цих правилах можуть вплинути на їхні витрати та оподаткування.

6. *Недостатня освіченість підприємців та споживачів щодо електронної комерції.* Багато малих підприємств та користувачів можуть бути недостатньо освіченими щодо можливостей електронної комерції, що може уповільнювати ріст цього сектору. Важливо проводити освітні заходи та інформаційні кампанії для підвищення рівня обізнаності.

Загальною метою є створення сприятливого та безпечного середовища для розвитку електронної комерції в Україні, яке б відповідало потребам як підприємців, так і споживачів.

## **1.2. Важливість веб-сайту для інтернет-магазину**

Веб-сайт як необхідний елемент інфраструктури сучасного інтернет-магазину має важливе значення для забезпечення успішної його діяльності. Розглянемо фактори, що роблять веб-сайт важливим для ефективного функціонування інтернет-магазину:

1. *Представлення продуктів та послуг.* Веб-сайт є ключовим засобом візуального представлення товарів та послуг інтернет-магазину. Він дозволяє докладно описати товарний асортимент, додати високоякісні фотографії, вказати технічні характеристики та ціни. Це надає можливість покупцям здійснювати обдуманий вибір, що є ключовим для збільшення обсягів продажів.

2. *Зручність для клієнтів.* Веб-сайт інтернет-магазину забезпечує зручну платформу для клієнтів, де вони можуть шукати товари, переглядати їх, порівнювати, читати відгуки і здійснювати покупки в будь-який час та з будь-якого місця. Це надзвичайно важливо для задоволення потреб сучасних клієнтів, які цінують свій час і комфорт.

3. *Можливість онлайн-замовлень.* Веб-сайт дозволяє клієнтам здійснювати замовлення онлайн без необхідності фізичного відвідування



магазину або проведення додаткових дій. Це спрощує процес покупки та забезпечує зручність для клієнтів.

4. *Маркетинг і просування.* Веб-сайт інтернет-магазину є платформою для впровадження різноманітних маркетингових стратегій і інструментів. Сюди входить пошукова оптимізація (SEO), контент-маркетинг, інтернет-реклама та реклама у соціальних медіа. Ці інструменти допомагають залучати нових клієнтів і підвищувати обсяги продажів.

5. *Підвищення рівня довіри.* Професійно розроблений веб-сайт містить важливу інформацію про компанію, контактні дані та політику конфіденційності. Це сприяє збільшенню довіри клієнтів до бренду та гарантує конфіденційність їх особистих даних. Користувачі також можуть знайти відгуки та рейтинги товарів, що дозволяє їм робити обдуманий вибір.

6. *Глобальний доступ та розширення ринків.* Веб-сайт інтернет-магазину дозволяє працювати в режимі 24/7 та обслуговувати клієнтів з будь-якого регіону чи країни. Це відкриває нові ринки і можливості для розширення бізнесу.

7. *Аналітика і відстеження.* Веб-сайт дозволяє збирати та аналізувати важливі дані про активність користувачів. Інструменти аналітики дозволяють власникам інтернет-магазину визначити популярні товари, підвищити ефективність маркетингу і поліпшити обслуговування клієнтів.

8. *Контроль над брендом і іміджем.* Веб-сайт дозволяє створити і керувати онлайн-брендом і іміджем інтернет-магазину. Це важливо для створення позитивного враження серед клієнтів і підвищення конкурентоспроможності на ринку.

Загалом, веб-сайт інтернет-магазину є важливим інструментом, який сприяє ефективному функціонуванню, залученню клієнтів, а також надає конкурентні переваги на сучасному ринку електронної комерції.

Використання веб-сайту стає стратегічним кроком для досягнення успіху в електронному бізнесі.

### 1.3. Потенціал для підвищення комерційної ефективності

У сучасній цифровій епохі інтернет-магазини відіграють ключову роль у розвитку електронної комерції, але успіх в цьому конкурентному середовищі залежить від багатьох чинників, які покращують комерційну ефективність веб-сайту. Серед головних таких чинників є:

1. *Збільшення конверсії.* Одне з основних завдань цієї роботи – збільшення конверсії. Це означає, що більше відвідувачів сайту перетворюються в клієнтів. Шляхом аналізу даних та проведення тестів А/В можна визначити, які зміни на сайті призводять до зростання конверсії і впроваджувати їх.

2. *Покращення користувацького досвіду.* Важливо створити веб-сайт, який надає користувачам зручний та приємний досвід. Це включає в себе швидкість завантаження сторінок, адаптивний дизайн для різних пристроїв, зрозумілу навігацію та інтуїтивний процес замовлення.

3. *Підвищення відомості про бренд.* Веб-сайт інтернет-магазину може виступати як важливий інструмент для підвищення відомості про бренд. Інформація про компанію, історія, місія та цінності можуть бути представлені на сайті, що допомагає створити позитивне враження про бренд серед клієнтів.

4. *Оптимізація для пошукових систем (SEO).* Ефективна оптимізація сайту для пошукових систем допомагає залучати більше органічного трафіку. Це включає в себе оптимізацію ключових слів, структуру сайту, місцевий пошук і багато інших факторів, які підвищують видимість сайту в результатах пошуку.

5. *Аналіз та відстеження даних.* Застосування інструментів аналітики дозволяє стежити за активністю користувачів на сайті, визначати популярність товарів і сторінок, а також ідентифікувати можливі проблеми в процесі замовлення. Аналітика надає дані, необхідні для вдосконалення функціональності сайту.

6. *Розробка мобільної версії.* Зростаюча кількість користувачів використовує мобільні пристрої для онлайн-покупок. Розробка мобільної версії сайту або використання адаптивного дизайну допоможе залучити та зберегти цю аудиторію.

7. *Захист від кібератак.* Забезпечення безпеки клієнтських даних та фінансових операцій є критичним завданням інтернет-магазину. Важливо вдосконалити захист веб-сайту від кіберзагроз і шахрайства, щоб зберегти довіру клієнтів.

8. *Персоналізація.* Використання алгоритмів машинного навчання дозволяє створювати персоналізовані рекомендації для клієнтів на основі їхньої попередньої активності на сайті. Це підвищує шанси на успішні продажі.

9. *Застосування соціальних мереж.* Інтеграція зі соціальними мережами дозволяє інтернет-магазину підтримувати спілкування з клієнтами, отримувати відгуки та реакції на продукти та послуги, а також залучати нових клієнтів через соціальний маркетинг.

10. *Ефективна обробка замовлень і інвентаризація.* Оптимізація процесів обробки замовлень і керування інвентарем може підвищити ефективність інтернет-магазину і запобігти проблемам з нестачею або надміром товарів.

11. *Контент-маркетинг і освіта клієнтів.* Веб-сайт може бути платформою для створення і публікації корисного контенту, такого як блоги, статті, відео, які не лише привертають трафік, але і освічають клієнтів, що сприяє підвищенню довіри і комерційній ефективності.

Таким чином, розробка і оптимізація веб-сайту інтернет-магазину є важливим аспектом для підвищення комерційної ефективності. Важливо систематично досліджувати та вдосконалювати цей процес, використовуючи сучасні технології і аналіз даних для досягнення успіху в сфері електронної комерції.

#### **1.4. Поняття та принципи веб-розробки**

Веб-розробка – це складний процес створення веб-сайтів і веб-додатків, що включає в себе розробку веб-сторінок, їх функціональність, інтерактивність та дизайн. Головна мета веб-розробки полягає в створенні веб-ресурсу, який буде доступний через веб-браузери та надаватиме користувачам інформацію, послуги або можливість взаємодії з різними функціями.

Веб-розробка ґрунтується на декількох ключових принципах, які становлять основу для створення якісних та ефективних веб-сайтів. Ось докладний розгляд цих принципів:

1. *Дотримання стандартів.* Веб-розробка вимагає використання стандартів, які визначені організаціями, такими як W3C (Консорціум Всесвітньої павутини). Це включає в себе використання мови розмітки HTML для створення веб-сторінок, CSS для оформлення і JavaScript для інтерактивності.

2. *Адаптивний дизайн.* Сучасні веб-розробники повинні створювати сайти з адаптивним дизайном, що дозволяє сайту коректно виглядати на різних пристроях і екранах.

3. *Зручність для користувача (UX).* Важливо забезпечити зручний та інтуїтивний користувацький досвід, зробити навігацію легкою та функціонал доступним для всіх користувачів.

4. *Захист.* Забезпечення безпеки сайту і даних користувачів від кібератак є надзвичайно важливим. Веб-розробники повинні використовувати кращі практики забезпечення безпеки.

5. *Швидкодія і продуктивність.* Сайти повинні завантажуватися швидко, а продуктивність має бути на високому рівні.

6. *Сумісність з різними браузерами.* Сайти повинні бути сумісними з різними веб-браузерами.

7. *Масштабованість і розширюваність.* Сайт повинен бути легко розширюваним та масштабованим.

8. *Підтримка мобільних і десктопних версій.* Важливо підтримувати як мобільні, так і десктопні версії сайту.

9. *Тестування і оптимізація.* Веб-розробники мають тестувати сайт і вдосконалювати його на різних етапах розробки.

10. *Підтримка і оновлення.* Після розгортання сайту важливо надавати підтримку та регулярно оновлювати його.

Ці принципи веб-розробки становлять фундамент для створення якісних і ефективних веб-сайтів, що відповідають потребам користувачів та бізнес-цілям. Розуміння цих принципів є важливим для успішної веб-розробки та оптимізації.

## **1.5. Оптимізація веб-сайтів по їх комерціалізації**

Підвищення комерційної ефективності веб-сайтів через їх оптимізацію є важливим аспектом як веб-розробки, так і електронної комерції. Розглянемо методи та принципи оптимізації веб-сайтів. Комерціалізація тут розуміється як процес перетворення веб-сайту зі звичайного інформаційного ресурсу в засіб для отримання прибутку.

*Оптимізація конверсії*

Збільшення конверсії є ключовою метою для більшості веб-сайтів, зокрема для інтернет-магазинів та платформ, які пропонують товари та послуги. Головна мета полягає в тому, щоб зробити так, щоб більше відвідувачів вашого сайту виконували бажані дії, такі як покупки, підписки або інші акції, які ви прагнете від них отримати.

Перед тим як розпочати оптимізацію конверсії, важливо чітко визначити, що саме ви розумієте під поняттям "конверсія". У цьому контексті конверсія може означати здійснення покупок, підписок на розсилку, заповнення контактної форми або будь-яких інших дій, які ви хочете, щоб ваші відвідувачі виконували на сайті.

Після визначення цілей конверсії, важливо аналізувати воронку конверсії. Воронка - це послідовність кроків, які користувач повинен здійснити для досягнення цілі. Наприклад, це може бути перехід від головної сторінки до сторінки товару, додавання товару в корзину та оформлення замовлення.

Для досягнення оптимальної конверсії важливо виявити можливі бар'єри, які перешкоджають відвідувачам досягнути цілі. Це може бути повільне завантаження сторінок, незрозумілий інтерфейс, складнощі з оплатою або будь-які інші фактори, які впливають на користувацький досвід.

Оптимізація конверсії являє собою перетворення відвідувачів веб-сайту в клієнтів або покупців. Для цього використовуються різні методи, включаючи:

- *Проведення A/B і A/B/n тестів*, щоб визначити найефективніші варіанти дизайну, контенту, цін і інших параметрів сайту.
- *Оптимізація лендінг-сторінок*, які спеціально створені для привертання уваги і перетворення відвідувачів в клієнтів.

## *Стратегії ціноутворення та просування*

Успішність комерціалізації визначається ефективністю встановлення цін і просування продуктів або послуг на веб-сайті. При цьому слід враховувати ключові аспекти:

1. *Цінова стратегія.* Обрана стратегія ціноутворення, будь то преміум, конкурентна або стратегія знижок, має великий вплив на споживачів і може позитивно вплинути на обсяги продажів. Важливо ретельно розглянути ринкові умови та відповідність цін попитові на вашому веб-сайті.

2. *Просування.* Використання ефективних маркетингових методів стає ключовим аспектом для залучення нових клієнтів. Це включає в себе рекламу в соціальних мережах, контент-маркетинг, SEO (пошукова оптимізація) та різні маркетингові кампанії. Справжнє використання цих методів може сприяти підвищенню обігу та впізнаваності бренду.

3. *Аналіз конкуренції.* Ретельний аналіз конкурентів на ринку допомагає визначити оптимальну цінову політику та маркетингові стратегії. Враховуючи, що пропонують інші гравці на ринку, можна підкреслити унікальні переваги вашого продукту або послуги.

4. *Споживча цільова аудиторія.* Розуміння потреб та уподобань вашої цільової аудиторії грає важливу роль у встановленні цін та створенні відповідних маркетингових кампаній. Наявність чіткого образу іншого бажаного клієнта допомагає у виборі оптимальних стратегій.

5. *Аналіз даних.* Використання аналітики дозволяє вам відстежувати результати вашої цінової політики та маркетингових кампаній. Це допомагає коригувати стратегії на основі даних та вдосконалювати підходи для досягнення кращих результатів.

6. *Комунікація з клієнтами.* Постійний контакт і взаємодія зі споживачами через соціальні мережі, електронну пошту та інші канали

комунікації дозволяють підтримувати і підсилити відносини з клієнтами, відповідати на їх запити та реагувати на їхні потреби.

Ці аспекти допомагають створити ефективну стратегію для встановлення цін та просування продуктів або послуг на веб-сайті, що є важливою частиною успішної комерціалізації.

### *Покращення користувацького досвіду*

Забезпечення задовільного та комфортного користувацького досвіду на веб-сайті є ключовим аспектом комерціалізації, при цьому слід враховувати :

1. *Адаптивний дизайн.* Сайт повинен коректно відображатися на різних типах пристроїв, включаючи мобільні телефони і планшети.

2. *Зручна навігація.* Інтуїтивно зрозуміла система навігації та логічна структура сайту сприяють зручному пошуку інформації і товарів для користувачів.

### *Вдосконалення обробки замовлень і доставки*

Ефективний процес оформлення та доставки замовлень є ключовим аспектом комерціалізації. Вдосконалення обробки замовлень і доставки характеризується:

1) *спрощенням процесу оформлення замовлення.* Зменшення кількості кроків та запитів під час оформлення замовлення сприяє підвищенню конверсії.

2) *швидкістю доставки та зручним сервісом.* Забезпечення оперативної та надійної доставки товарів разом із наданням зручних опцій для повернення товарів створює позитивне враження у покупців.



## *Аналіз та відстеження результатів*

Аналіз та моніторинг даних є критичними аспектами оптимізації в контексті комерціалізації. Для цього використовуються наступні методи:

1. *Веб-аналітика*. Використовуються інструменти веб-аналітики для відстеження відвідуваності сайту, конверсії, джерел трафіку та інших ключових метрик.

2. *Отримання фідбеку від користувачів*. Звертаються до покупців, щоб дізнатися про їхній досвід та отримувати корисний фідбек, який допомагає виявити слабкі сторони та вдосконалити їх.

Оптимізація веб-сайтів для комерціалізації - це неперервний процес вдосконалення, спрямований на досягнення більшого комерційного успіху. У цьому розділі було надано ключові аспекти цього процесу, які є важливими для розуміння та впровадження як веб-розробниками, так і власниками бізнесу.

### **1.6. Постановка задач дослідження**

Головною метою даного дослідження є розробка та оптимізація веб-сайтів інтернет-магазинів з метою підвищення їх комерційної ефективності. Ця мета передбачає створення рекомендацій, які дозволять покращити функціональність інтернет-магазинів, збільшити конверсію відвідувачів в покупців та зробити веб-сайти більш прибутковими для бізнесу. Для досягнення поставленої мети було сформульовано наступні основні завдання:

- Проаналізувати існуючі методи оптимізації (розгляд і критичний аналіз попередніх підходів до оптимізації веб-сайтів інтернет-магазинів для ідентифікації недоліків та можливостей для вдосконалення);

- Розробити відповідні рекомендації щодо покращення функціональності та дизайну веб-сайтів інтернет-магазинів з урахуванням сучасних тенденцій та очікувань клієнтів;
- Застосувати розроблені рекомендації в реальні проекти інтернет-магазинів з метою оцінки їхнього впливу на комерційну ефективність;
- Провести порівняльний аналіз отриманих результатів дослідження.

## Висновки до розділу

Завдяки поліпшенню інфраструктури та розширенню доступу до Інтернету в Україні стає все більш активніший розвиток сектору електронної комерції. Розробка і оптимізація веб-сайту інтернет-магазину є важливим аспектом для підвищення комерційної ефективності. Були досліджені та проаналізовані існуючі методи та принципи оптимізації веб-сайтів.

Розглянуті ключові моменти процесу оптимізації, які є важливими для розуміння та впровадження як веб-розробниками, так і власниками бізнесу: оптимізація конверсії, стратегія ціноутворення та просування, покращення користувацького досвіду, процес обробки замовлень і доставки, моніторинг даних.

Встановлено, що застосування і вдосконалення представлених заходів по оптимізації веб-сайту сприяють підвищенню комерційної ефективності.

## РОЗДІЛ 2

### ІНФОРМАЦІЙНА МОДЕЛЬ ДАНИХ ТА ЇЇ АНАЛІЗ

#### 2.1. Технічні параметри при створенні інтернет-магазину

Визначення технічних та функціональних параметрів інтернет-магазину є ключовим елементом дослідження, оскільки вони визначають його комерційну ефективність. Визначення та врахування вимог до функціональності, дизайну та безпеки інтернет-магазину є важливим для підвищення комерційної ефективності інтернет-магазину.

#### *Коригування оптимізації*

Для кожної сторінки сайту має бути можливим прописувати вручну унікальні заголовки (Title), описи (Description), ключові слова (Keywords) та заголовки першого рівня (H1).

При цьому, потрібно передбачити автоматичну генерацію цих параметрів, оскільки вручну обробити всі сторінки інтернет-магазину може бути неможливо та неефективно. Достатньо встановити шаблон, за яким всі зазначені теги будуть генеруватися автоматично. Важливо:

1) Не повинно бути залежності між URL сторінки та її заголовком H1 або Title.

2) Необхідно уникати залежності між заголовком першого рівня (H1) та назвою пункту меню. Пункт меню може мати просту назву, наприклад "Контакти", тоді як заголовок може бути більш детальним, наприклад, "Контакти компанії".

*Можливість ручного налаштування URL – адреси*

Повинна бути надана можливість не лише прописувати довільну структуру сторінки, але і задавати її відповідність до певного розділу. При цьому система управління повинна автоматично генерувати URL-адресу з відповідною вкладеністю.

### *Меню*

На будь-якому сайті повинно бути більше одного меню: мінімум два, а іноді й більше. Перше меню - це головне меню, в якому знаходяться такі пункти, як "Головна", "Про компанію", "Контакти" та інші. Друге меню - це меню послуг або каталогу, яке зазвичай розташовується збоку. У цьому меню вже представлені товари та послуги.

### *Товари та продукти*

У будь-якому Інтернет-додатку інформація про товар повинна включати:

- артикул;
- назву;
- ціну (роздрібна, оптова або прибуткова ціна, яка буде видно в адмінці магазину);
- короткий опис – він потрібен для показу невеликої інформації про товар у списку категорій;
- детальний опис – якщо клієнт зацікавився і захотів дізнатися більше про товар, він натискає на нього та може прочитати більше інформації (HTML – відформатований текст, фото чи навіть відео);
- прив'язку товару до потрібної категорії;

- комбінації або властивості – це потрібно, якщо ви продаєте взуття чи футболки, які мають свій розмір, колір та відповідну ціну;
- фото товару;
- метадані: заголовок, опис, ключові слова.

### *Категорії*

Категорії товарів використовуються для того, щоб згрупувати продукти за типом, маркою або функціоналом. Ця можливість дозволяє клієнтам швидко знайти необхідний товар зі списку.

Категорії можуть бути вкладені. Вони також повинні містити:

- код категорії;
- назву;
- опис категорії, щоб користувач зрозумів, які продукти містяться у середині;
- фото або логотип категорії.

### *Користувачі та групи*

Для ефективної і продуктивної роботи інтернет-магазину потрібно мати зручну клієнтською базу з можливістю її фільтрації та сортуванні. Кожен користувач повинен зберігати в Інтернет-додатку таку мінімальну інформацію:

- електронну адресу;
- повне ім'я;
- адресу доставки та оплати, а також номер телефону;
- історію покупок;
- приналежність до групи магазину.

## *Замовлення*

Менеджерській відділ замовлень повинен контролювати замовлення та вміти їм керувати. У цьому випадку, необхідно мати можливість змінювати статус замовлення (в очікуванні, оплачений, у підготовці, доставка, доставлений) та повідомляти клієнта про зміни у замовленні на його електронну пошту.

## *Додатковий функціонал*

Робота з Інтернет-додатком буде ефективніша, якщо на сайті буде реалізовано додатковий функціонал:

- Супутні товари.
- Фільтри та сортування товарів.
- Онлайн-калькулятор.
- Сервіси онлайн-взаємодії з відвідувачами.

### **2.1.1. Функціонал при розробці веб-сайту**

При розробці веб-застосунку інтернет-магазину необхідно враховувати такі ключові етапи та функціональні можливості:

#### *1) Реєстрація користувача:*

- Форма реєстрації для нових користувачів;
- Перевірка унікальності інформації та створення облікового запису;
- Аутентифікація та авторизація;
- Вхід в систему за допомогою облікового запису;
- Перевірка доступу до функцій залежно від ролі користувача;

*2) Каталог товарів:*

- Відображення списку доступних виробів;
- Пошук, фільтрація та сортування виробів за різними параметрами;

*3) Кошик покупок:*

- Додавання виробу до кошика;
- Зміна кількості виробів в кошику;
- Видалення виробів з кошика;
- Обчислення загальної вартості замовлення;

*4) Оформлення замовлення:*

- Заповнення форми замовлення (адреса доставки, контактні дані тощо);
- Вибір способу доставки та оплати;
- Підтвердження та збереження замовлення;

*5) Управління замовленнями:*

- Відстеження статусу замовлення (нове, оброблене, доставлене тощо);
- Оновлення статусу замовлення після обробки;

*б) Управління адміністратора:*

- Додавання, редагування та видалення виробів;



- Управління категоріями користувачів та ролей;
- Створення облікових записів адміністраторів, менеджерів та звичайних користувачів;
- Надання доступу до певних функцій в залежності від ролі користувача;

*7) Забезпечення безпеки:*

- Захист персональних даних користувачів;
- Шифрування та безпечне зберігання паролів;
- Захист від несанкціонованого доступу до системи;

*8) Система оглядів та відгуків:*

- Можливість користувачам залишати відгуки та оцінки для виробів;
- Відображення середньої оцінки та коментарів для кожного виробу.

## **2.1.2. Дизайн та інтерфейс**

Дизайн та інтерфейс інтернет-магазину грають важливу роль у створенні комфортного та привабливого середовища для користувачів. Враховуючи використані для розробки технології, такі як HTML, CSS, JS, Bootstrap, PHP, AJAX, та MySQL, можна створити дизайн та інтерфейс, які відповідають вимогам та сприятимуть підвищенню комерційної ефективності магазину.

*Адаптивність*

Один із ключових аспектів дизайну - адаптивність. Інтернет-магазин повинен бути доступним та зручним для користувачів на різних пристроях, включаючи комп'ютери, планшети та мобільні телефони. Використані технології, такі як Bootstrap і CSS, дозволяють створити адаптивний дизайн, який автоматично адаптується до розмірів екрану користувача.

### *Візуальна привабливість*

Дизайн інтерфейсу має бути привабливим та естетично приємним для користувачів. Використані технології CSS та Bootstrap дозволяють створити сучасний та стильний вигляд інтернет-магазину. Важливо також враховувати баланс між візуальною привабливістю та швидкістю завантаження сторінок.

### *Інтуїтивна навігація*

Ефективна навігація на сайті важлива для забезпечення зручного пошуку та вибору товарів. Меню, кнопки та посилання повинні бути розташовані логічно та інтуїтивно зрозуміло для користувачів. Використані технології JavaScript і AJAX можуть допомогти створити динамічні елементи для покращення навігації.

### *Завантаження сторінок*

Завантаження сторінок має бути швидким і ефективним. Використані технології PHP та MySQL дозволяють оптимізувати роботу бази даних та запитів, щоб забезпечити швидке завантаження сторінок. Також важливо

мінімізувати використання зображень великого розміру та оптимізувати їх для швидкого завантаження.

### *Спільнотність та взаємодія*

Додавання соціальних медіа інтеграцій та функцій взаємодії (наприклад, коментарі, обговорення, можливість поділитися продуктами в соцмережах) може підвищити залучення користувачів і зробити інтернет-магазин більш соціально орієнтованим.

### *Браузерна сумісність*

Для забезпечення широкого охоплення аудиторії, веб-сайт повинен бути сумісним з різними веб-браузерами, такими як Chrome, Firefox, Safari, і Internet Explorer. Дизайн та кодування повинні враховувати різницю в рендерингу сторінок.

### *Мультимедійні елементи*

Додавання мультимедійних елементів, таких як відео та фотографії високої якості, може поліпшити споживчий досвід і допомогти користувачам краще оцінити продукти.

Враховуючи вищезазначені фактори, дизайн та інтерфейс інтернет-магазину повинні бути спроектовані таким чином, щоб забезпечити комфорт і задоволення користувачів та сприяти досягненню цілей комерціалізації.

### 2.1.3. Безпека інформації

Безпека є важливим аспектом для забезпечення довіри користувачів та захисту їхніх особистих даних та фінансової інформації. Тож, при розробленні веб-сайтів необхідно вживати заходи, що пов'язані з забезпеченням безпеки інформації в інтернет-магазині.

#### *Шифрування даних*

Використання протоколу SHA-256 є кроком у правильному напрямку для забезпечення безпеки інформації. Протокол SHA-256 використовується для шифрування та хешування паролів та інших чутливих даних користувачів. Цей метод допомагає унеможливити доступ до паролів та інших особистих даних навіть у разі незаконного доступу до бази даних.

#### *Захист від SQL-ін'єкцій*

Захист від SQL-ін'єкцій є важливим аспектом безпеки веб-сайту. Використання підготовлених запитів та механізмів очистки вхідних даних допомагає уникнути SQL-ін'єкцій, які можуть призвести до витoku чутливої інформації або порушення безпеки бази даних.

#### *Захист від крос-сайтових атак (XSS)*

Для захисту від крос-сайтових атак важливо валідувати та очищувати дані, які вводять користувачі у форми та поля на веб-сайті. Також можна використовувати HTTP-заголовки безпеки, такі як заголовок Content Security Policy (CSP), для обмеження можливостей впливу зовнішніх скриптів.

### *Захист від атак на сесії*

Забезпечення безпеки сесій є важливим аспектом. Використання захищених кукісів (secure cookies) та відповідні налаштування сервера допомагають захистити сесії користувачів від атак на них.

### *Регулярні апдейти та моніторинг*

Для забезпечення безпеки інформації важливо регулярно оновлювати програмне забезпечення, включаючи сервер, базу даних та інші компоненти. Моніторинг системи на наявність потенційних загроз і швидка реакція на події безпеки також грають важливу роль у забезпеченні безпеки.

Враховуючи вищезазначені заходи та використану технологію протоколу SHA-256, інтернет-магазин може досягти високого рівня безпеки інформації, що сприяє довірі користувачів та успішному функціонуванню бізнесу.

## **2.2. Інформаційна модель системи**

Інформаційна модель системи - це важлива складова будь-якого інтернет-магазину, оскільки вона визначає структуру та організацію даних, які обробляються та зберігаються в системі. Вона враховує всі аспекти, пов'язані з інформаційним потоком, від збору даних користувачів до їх обробки, зберігання та виведення. У контексті розробки та оптимізації веб-сайту для підвищення комерційної ефективності інтернет-магазину, інформаційна модель системи має вирішальне значення. Розглянемо деталі цієї моделі.

### 2.2.1. Структура бази даних

Система керування базами даних або СКБД (англ. Database Management System, скор. DBMS), є сукупністю програмних та лінгвістичних засобів загального або спеціального призначення, які забезпечують управління створенням та використанням баз даних.

*СКБД* – це комплекс програм, який дозволяє створити базу даних (БД) та маніпулювати даними (вставляти, оновлювати, видаляти та вибирати). Система забезпечує безпеку, надійність зберігання та цілісність даних, а також надає засоби для адміністрування БД.

Типи СКБД:

1. Реляційні
2. Ключ – значення
3. Документні
4. Графові
5. Колонкові

*Реляційні СКБД*

Класичні, реляційні СКБД найчастіше використовують для побудови рішень OLTP (Online Transaction Processing). У таких рішеннях СКБД працює з невеликими за розмірами транзакціями, але такими, що йдуть великим потоком, і при цьому від системи потрібен мінімальний час відгуку, а також можливість, за певних умов, скасувати будь-які зміни виконуваних у рамках транзакції. Якщо ви будете систему, в рамках якої потрібно зберігати значну кількість сутностей (таблиць), з різними типами зв'язків

між ними (один – до – одного, один – до – багатьох, багато – багатьох), то це швидше за все про реляційні СКБД.

Найбільш відомі СКБД такого типу – Oracle, Microsoft SQL, PostgreSQL, MySQL.

### *СКБД типу ключ – значення*

Одним із найпростіших типів СКБД є системи, що працюють у спрощеному вигляді і представляють собою таблицю з унікальним ключем та відповідними значеннями. Такі системи зазвичай використовуються для кешування, оскільки вони працюють дуже швидко, аби задовольнити запити, що повертають тільки одне значення. У деяких системах СКБД цього типу є можливість працювати повністю в пам'яті, а також задавати термін життя запису, після закінчення якого він автоматично видаляється. Redis та Memcached є одними з найбільш відомих СКБД цього типу.

### *Документні СКБД*

Документні або документно-орієнтовані СКБД – це один з найпопулярніших різновидів NoSQL СКБД, де основною одиницею логічної моделі даних є документ – структурований текст з певним синтаксисом.

Іноді зустрічаються думки, що модель даних у документних БД схожа на модель даних в об'єктно-орієнтованих базах даних. У цьому є частка правди, єдина реальна різниця між ними полягає в тому, що бази даних документів зберігають лише стан, але не поведінку.

Так само сама назва "документно-орієнтована" часом вводить в оману, і мені зустрічалися колеги, які вважали, що це база для систем документообігу. Ні, це не так.

Цікаво, що документні СКБД розвиваються досить активно, і зараз деякі з них, зокрема, підтримують перевірку схеми.

Відомими представниками таких СКБД є CouchDB, MongoDB та Amazon DocumentDB.

### *Графові СКБД*

Графові СКБД – специфічний тип, призначені для роботи з графами, з їх вузлами, властивостями та довільними відносинами між вузлами.

Дуже простий приклад, це організація зв'язків у різного типу соціальних мереж, де потрібно зберігати зв'язки між користувачами (вузлами) за різними критеріями (родинні зв'язки, колеги, спільні інтереси).

Відомі представники цього типу СКБД – Neo4j, Amazon Neptune, InfiniteGraph, InfoGrid.

### *Колонкові СКБД*

Колонкові СКБД дуже схожі на реляційні. Вони також складаються з рядків, які мають атрибути, а рядки групуються в таблиці. Відмінності в логічних моделях незначні, а на рівні фізичного зберігання даних відмінності значні.

У реляційних СКБД дані зберігаються "построчно", що означає, що для зчитування значення певної колонки, доведеться прочитати практично весь рядок, як мінімум від першої до потрібної колонки. У колонковій СКБД дані зберігаються "поколоночно", тобто колонка – це як окрема таблиця. Відповідно, читання відбувається з конкретного стовпця відразу. Насправді це працює дуже швидко (перевірено мною на кількох реалізаціях сховищ даних).



Основні переваги колонкових СКБД – ефективне виконання складних аналітичних запитів на великих обсягах, легка, практично миттєва, зміна структури таблиць з даними, плюс суттєва компресія та стиснення, що дозволяє значно економити місце.

Яскраві представники колонкових СКБД – Sybase IQ (нині SAP IQ), Vertica, ClickHouse, Google BigTable, InfoBright, Cassandra – представлені в табл.2.1.

Таблиця 2.1. Типи СКБД

Тип СКБД	Коли обирати	Приклади популярних СКБД
Реляційні	Потрібна транзакційність; висока нормалізація; велика частка операцій на вставку	Oracle, MySQL, Microsoft SQL Server, PostgreSQL
Ключ – значення	Завдання кешування та брокери повідомлень	Redis, Memcached
Документні	Для зберігання об'єктів у одній сутності, але з різною структурою; зберігання структур на основі JSON	CouchDB, MongoDB, Amazon DocumentDB
Графові	Завдання подібні до соціальних мереж; системи оцінок та рекомендацій	Neo4j, Amazon Neptune, InfiniteGraph, InfoGrid
Колонкові	Сховища даних; вибірки із складними аналітичними обчисленнями; кількість рядків у таблиці перевищує сотні мільйонів	Vertica, ClickHouse, Google BigTable, Sybase \ SAP IQ, InfoBright, Cassandra

При виборі типу СКБД слід перш за все виходити з типу розв'язуваних завдань, типів даних, перспектив зростання і масштабування. Для проекту обрана реляційна СКБД MySQL.

Схема бази даних та опис таблиць використаних для розробленого продукту представлено на рис. 2.1.

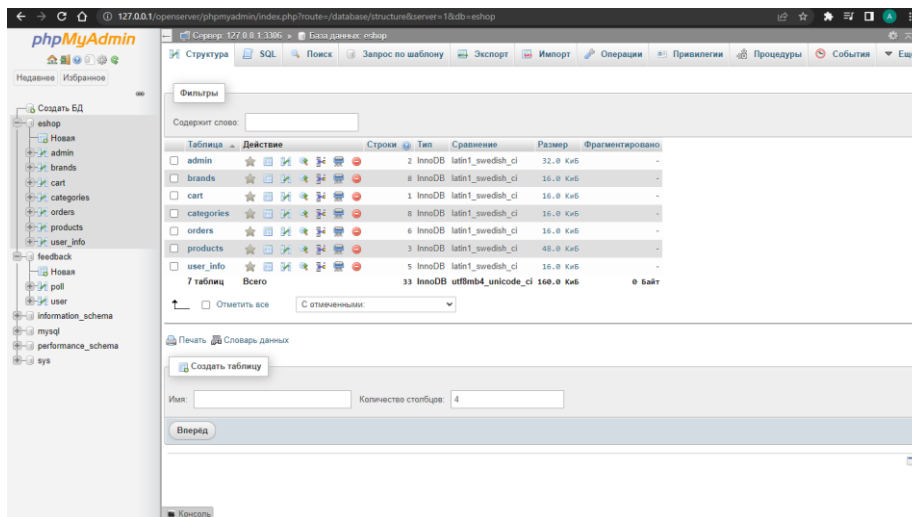


Рисунок 2.1 – Основна СУБД в PhpMyAdmin

Для роботи нашого додатку ми використовуємо основну базу даних з назвою "eshop". Ця база даних містить наступні таблиці:

- Admin - таблиця для зберігання інформації про адміністраторів додатку;
- Brands - таблиця для зберігання інформації про брендові вироби;
- Cart - таблиця для зберігання інформації про кошики покупок користувачів;
- Categories - таблиця для зберігання інформації про категорії виробів;
- Orders - таблиця для зберігання інформації про замовлення;
- Products - таблиця для зберігання інформації про роботи;
- User\_info - таблиця для зберігання інформації про користувачів додатку.

Розглянемо кожну таблицю детальніше:

*Таблиця Admin* представлена на рис. 2.2.

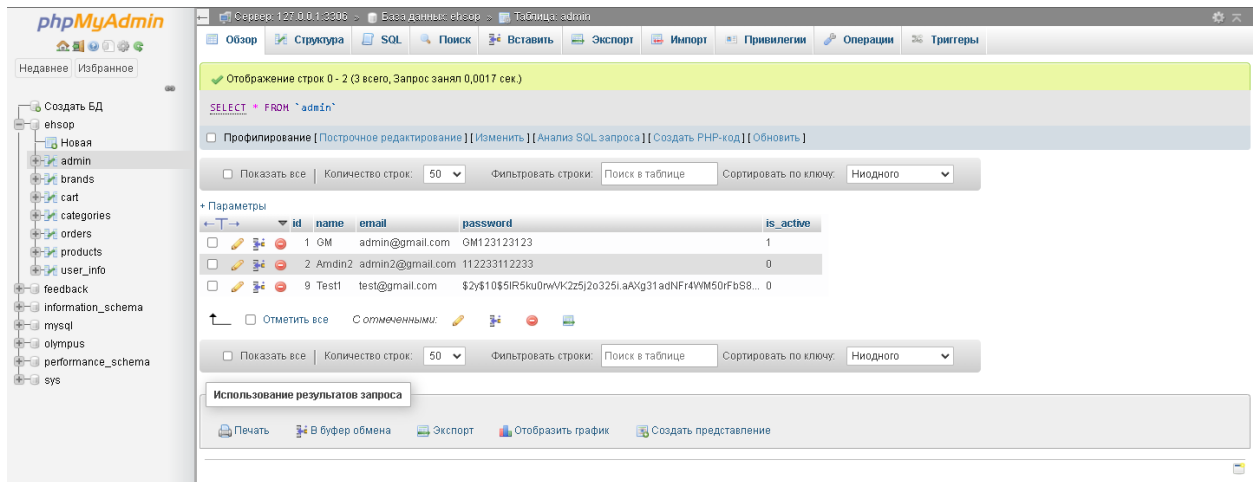


Рисунок 2.2 – Таблица Admin

Основні поля таблиці:

1. ID. Унікальний ідентифікатор користувача, в данному випадку Адміністратора. Для поля id маємо автоінкремент, який додає до кожного нового користувача новий унікальний номер.
2. Name. Поле імені, містить ім'я користувача.
3. Email. Поле електронної адреси.
4. Password. Поле пароля користувача. Якщо додавати адміністратора через бд, пароль буде без хешування. Якщо ми створюємо акаунт адміністратора через реєстрацію на сайті, пароль буде захешований і міститиме додаткові символи.
5. Is\_active. Поле активності користувача. За замовченням встановлюється 0. Для того, щоб активувати, потрібно виставити 1.

Таблиця Brands представлена на рис. 2.3.

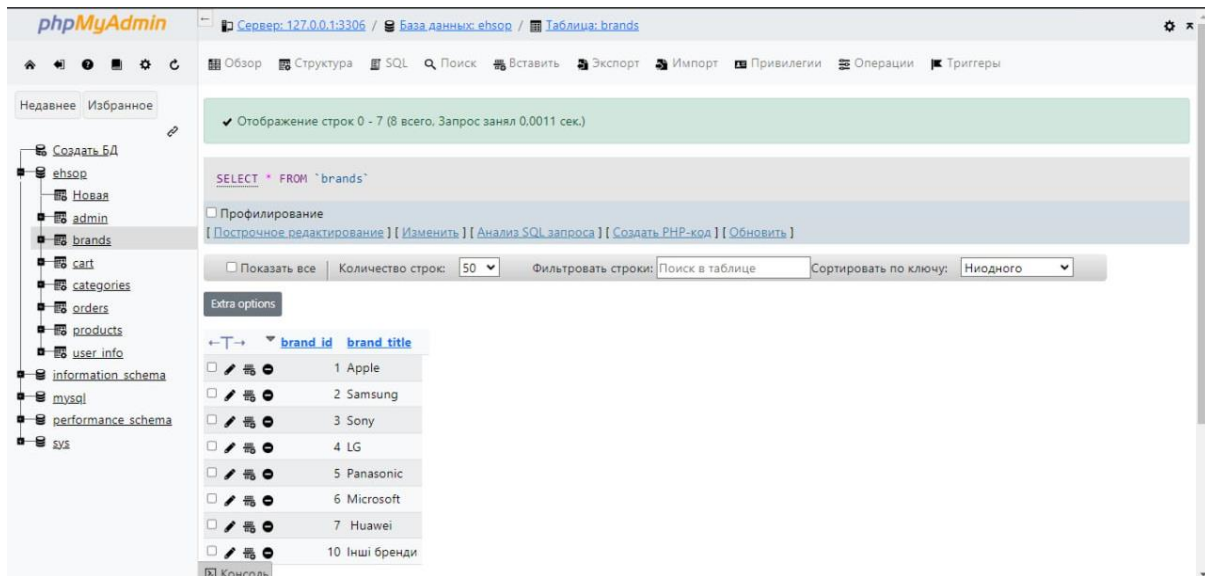


Рисунок 2.3 Таблица brands

Ця таблиця містить у собі бренди та має такі основні поля:

- Brand\_id – унікальний ідентифікатор бренду у базі даних.
- Brand\_title – поле для опису бренду.

Таблиця Cart представлена на рис. 2.4.

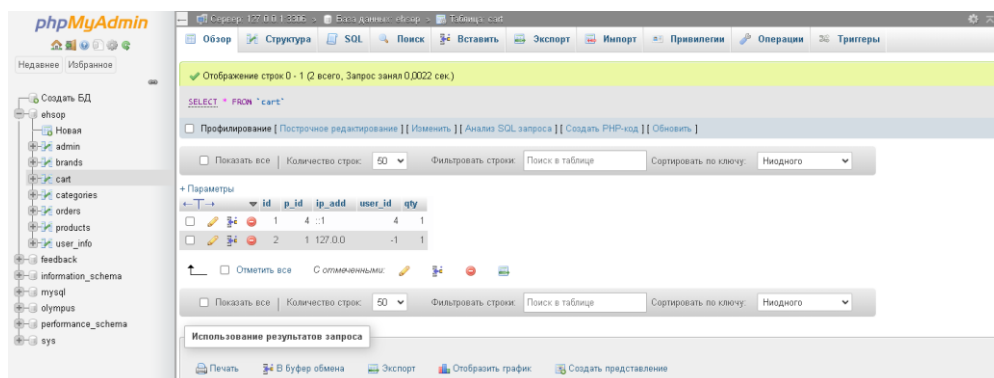


Рисунок 2.4 – Таблица кошика

Ця таблиця призначена для кошика веб-сайту та містить такі основні поля:

- Id - поле ідентифікатора

- P\_id - поле ідентифікатора товару
- Ip\_add - IP-адреса кошика, що використовується для ідентифікації при оплаті та додаванні товару
- User\_id - поле ідентифікатора користувача, який додає товар в кошик
- Qty - поле кількості товару.

Таблиця *Categories* представлена на рис. 2.5.

cat_id	cat_title
2	Смартфони і аксесуари
3	Планшети і ноутбуки
4	Аудіотехніка та навушники
5	Геймінг і ігрові консолі
6	Смарт-годинники і фітнес-гаджети
12	Дрони і роботи
13	Телевізори та домашні кінотеатри
14	Інші товари

Рисунок 2.5 – Таблиця категорій

У таблиці категорій містяться основні категорії і вона містить такі поля:

- Cat\_id - ідентифікатор категорії
- Cat\_title - поле опису категорії.

Таблиця *Orders* представлена на рис. 2.6.

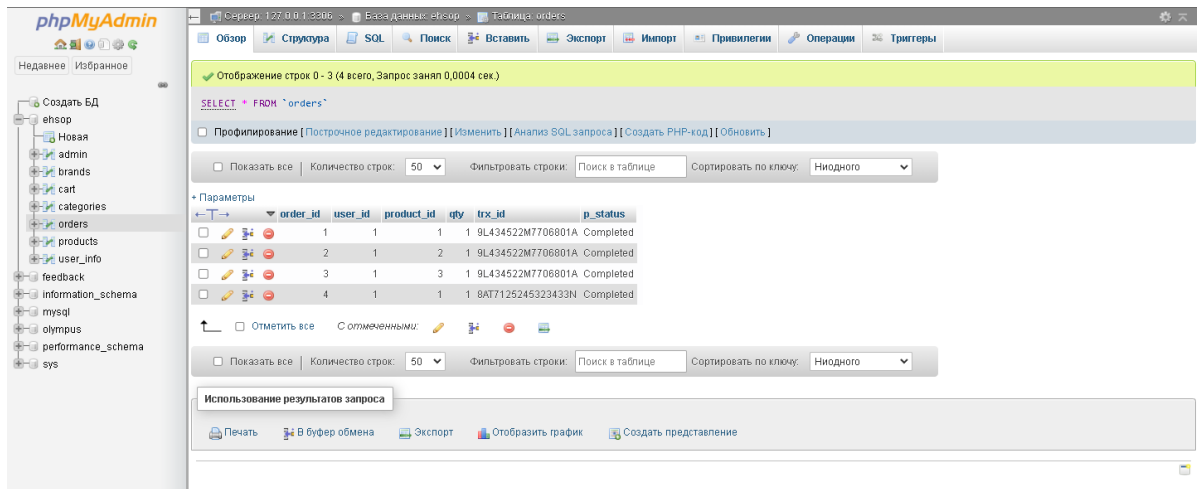


Рисунок 2.6 – Таблица order

Таблица послуг клієнтів, містить такі поля:

- Order\_id – ідентифікатор замовлення
- User\_id – ідентифікатор клієнту
- Product\_id – ідентифікатор виробу
- Qty – кількість товару для замовлення
- trx\_id – ідентифікатор замовлення, містить унікальний код замовлення

замовлення

- p\_status – стан замовлення успішне\не успішне

Таблиця Products представлена на рис. 2.7.

The screenshot shows the phpMyAdmin interface with the 'products' table selected. The table structure is as follows:

product_id	product_cat	product_brand	product_title	product_price	product_qty	product_desc	product_image	product_keywords
1	3	1	Apple MacBook Pro	1400	21	The Apple MacBook Pro is a powerful and sleek lapt...	1.jpg	macbook m1, silicon, pro, apple
2	2	2	SAMSUNG Galaxy Fold	2350	122	The SAMSUNG Galaxy Fold is a cutting-edge smartpho...	2.jpg	SAMSUNG Galaxy Fold, phone, smartphone, top
3	13	10	Домашний кинотеатр Devialet	5980	144	The Devialet Phantom II 7.0.4 DOLBY Atmos System i...	3.webp	home cinema, tv-system, audio-system
4	12	10	Дрон DJI Mini 3	1190	112	The DJI Mini 3 is a compact and	4.jpg	drone, dgi mini 3, electronics

Рисунок 2.7 – Таблица products

Таблица "products" містить картки товарів та наступні поля:

- product\_id - ідентифікатор виробу
- product\_cat - ідентифікатор категорії
- product\_brand - ідентифікатор бренду
- product\_title - назва виробу
- product\_price - ціна виробу
- product\_qty - кількість у наявності
- product\_desc - поле опису виробу
- product\_image - головне зображення для картки
- product\_keywords - ключові слова для пошуку на сайті.

Таблиця user\_info представлена на рис. 2.8.

user_id	first_name	last_name	email	password	mobile	address1	address2
1	Christine	Randolph	randolphc@gmail.com	25f9e794323b453885f51811b624d0b	8389080183	2133 Hill Haven Drive	Terra Stree
2	Will	Williams	willainswill@gmail.com	25f9e794323b453885f51811b624d0b	8389080183	4567 Orphan Road	WI
3	Demo	Name	demo@gmail.com	password	9876543210	demo ad1	ademo ad2
5	Steeve	Rogers	steeve1@gmail.com	305e4f55ce823e111a46a9d500bcb86c	9876547770	573 Pinewood Avenue	MN
6	Melissa	Gilbert	gilbert@gmail.com	305e4f55ce823e111a46a9d500bcb86c	7845554582	1711 McKinley Avenue	MA

Рисунок 2.8 – Таблиця користувачів

Таблиця користувачів містить список зареєстрованих користувачів та дані, які були вказані при реєстрації. Вона містить наступні поля:

- `user_id` – поле ідентифікатора користувача
- `first_name` – ім'я користувача
- `last_name` – прізвище користувача
- `email` – поле електронної пошти
- `password` – поле пароля користувача
- `mobile` – номер телефону користувача, має містити 10 символів
- `address1` – адреса користувача
- `address2` – альтернативна адреса.

### 2.2.2. Об'єкти та їх взаємодія

Система включає наступні об'єкти:

1. Користувачі. Зареєстровані та незареєстровані користувачі, які взаємодіють з системою через веб-інтерфейс.
2. Товари. Товари, які пропонуються в інтернет-магазині з відомостями про назву, ціну, опис і наявність.
3. Замовлення. Замовлення, створені користувачами, включають інформацію про обрані товари та адресу доставки.



4. Адміністратори. Користувачі з правами адміністратора, які керують товарами, замовленнями та іншими аспектами системи через адмін-панель.

Розробка алгоритму і його опис представлена на рис. 2.9.

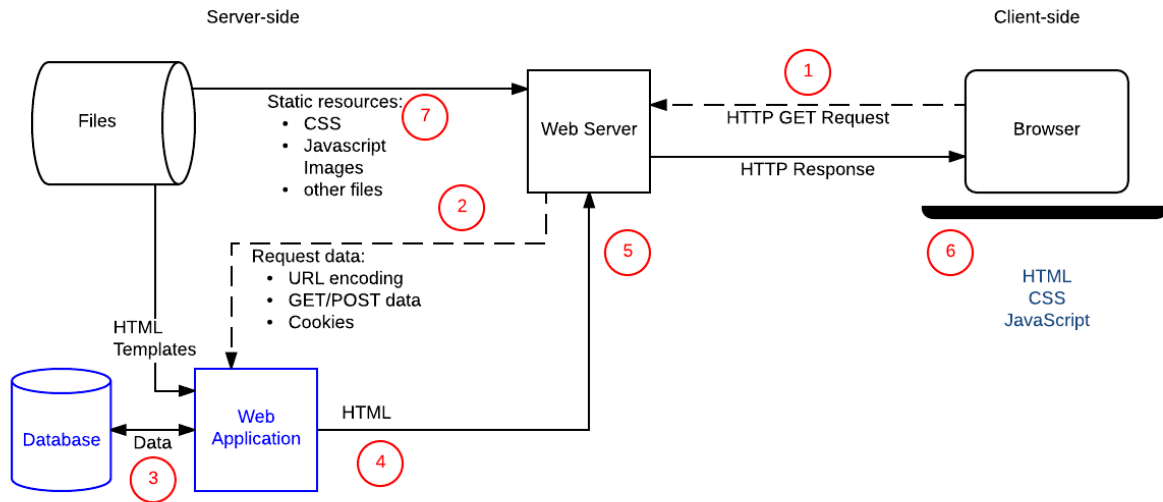


Рисунок 2.9 – Діаграма клієнт-серверної архітектури

Розробка веб-сайту та бази даних інтернет-магазину включає наступні кроки:

1. Користувач відкриває веб-браузер і вводить адресу сайту в адресний рядок.
2. Веб-браузер надсилає HTTP-запит GET на сервер, що містить базову URL-адресу ресурсу інтернет-магазину.
3. Веб-сервер отримує запит і визначає, що він відноситься до динамічної обробки.
4. Запит передається веб-додатку для подальшої обробки. Веб-сервер використовує правила відповідності шаблонів URL, визначені в його конфігурації, щоб визначити, як обробити різні URL-адреси.
5. Веб-додаток аналізує ціль запиту, яка полягає в отриманні списку товарів. Він витягує необхідні дані з бази даних, за URL-адресою.

6. Веб-додаток запитує необхідну інформацію з бази даних, використовуючи внутрішні параметри для визначення необхідної колекції. За потреби, він також може використовувати інформацію з cookie на стороні клієнта для визначення особи зареєстрованого користувача.
7. На основі отриманих даних з бази даних, веб-додаток динамічно генерує HTML-сторінку. Дані поміщаються в заповнювачі всередині HTML-шаблону.
8. Веб-додаток надсилає згенерований HTML-код разом з кодом стану HTTP 200 ("успіх") назад у веб-браузер через веб-сервер. Якщо виникають проблеми, веб-додаток може повернути інший код, наприклад, "404", що вказує, що запитана сторінка не знайдена.
9. Веб-браузер починає обробляти отриманий HTML-код і надсилає додаткові запити для завантаження будь-яких пов'язаних файлів CSS або JavaScript, на які посилається сторінка.
10. Веб-сервер завантажує статичні файли CSS і JavaScript з файлової системи і повертає їх безпосередньо в веб-браузер відповідно до правил відповідності URL-шаблонів.

Отже, процес розробки веб-сайту та бази даних інтернет-магазину включає взаємодію між веб-браузером, веб-сервером та веб-додатком, що дозволяє користувачам отримувати та взаємодіяти з необхідною інформацією.

*Діаграма прецедентів* представлена на рис. 2.10.

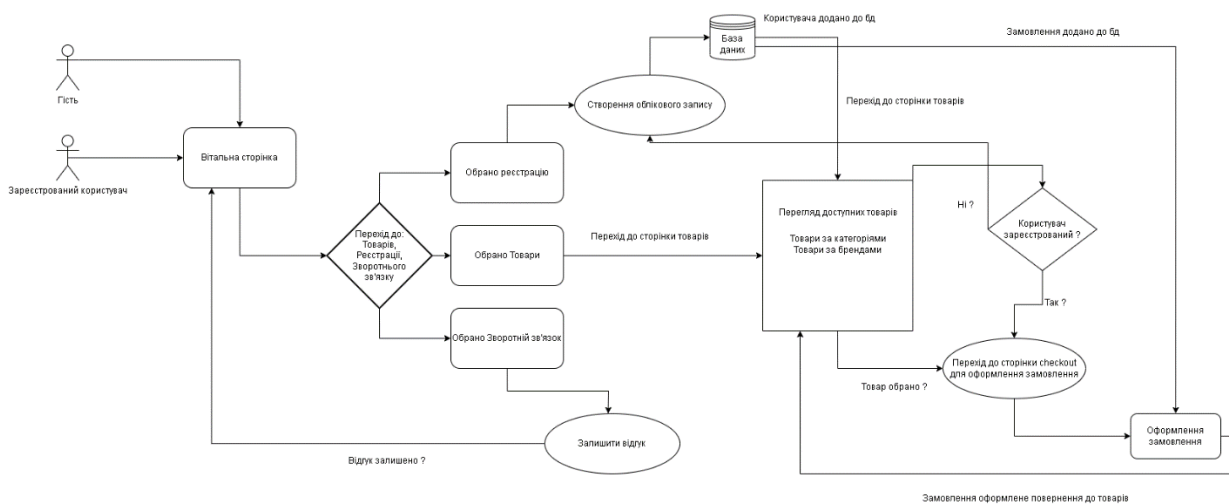


Рисунок 2.10 – Діаграма прецедентів

Нижче наведено опис діаграми прецедентів для кваліфікаційної роботи:

1. Користувач відкриває розроблений додаток у веб-браузері.
2. Система відображає головну сторінку веб-сайту зі списком доступних категорій товарів.
3. Користувач обирає категорію товарів.
4. Система відображає перелік товарів у вибраній категорії.
5. Користувач переглядає деталі товару.
6. Система відображає інформацію про обраний товар, включаючи його назву, опис, ціну та зображення.
7. Користувач додає товар до кошика.
8. Система оновлює вміст кошика, відображаючи кількість обраних товарів та загальну суму.
9. Користувач переглядає зміст кошика та робить необхідні зміни.
10. Користувач оформлює замовлення на покупку товарів.
11. Система отримує від користувача необхідну інформацію для оформлення замовлення (ім'я, адреса доставки тощо).
12. Система перевіряє наявність товарів у кошику та доступність вказаної адреси доставки.

13. Система формує замовлення та зберігає його в базі даних.
14. Система генерує підтвердження замовлення для користувача.
15. Користувач отримує підтвердження замовлення та інформацію про спосіб оплати.
16. Користувач здійснює оплату за допомогою обраного способу.
17. Система підтверджує отримання оплати та оновлює статус замовлення.
18. Користувач отримує підтвердження оплати та інформацію про очікуваний термін доставки.
19. Адміністратор системи отримує повідомлення про нове замовлення.
20. Адміністратор системи перевіряє наявність товарів на складі та підтверджує замовлення для подальшої обробки.
21. Адміністратор системи формує замовлення для доставки та надсилає його службі доставки.
22. Система оновлює статус доставки та надсилає відповідне повідомлення користувачеві.
23. Користувач отримує повідомлення про відправку замовлення та інформацію про номер відстеження посилки.
24. Користувач отримує замовлення та надає відгук про придбаний товар та якість обслуговування.
25. Система зберігає відгук користувача та відображає його на веб-сайті.

Це загальний опис послідовності подій, які відбуваються у системі під час взаємодії з користувачами.

## Висновки до розділу

У другому розділі роботи розглянуті вимоги до інтернет-магазину, включаючи функціональні вимоги, дизайн та інтерфейс, а також заходи безпеки інформації. Зокрема використання алгоритму SHA-256 для захисту паролів, дозволило забезпечити конфіденційність та цілісність даних користувачів.

Було розглянуто інформаційну модель системи, включаючи структуру бази даних і опис об'єктів та їх взаємодію.

Інформаційна модель системи була детально розглянута, включаючи структуру бази даних і опис об'єктів та їх взаємодію, що сприятиме подальшій розробці та оптимізації веб-сайту інтернет-магазину.

## РОЗДІЛ 3

### ВИБІР ТА ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ПО ПІДВИЩЕННЮ КОМЕРЦІЙНОЇ ЕФЕКТИВНОСТІ ВЕБ-САЙТУ

#### 3.1. Вибір технологічного стеку

Вибір серверного середовища є важливим кроком при розробці інтернет-магазину, оскільки від цього залежить продуктивність, надійність і швидкість роботи системи. Нижче розглянуто основні аспекти вибору серверного середовища:

1. *Операційна система.* Для операційної системи рекомендується використовувати Windows, оскільки вона відома своєю стабільністю, безпекою і підтримкою великої кількості серверних програм.
2. *Веб-сервер.* У зв'язку з використанням PHP, Apache або Nginx є відмінними виборами. Вони дозволять обробляти HTTP-запити та відповіді на них ефективно та надійно.
3. *Мови програмування.* За обраним стеком технологій PHP та JS ідеально підходять для розробки серверної та клієнтської частини веб-додатка.
4. *База даних.* Враховуючи потреби інтернет-магазину, MySQL є добрим вибором для зберігання та обробки даних, оскільки вона підтримує об'ємність та швидкодію.
5. *Хмарні сервіси.* Якщо потрібна масштабованість та надійність, можна розглянути можливість використання хмарних сервісів, таких як AWS або Azure, для забезпечення високої доступності.
6. *Заходи безпеки.* Оскільки безпека є пріоритетом, рекомендується використовувати SHA-256 для хешування паролів користувачів і застосовувати засоби захисту від атак.

7. *Підтримка і спільнота.* Важливо переконатися, що вибране серверне середовище має активну спільноту та надає регулярні оновлення та підтримку.

### *Вибір середовища розробки*

*Visual Studio Code* – це повнофункціональний текстовий редактор для редагування локальних файлів чи бази коду. Він включає різні функції для редагування бази коду, яка допомагає розробникам відстежувати зміни. Різні функції, що підтримуються в VScode:

- Підсвічування синтаксису
- Технологія автодоповнення
- Розпізнавання типів файлів
- Бічна панель із файлами вказаного каталогу
- Відлагодження
- Розширення та пакети

VS Code використовується як інтегрований редактор розробки (IDE), як Sublime і NetBeans. Поточна версія редактора VS Code сумісна з різними операційними системами, такими як Windows, Linux та MacOS.

*Open Server Panel* – це портативний програмний комплекс, створений для того, щоб допомогти веб – майстрам у розробці, налагодженні та тестуванні сайтів безпосередньо на комп'ютері (навіть якщо на ньому немає інтернету) під керуванням ОС Windows.

Можливості програми:

- Непомітна робота у треї Windows;
- Швидкі старт та зупинка;
- Автостарт сервера під час запуску програми;

- Декілька режимів управління доменами;
- Монтування віртуального диска;
- Підтримка керування через командний рядок;
- Підтримка профілів установок;
- Зручний перегляд логів усіх компонентів;
- Перемикання HTTP, MySQL та PHP модулів;
- Детальна та зрозуміла документація;
- Швидкий доступ до шаблонів конфігурації;
- Мультимовний інтерфейс;
- Автозапуск програм за списком;

#### Особливості середовища

- Не потребує встановлення (портативність);
- Одночасна робота з Denwer, Xampp тощо;
- Робота на локальному/мережевому/зовнішньому IP;
- Підтримка SSL без будь – якої додаткової. налаштування;
- Створення домену шляхом створення звичайної папки;
- Підтримка кирилических доменів;
- Підтримка аліасів (доменних покажчиків);
- Захист сервера від зовнішнього доступу;
- Runuscode конвертер доменних імен;

*PhpMyAdmin* - це веб-додаток з відкритим кодом, написаний мовою PHP, який є веб-інтерфейсом для адміністрування СКБД MySQL. PhpMyAdmin дозволяє здійснювати адміністрування сервера MySQL через браузер і не тільки, запускати команди SQL та переглядати вміст таблиць та баз даних. Програма користується великою популярністю серед веб-розробників, оскільки дозволяє управляти СКБД MySQL без безпосереднього введення SQL-команд.



### 3.1.1. Вибір фреймворку для веб-розробки

При розробці інтернет-магазину на основі технологій PHP, JS, HTML, CSS, BOOTSTRAP, AJAX та із заходом безпеки SHA-256, вибір фреймворку є критичним кроком, який може суттєво спростити процес розробки та покращити структуру та продуктивність веб-додатку. Ось детальний аналіз вибору фреймворку:

1. PHP Фреймворки. Оскільки PHP використовується як серверна мова програмування, можливо розглянути популярні PHP фреймворки, такі як Laravel, Symfony, або Yii. Laravel, наприклад, відомий своєю елегантним синтаксисом та багатим функціоналом, що спрощує розробку та забезпечує безпеку додатку.
2. JS Фреймворки. Для клієнтської сторони веб-додатку, можна розглянути використання JS фреймворків, таких як React, Angular або Vue.js. Вони надають інструменти для створення інтерактивних та ефективних інтерфейсів.
3. Інтеграція технологій. Вибір фреймворку повинен підтримувати легку інтеграцію з вже використовуваними технологіями та бібліотеками, такими як BOOTSTRAP для розробки користувацького інтерфейсу та AJAX для асинхронного обміну даними.
4. Безпека. Фреймворк повинен мати вбудовані засоби безпеки та підтримувати практики безпеки, такі як валідація даних, захист від SQL-ін'єкцій та CSRF атак.
5. Швидкодія. Фреймворк повинен бути відомий своєю швидкістю та ефективністю, щоб забезпечити швидку відповідь веб-додатку та мінімізувати завантаження сервера.

6. Підтримка і спільнота. Необхідно обирати фреймворк, який має активну спільноту розробників та регулярні оновлення, щоб забезпечити тривалу підтримку.

Зважаючи на ці фактори, вибір фреймворку для веб-розробки важливий для успішної розробки інтернет-магазину. Розглянуті аспекти допоможуть зробити обґрунтований вибір, який відповідає потребам проекту.

### 3.1.2. Вибір мови програмування

#### JavaScript

*JavaScript* – це динамічна мова комп'ютерного програмування. Він легкий і найчастіше використовується як частина веб – сторінок, реалізації яких дозволяють сценарію на стороні клієнта взаємодіяти з користувачем і створювати динамічні сторінки. Це інтерпретована мова програмування з об'єктно – орієнтованими можливостями.

Специфікація ECMA – 262 визначила стандартну версію основної мови JavaScript.

- JavaScript — це легка інтерпретована мова програмування.
- Призначений для створення мережевих додатків.
- Доповнює та інтегрується з Java.
- Доповнює та інтегрується з HTML.
- Відкрита та кросплатформна

#### Php

*Php* – це популярна мова програмування загального призначення, яку можна використовувати для різноманітних програм. Вона включає

високорівневі структури даних, динамічний тип, динамічне зв'язування та багато інших функцій, які роблять його таким же корисним для розробки складних програм, як і для сценаріїв або «склеюючого коду», який з'єднує компоненти разом. Його також можна розширити, щоб здійснювати системні виклики майже до всіх операційних систем і виконувати код, написаний на C або C++. Завдяки повсюдному поширенню та здатності працювати майже на будь-якій системній архітектурі, php є універсальною мовою, яку можна знайти в різних веб-застосунках.

*HTML* – це мова розмітки, яка використовується для створення статичних веб – сторінок. HTML використовується для створення різних компонентів сторінки, таких як заголовки, списки, форми і т.д.

*CSS* – це мова опису стилів, яка використовується для керування виглядом і форматуванням веб-сторінок. CSS дозволяє відокремлювати зовнішній вигляд веб-сторінки від її вмісту, що дозволяє забезпечити більш просте і швидке редагування вигляду сторінки.

*Bootstrap* – це набір інструментів і бібліотек для розробки веб-додатків, який дозволяє створювати відзнаки, кнопки, форми, таблиці і інші елементи веб – сторінки. Bootstrap забезпечує просту та швидку розробку веб-сторінок з використанням готових компонентів.

*AJAX* – це технологія, яка дозволяє взаємодіяти з сервером без перезавантаження сторінки. AJAX дозволяє отримувати і передавати дані асинхронно, без перезавантаження сторінки, що дозволяє створювати більш динамічні інтерфейси для користувачів. Завдяки AJAX, користувач може виконувати різні дії, наприклад, відправляти форми, оновлювати дані на сторінці, без необхідності перезавантаження сторінки.

*Технології PHP, HTML, CSS та JS* є основними технологіями веб-розробки. *HTML (Hypertext Markup Language)* є основною мовою розмітки для створення веб-сторінок, яка відповідає за структуру та контент сторінки. *CSS (Cascading Style Sheets)* використовується для оформлення та стилізації

сторінки, дозволяючи змінювати вигляд сторінки з метою покращення її естетичної привабливості та читабельності.

*JavaScript (JS)* є мовою програмування, яка використовується для динамічного створення веб-сторінок, додавання взаємодії з користувачем та створення багатофункціональних додатків. *PHP (Hypertext Preprocessor)* є мовою програмування, яка використовується для створення динамічних веб-сторінок та додатків. *PHP* дозволяє забезпечувати взаємодію з базами даних, оброблювати форми, генерувати контент на сторінках та багато іншого.

Таким чином, вибір цих технологій для розробки веб-додатку є досить популярним серед веб-розробників, тому що вони є основними технологіями веб-розробки та мають велику кількість готових бібліотек та інструментів для швидкої розробки веб-додатків. Використання Bootstrap дозволяє швидко створювати готові елементи дизайну, що економить час розробки. AJAX дозволяє створювати більш динамічні та інтерактивні веб-додатки, забезпечуючи швидкий та безперервний обмін даними між користувачем та сервером.

Використання HTML, CSS та JS дозволяє створювати красиві та ефективні веб-сторінки з багатофункціональними можливостями. Використання PHP дозволяє створювати динамічні веб-сторінки зі змінним контентом та можливістю взаємодії з базами даних.

Загалом, використання цих технологій забезпечує швидку та ефективну розробку веб-додатків з багатофункціональними можливостями та інтерактивними елементами, що відповідають потребам сучасного користувача.

### **3.2. Проектування структури веб-сайту**

Проектування структури веб-сайту є критичним етапом в розробці інтернет-магазину, оскільки воно визначає організацію інформації, навігацію та способи взаємодії користувачів із системою. Процес проектування структури включає наступні кроки:

*1. Аналіз функціональних вимог:*

- Першим кроком є докладний аналіз функціональних вимог до веб-сайту, які були визначені на попередніх етапах розробки.
- Розглядаються функції, які повинні бути доступні для користувачів, такі як перегляд товарів, додавання товарів до кошика, реєстрація та авторизація користувачів, оформлення замовлення і багато інших.

*2. Визначення структури сторінок:*

- На основі функціональних вимог визначається, які сторінки будуть присутні на сайті. Це може включати головну сторінку, сторінки категорій товарів, сторінки товарів, сторінки кошика, сторінки оплати та інші.
- Для кожної сторінки визначається її мета і функціонал.

*3. Розробка структури навігації:*

- Визначається, як користувачі будуть переходити між різними сторінками сайту. Розробляються системи меню, посилань та кнопок для навігації.
- Навігація повинна бути логічною та інтуїтивно зрозумілою для користувачів.

*4. Розробка структури URL-адрес:*

- Для кожної сторінки визначається URL-адрес, який допомагає організувати структуру сайту та полегшує пошукову оптимізацію (SEO).
- URL-адреси повинні бути описовими і відображати зміст сторінки.

*5. Розробка схеми сайту (карти сайту):*

- Схема сайту включає в себе ієрархію всіх сторінок із зазначенням зв'язків між ними. Це допомагає визначити структуру сайту та логіку навігації.

- Схема сайту може бути представлена у вигляді діаграми чи списку сторінок.

#### *6. Планування адаптивного дизайну:*

- Враховуючи різні пристрої, на яких буде відкриватися сайт (планшети, смартфони, комп'ютери), розробляється план адаптивного дизайну. Це включає в себе розміщення та структуру елементів для оптимального відображення на різних розмірах екрану.

#### *7. Тестування та вдосконалення:*

- Завершальним етапом є тестування структури сайту, перевірка на відповідність функціональним вимогам та коректність навігації.

- Після виявлення помилок проводиться вдосконалення структури.

Правильно розроблена структура сайту допомагає забезпечити зручну навігацію, логічну організацію контенту та задоволення потреб користувачів, що є важливим для успішного інтернет-магазину.

### **3.2.1. Створення структури сторінок**

Створення структури сторінок є ключовим етапом у розробці веб-сайту інтернет-магазину. На цьому етапі визначається, які сторінки будуть присутні на сайті та як вони будуть взаємодіяти між собою.

Приклад основної файлової структури проекту представлено на рис. 3.1.

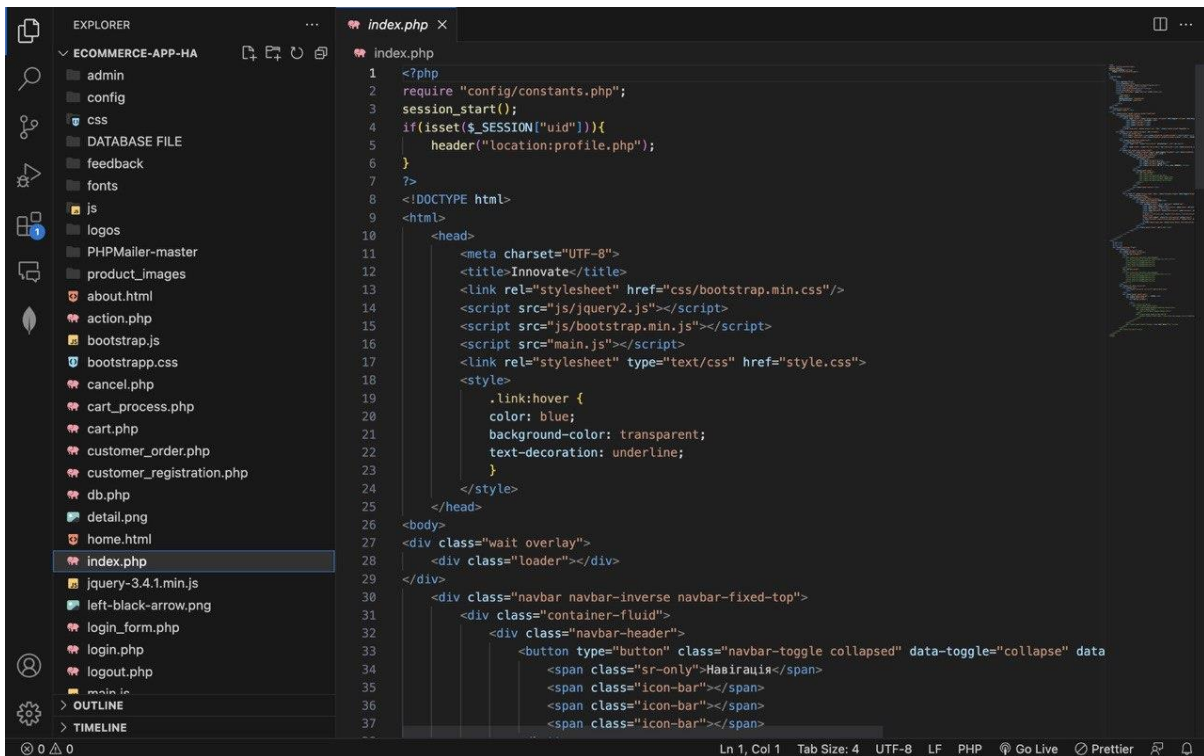


Рисунок 3.1 – Основна файлова структура проекту

Для створення структури сторінок інтернет-магазину на основі обраних технологій (HTML, CSS, JS, BOOTSTRAP, PHP, AJAX), були використані сучасні підходи та принципи веб-розробки, щоб забезпечити зручну навігацію та ефективне відображення інформації. Нижче наведено опис структури сторінок та функціоналу, що реалізовано на основі вищезазначених технологій.

*Головна сторінка.* Головна сторінка є вітриною вашого інтернет-магазину. На ній розміщена загальна інформація про акції, новини та популярні товари. Використовується HTML, CSS та JavaScript для створення динамічних блоків та слайдерів. На рис. 3.2 представлено приклад реалізації сторінки Index.php на *back-end*, а на рис. 3.3 – *Front-end*.

```

index.php
index.php
25 </head>
26 <body>
27 <div class="wait overlay">
28 <div class="loader"></div>
29 </div>
30 <div class="navbar navbar-inverse navbar-fixed-top">
31 <div class="container-fluid">
32 <div class="navbar-header">
33 <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#collapse" aria-expanded="fal
34 <span class="sr-only">Навігація</span>
35 <span class="icon-bar"></span>
36 <span class="icon-bar"></span>
37 <span class="icon-bar"></span>
38 </button>
39 <a href="home.html" style="margin-left: 20px;" class="navbar-brand">Innovate</a>
40 </div>
41 <div class="collapse navbar-collapse" id="collapse">
42 <ul class="nav navbar-nav">
43 <li><a href="/home.html"><span class="glyphicon glyphicon-home"></span> Домашня</a></li>
44 <li><a href="index.php"><span class="glyphicon glyphicon-modal-window"></span> Продукція</a></li>
45 </ul>
46 <form class="navbar-form navbar-left">
47 <div class="form-group">
48 <input type="text" class="form-control" placeholder="Пошук" id="search">
49 </div>
50 <button type="submit" class="btn btn-primary" id="search_btn"><span class="glyphicon glyphicon-search"></span></button>
51 </form>
52 <ul class="nav navbar-nav navbar-right">
53 <li><a href="#" class="dropdown-toggle" data-toggle="dropdown"><span class="glyphicon glyphicon-shopping-cart"></span></a>
54 <div class="dropdown-menu" style="width:400px;">
55 <div class="panel panel-success">
56 <div class="panel-heading">
57 <div class="row">
58 <div class="col-md-3">№</div>
59 <div class="col-md-3">Зображення</div>
60 <div class="col-md-3">Назва</div>
61 <div class="col-md-3">Вартість <?php echo CURRENCY;></div>

```

Рисунок 3.2 – Приклад реалізації сторінки Index.php на *back-end*

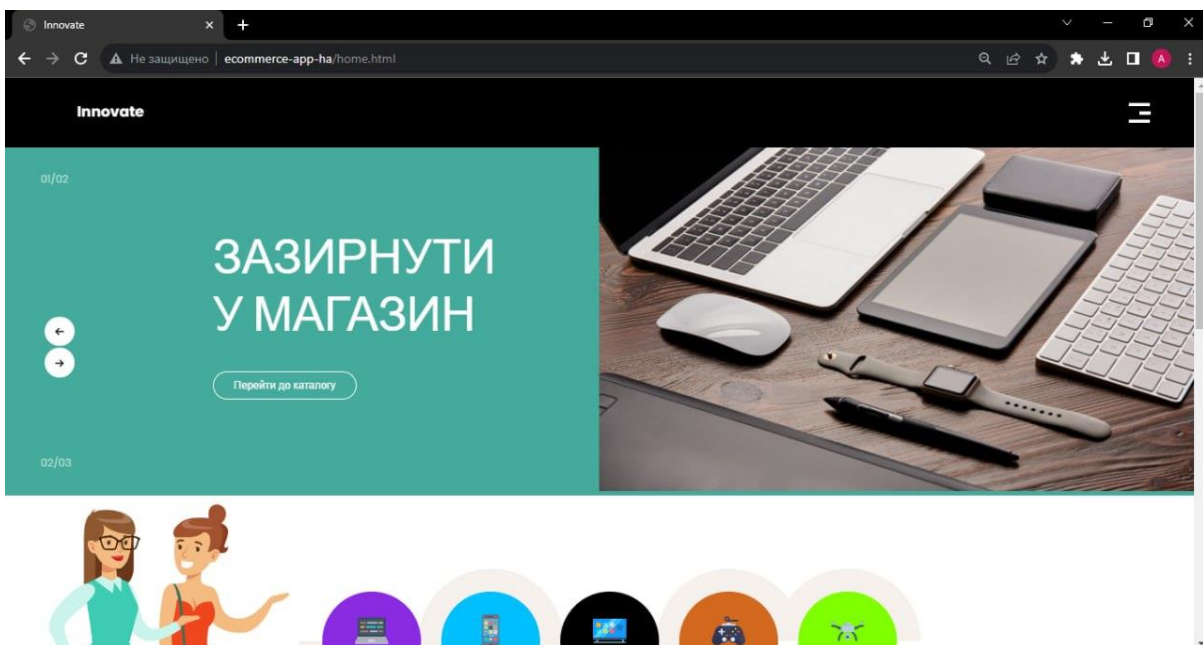


Рисунок 3.3 – Приклад реалізації сторінки індекса на *Front-end*

1. Сторінки категорій товарів (*category.php*). Кожна категорія товарів має свою сторінку, де товари групуються за певними



характеристиками. Використовуються HTML та PHP для відображення списку товарів та фільтрації (рис. 3.4 і рис.3.5).

```

45
46
47 <!-- Modal -->
48 <div class="modal fade" id="add_category_modal" tabindex="-1" role="dialog" aria-labelledby="exam
49 <div class="modal-dialog" role="document">
50 <div class="modal-content">
51 <div class="modal-header">
52 <h5 class="modal-title" id="exampleModalLabel">Оновити</h5>
53 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
54 <span aria-hidden="true">&times;</span>
55 </button>
56 </div>
57 <div class="modal-body">
58 <form id="add-category-form" enctype="multipart/form-data">
59 <div class="row">
60 <div class="col-12">
61 <div class="form-group">
62 <label>Обрати категорію</label>
63 <input type="text" name="cat_title" class="form-control" placeholder="Enter Brand
64 </div>
65 </div>
66 <input type="hidden" name="add_category" value="1">
67 <div class="col-12">
68 <button type="button" class="btn btn-primary add-category">Додати</button>
69 </div>
70 </div>
71 </form>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 <!-- Modal -->
78
79 <!-- Edit category Modal -->
80 <div class="modal fade" id="edit_category_modal" tabindex="-1" role="dialog" aria-labelledby="exa
81 <div class="modal-dialog" role="document">

```

Рисунок 3.4 – Реалізація сторінки категорій

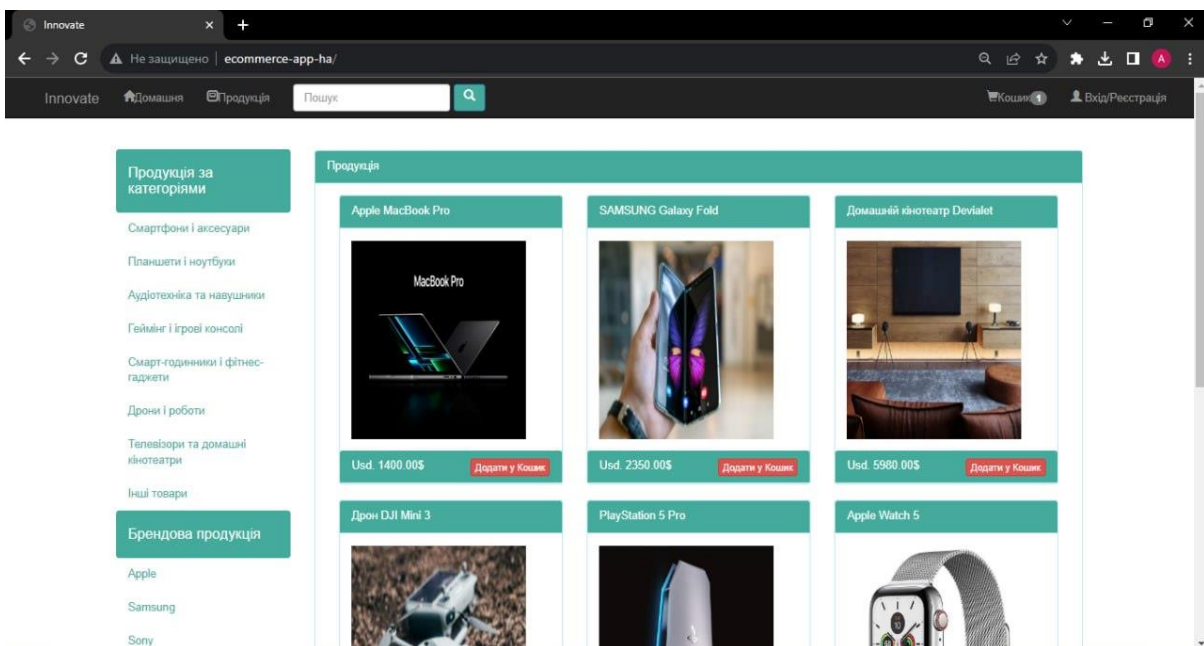
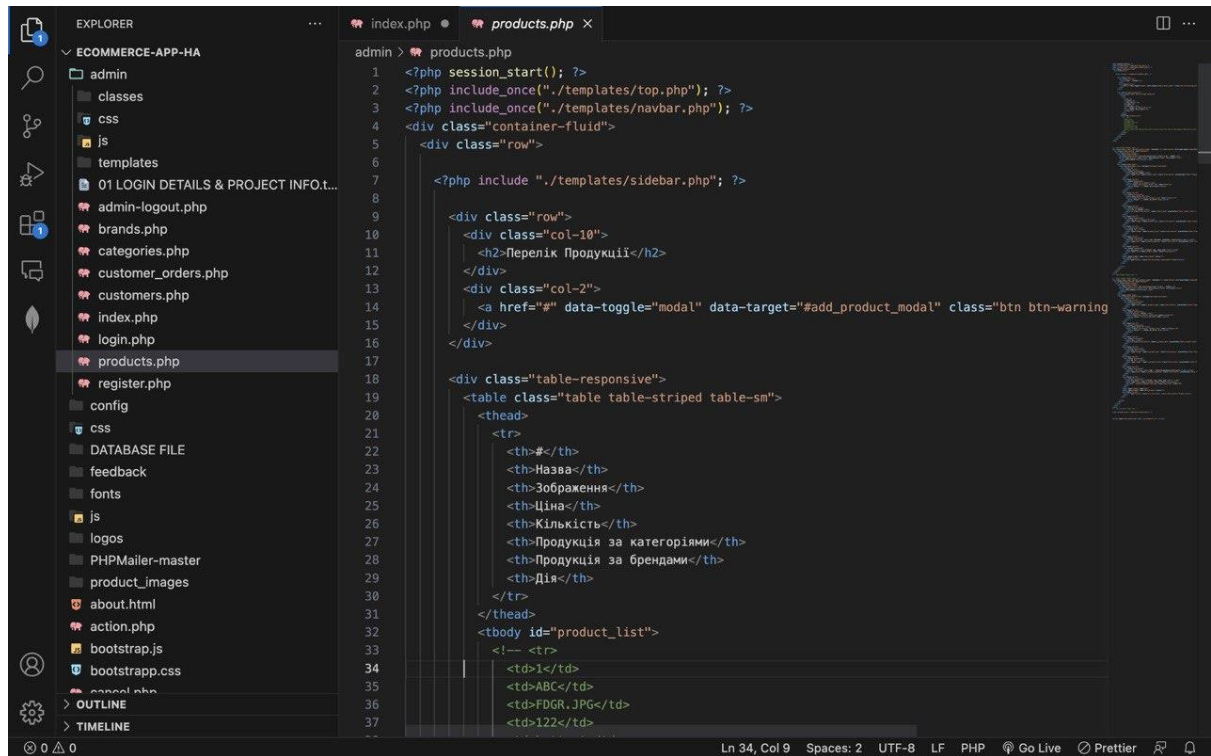


Рисунок 3.5 – Приклад вигляду на *Front-end*

2. *Сторінки товарів (product.php) зображена на рис. 3.6. Кожний товар має свою окрему сторінку з докладною інформацією, фотографіями та кнопкою "Додати до кошика". Використовується HTML, PHP і JavaScript для відображення даних товару та взаємодії з користувачем.*



```

1 <?php session_start(); ?>
2 <?php include_once("../templates/top.php"); ?>
3 <?php include_once("../templates/navbar.php"); ?>
4 <div class="container-fluid">
5   <div class="row">
6
7     <?php include "../templates/sidebar.php"; ?>
8
9     <div class="row">
10      <div class="col-10">
11        <h2>Непенік Продукції</h2>
12      </div>
13      <div class="col-2">
14        <a href="#" data-toggle="modal" data-target="#add_product_modal" class="btn btn-warning">
15        </div>
16      </div>
17
18    <div class="table-responsive">
19      <table class="table table-striped table-sm">
20        <thead>
21          <tr>
22            <th>#</th>
23            <th>Назва</th>
24            <th>Зображення</th>
25            <th>Ціна</th>
26            <th>Кількість</th>
27            <th>Продукція за категоріями</th>
28            <th>Продукція за брендами</th>
29            <th>Дія</th>
30          </tr>
31        </thead>
32        <tbody id="product_list">
33          <!-- <tr>
34            <td>1</td>
35            <td>ABC</td>
36            <td>FDGR.JPG</td>
37            <td>122</td>

```

Рисунок. 3.6 – Програмна реалізація сторінки товару

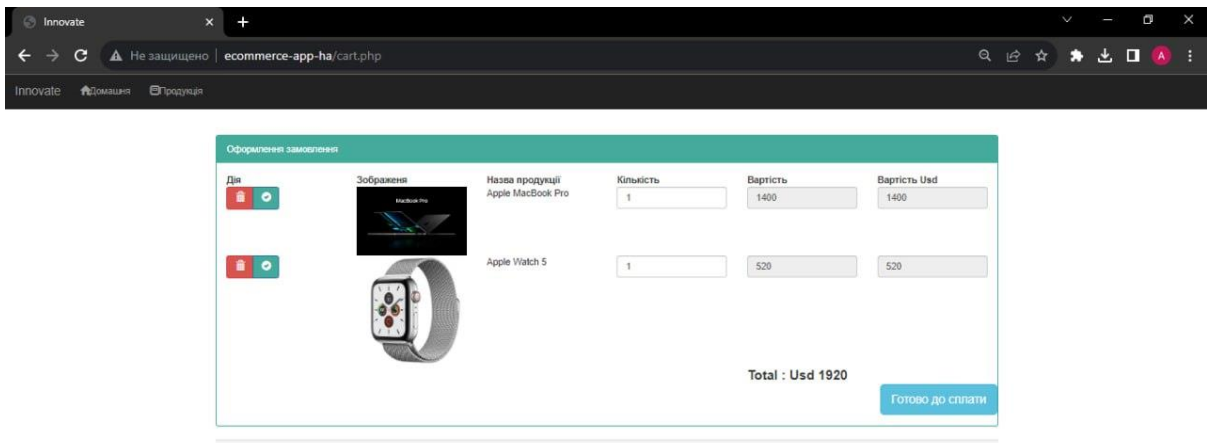
3. *Сторінка кошика (cart.php) зображена на рис. 3.7. та рис. 3.8. На цій сторінці користувач може переглянути вміст свого кошика, змінити кількість товарів та оформити замовлення. Використовується HTML, PHP і JavaScript для взаємодії з кошиком користувача.*

```

1  <?php
2
3  require "config/constants.php";
4
5  ?>
6  <!DOCTYPE html>
7  <html>
8
9      <head>
10         <meta charset="UTF-8">
11         <title>Innovate</title>
12         <link rel="stylesheet" href="css/bootstrap.min.css"/>
13         <script src="js/jquery2.js"></script>
14         <script src="js/bootstrap.min.js"></script>
15         <script src="main.js"></script>
16         <link rel="stylesheet" type="text/css" href="style.css"/>
17     </head>
18     <body>
19         <div class="wait overlay">
20             <div class="loader"></div>
21         </div>
22         <div class="navbar navbar-inverse navbar-fixed-top">
23             <div class="container-fluid">
24                 <div class="navbar-header">
25                     <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#collapse" aria-expanded="false">
26                         <span class="sr-only">Навігація</span>
27                         <span class="icon-bar"></span>
28                         <span class="icon-bar"></span>
29                         <span class="icon-bar"></span>
30                     </button>
31                     <a href="home.html" class="navbar-brand">Innovate</a>
32                 </div>
33                 <div class="collapse navbar-collapse" id="collapse">
34                     <ul class="nav navbar-nav">
35                         <li><a href="index.php"><span class="glyphicon glyphicon-home"></span> Домашня</a></li>
36                         <li><a href="index.php"><span class="glyphicon glyphicon-modal-window"></span> Продукція</a></li>
37                     </ul>
38                 </div>
39             </div>
40         </div>

```

Рисунок 3.7 – Програмна реалізація кошику

Рисунок 3.8 – Приклад реалізації кошика на *Front-end*

4. Сторінка оформлення замовлення (*customer\_order.php*) зображена на рис.3.9. Тут користувач вводить свої дані для оформлення замовлення та вибирає метод оплати та доставки. Використовуються HTML, PHP та AJAX для валідації та обробки даних.

```

31     </ul>
32 </div>
33 </div>
34 <p><br/></p>
35 <p><br/></p>
36 <p><br/></p>
37 <div class="container-fluid">
38
39     <div class="row">
40         <div class="col-md-2"></div>
41         <div class="col-md-8">
42             <div class="panel panel-default">
43                 <div class="panel-heading"></div>
44                 <div class="panel-body">
45                     <h1>Деталі замовлення</h1>
46                     <hr/>
47                     <?php
48                         include_once("db.php");
49                         $user_id = $_SESSION["uid"];
50                         $orders_list = "SELECT o.order_id,o.user_id,o.product_id,o.qty,o.trx_id,o.p_status,p.product_title,p.product
51                         $query = mysqli_query($con,$orders_list);
52                         if (mysqli_num_rows($query) > 0) {
53                             while ($row=mysqli_fetch_array($query)) {
54                                 ?>
55                                     <div class="row">
56                                         <div class="col-md-6">
57                                             
59                                         <div class="col-md-6">
60                                             <table>
61                                                 <tr><td>Назва продукції</td><td><b><?php echo $row["product_title"]; ?></b></td></
62                                                 <tr><td>Вартість</td><td><b><?php echo CURRENCY." ".$row["product_price"]; ?></b></
63                                                 <tr><td>Кількість</td><td><b><?php echo $row["qty"]; ?></b></td></tr>
64                                                 <tr><td>ID транзакції</td><td><b><?php echo $row["trx_id"]; ?></b></td></tr>
65                                             </table>
66                                         </div>
67                                     </div>

```

Рисунок 3.9 – Приклад програмної реалізації сторінки замовлення

### 3.2.2. Меню та навігація

Меню та навігація є важливою частиною дизайну веб-сайту та можуть значно покращити користувацький досвід.

Реалізація меню та навігації складається з основних конструктивних елементів:

#### 1. Верхнє меню (navbar):

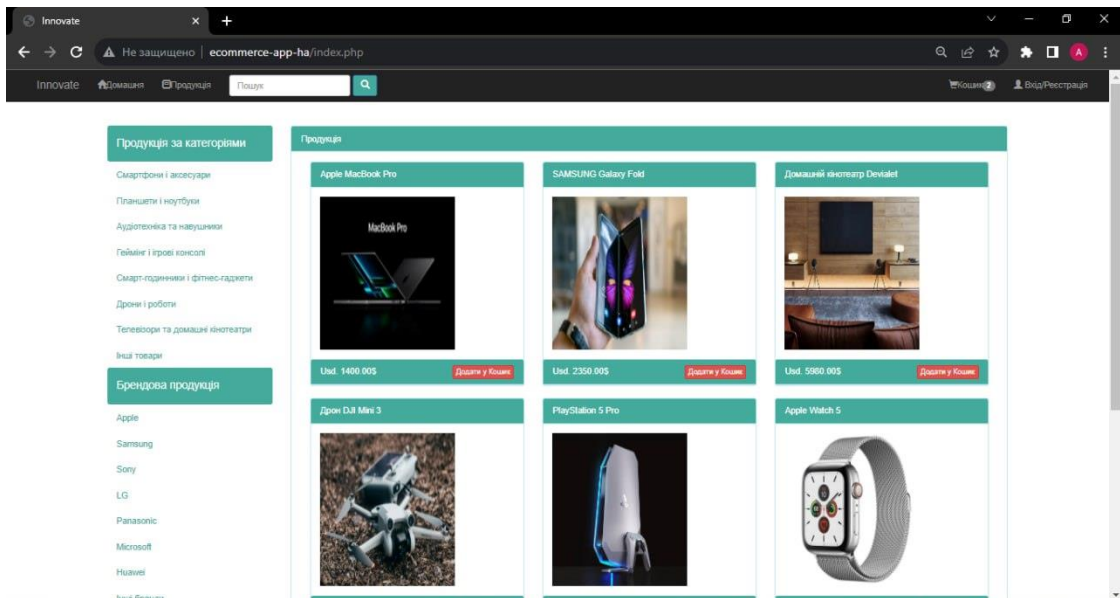


Рисунок 3.10 – Сторінка з товарами

- Верхнє меню (navbar) розташоване вгорі сторінки, його можна побачити на рис. 3.10. та рис. 3.11, і містить логотип сайту, поле пошуку та елементи навігації.

- Використовується клас `navbar` для створення темного фіксованого меню.

- Логотип "Innovate" розташований зліва і має посилання, яке веде на головну сторінку сайту.

- Поле пошуку знаходиться в центрі та має клас `form-control` для введення тексту.

- Справа маємо розділи "Вихід" або "Вхід", відповідно до статусу користувача. При кліці на це посилання відбувається вхід або вихід користувача.

```

1 <nav class="navbar navbar-dark fixed-top bg-dark flex-md-nowrap p-0 shadow">
2 <a class="navbar-brand col-sm-3 col-md-2 mr-0" href=".." index.php">Innovate</a>
3 <input class="form-control form-control-dark w-100" type="text" placeholder="Search" aria-label="
4 <ul class="navbar-nav px-3">
5 <li class="nav-item text-nowrap">
6 <?php
7     if (isset($_SESSION['admin_id'])) {
8         >?
9         <a class="nav-link" href=".." admin/admin-logout.php">Вихід</a>
10        <?php
11    }else{
12        $uriAr = explode("/", $_SERVER['REQUEST_URI']);
13        $page = end($uriAr);
14        if ($page == "login.php") {
15            >?
16            <a class="nav-link" href=".." admin/register.php">Реєстрація</a>
17            <?php
18        }else{
19            >?
20            <a class="nav-link" href=".." admin/login.php">Вхід</a>
21            <?php
22        }
23    }
24 </li>
25 </ul>
26 </nav>

```

Рисунок 3.11 – Програмна реалізація меню navbar

2. Бічне меню (*sidebar*) представлене на рис. 3.12.

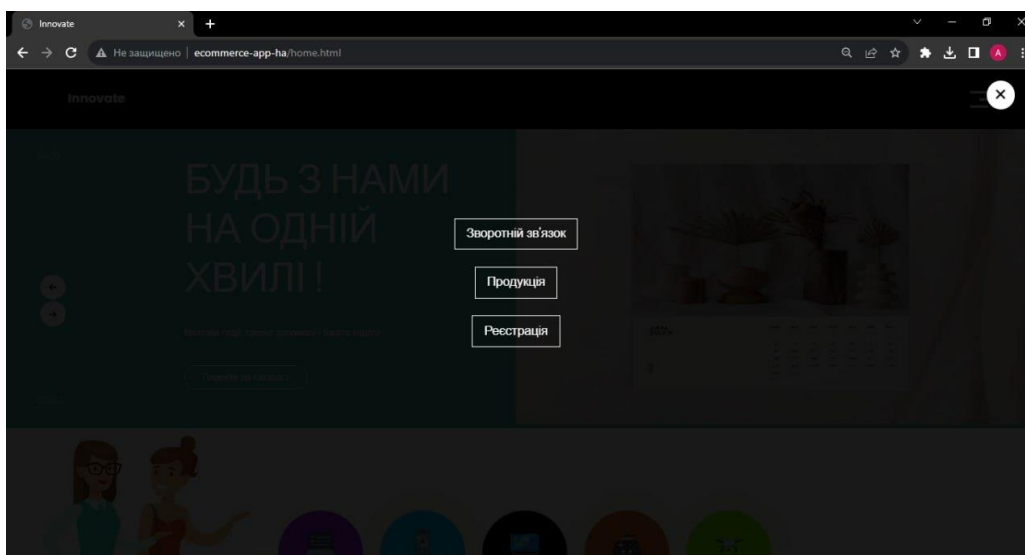


Рисунок 3.12 – Бічне меню(*sidebar*)

Бічне меню (*sidebar*), зображене на рис. 3.12. та на рис. 3.13, розташоване зліва на сторінці і містить пункти навігації.

```

1 <nav class="col-md-2 d-none d-md-block bg-light sidebar">
2 <div class="sidebar-sticky">
3 <ul class="nav flex-column">
4
5 <?php
6
7
8 $uri = $_SERVER['REQUEST_URI'];
9 $uriAr = explode("/", $uri);
10 $page = end($uriAr);
11
12 ?>
13
14
15 <li class="nav-item">
16 <a class="nav-link <?php echo ($page == '' || $page == 'index.php') ? 'active' : '';"
17 <span data-feather="home"></span>
18 Панель приладів <span class="sr-only">(current)</span>
19 </a>
20 </li>
21 <li class="nav-item">
22 <a class="nav-link <?php echo ($page == 'customer_orders.php') ? 'active' : '';" ?> h
23 <span data-feather="clipboard"></span>
24 Замовлення
25 </a>
26 </li>
27 <li class="nav-item">
28 <a class="nav-link <?php echo ($page == 'products.php') ? 'active' : '';" ?> href="pr
29 <span data-feather="shopping-cart"></span>
30 Продукція
31 </a>
32 </li>
33 <li class="nav-item">
34 <a class="nav-link <?php echo ($page == 'brands.php') ? 'active' : '';" ?> href="bran
35 <span data-feather="box"></span>
36 Брендова продукція
37 </a>

```

Рисунок 3.13 – Програмна реалізація sidebar

Використовується клас `bg-light` для задання світлого фону бокового меню.

Пункти навігації мають активний стиль (клас `active`), який виділяє поточну сторінку.

### 3.3. Розробка функціоналу

Розробка функціоналу є критично важливим етапом у створенні веб-сайту, особливо для інтернет-магазину. Основними перевагами при розробці функціоналу є:

1. *Покращення користувацького досвіду.* Розробка функціоналу дозволяє покращити зручність та ефективність користувацького досвіду. Впровадження нових функцій може робити сайт більш привабливим та корисним для відвідувачів.

2. *Збільшення конверсії.* Деякі функціональні можливості, такі як швидка реєстрація, спрощена оплата, додавання товарів до кошика тощо, допомагають збільшити конверсію, тобто перетворення відвідувачів у покупців або клієнтів.

3. *Підвищення продуктивності.* Веб-сайти з розвинутим функціоналом можуть спрощувати рутинні завдання та процеси, що підвищує продуктивність роботи та зменшує ризик помилок.

4. *Розширення можливостей.* Додавання нового функціоналу розширює можливості веб-сайту. Це дозволяє вам впроваджувати нові ідеї та інновації, які можуть забезпечити конкурентну перевагу.

5. *Безпека та захист даних.* Важливо розробляти функціонал для забезпечення безпеки користувачів і захисту їхніх даних. Це включає в себе захист від несанкціонованого доступу, шифрування даних та регулярні оновлення з метою виявлення та усунення потенційних загроз.

6. *Підтримка бізнес-процесів.* Функціональність дозволяє вам ефективно впроваджувати та підтримувати бізнес-процеси, такі як обробка замовлень, керування запасами, взаємодія з клієнтами та інші операції, необхідні для успішної діяльності інтернет-магазину.

Отже, розробка функціоналу є важливою складовою будь-якого веб-сайту, особливо для інтернет-магазинів, і вона спрямована на поліпшення користувацького досвіду, збільшення конверсії та забезпечення безпеки та ефективності бізнес-процесів.

### **3.3.1. Реєстрація та авторизація користувачів**

Реєстрація та авторизація користувачів є важливими функціями веб-сайту, оскільки вони дозволяють користувачам створювати облікові записи, входити на сайт і виконувати дії, які доступні лише авторизованим



користувачам. Нижче наведено детальний опис процесу реєстрації та авторизації, а також приклади сторінок для кожного з цих процесів.

1. Реєстрація користувача зображена на рис.3.14. та 3.15.

The screenshot shows a web browser window with the URL `ecommerce-app-ha/customer_registration.php?register=1`. The page displays a registration form with the following fields:

- Ім'я (Name)
- Фамілія (Surname)
- Email
- Пароль (Password)
- Підтвердити пароль (Confirm Password)
- Контактний телефон (Contact Phone)
- Основна Адреса (Main Address)
- Додаткова Адреса (Additional Address)

A green button labeled "Зареєструватися" (Register) is positioned at the bottom of the form.

Рисунок 3.14 – Реалізація форми реєстрації на веб-сайті

```

37 <div class="col-md-2"></div>
38 <div class="col-md-8" id="signup_msg">
39 <!--Alert from signup form-->
40 </div>
41 <div class="col-md-2"></div>
42 </div>
43 <div class="row">
44 <div class="col-md-2"></div>
45 <div class="col-md-8">
46 <div class="panel panel-primary">
47 <div class="panel-heading text-center">Форма реєстрації клієнта</div>
48 <div class="panel-body">
49
50 <form id="signup_form" onsubmit="return false">
51 <div class="row">
52 <div class="col-md-6">
53 <label for="f_name">Ім'я</label>
54 <input type="text" id="f_name" name="f_name" class="form-control" required>
55 </div>
56 <div class="col-md-6">
57 <label for="f_name">Фамілія</label>
58 <input type="text" id="l_name" name="l_name" class="form-control" required>
59 </div>
60 </div>
61 <div class="row">
62 <div class="col-md-12">
63 <label for="email">Email</label>
64 <input type="text" id="email" name="email" class="form-control" required>
65 </div>
66 </div>
67 <div class="row">
68 <div class="col-md-12">
69 <label for="password">Пароль</label>
70 <input type="password" id="password" name="password" class="form-control" required>
71 </div>
72 </div>
73 <div class="row">

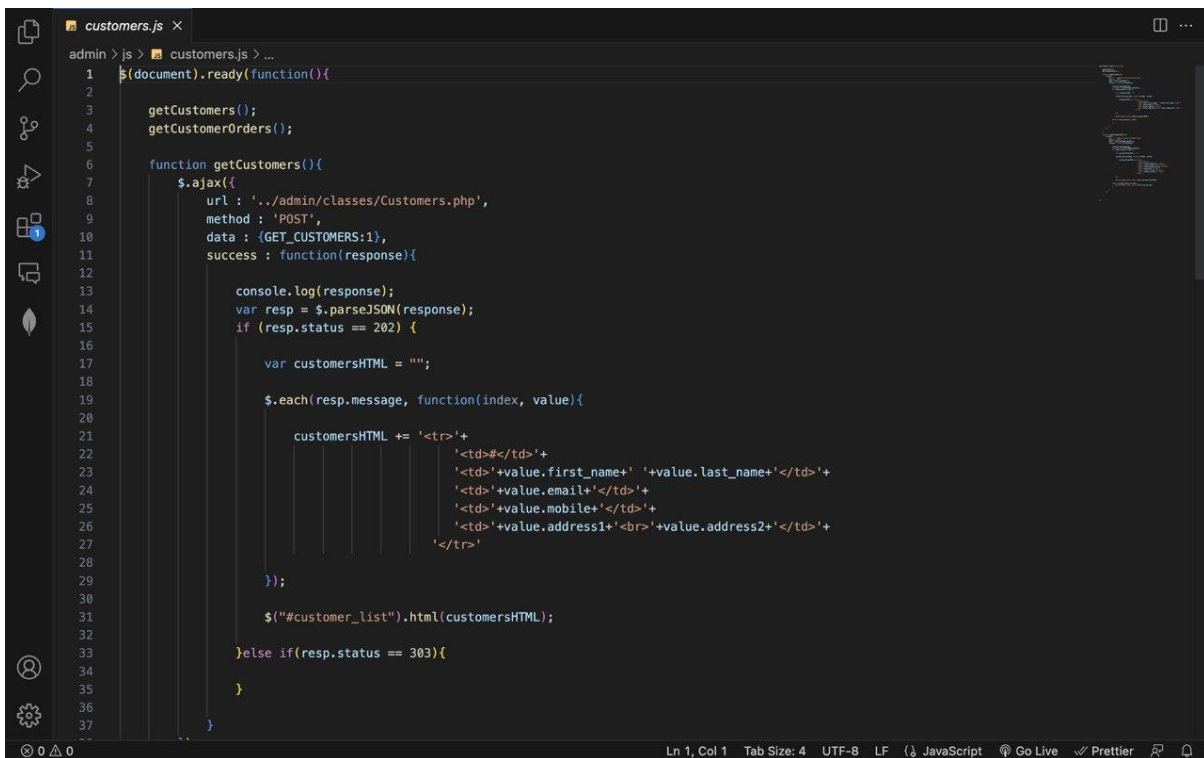
```

Рисунок 3.15 – Програмна реалізація форми реєстрації

Створення облікового запису для нового користувача передбачає такі основні кроки:

- Форма реєстрації. Користувач заповнює спеціальну форму реєстрації на веб-сайті.

- Збереження інформації наведено на рис. 3.16. Введені користувачем дані, такі як ім'я, фамілія, електронна пошта, пароль і інші, зберігаються в базі даних після перевірки їх на валідність.



```
customers.js X
admin > js > customers.js > ...
1 $(document).ready(function(){
2
3   getCustomers();
4   getCustomerOrders();
5
6   function getCustomers(){
7     $.ajax({
8       url : '../admin/classes/Customers.php',
9       method : 'POST',
10      data : {GET_CUSTOMERS:1},
11      success : function(response){
12
13        console.log(response);
14        var resp = $.parseJSON(response);
15        if (resp.status == 202) {
16
17          var customersHTML = "";
18
19          $.each(resp.message, function(index, value){
20
21            customersHTML += '<tr>'+
22              '<td>#</td>'+
23              '<td>'+value.first_name+' '+value.last_name+'</td>'+
24              '<td>'+value.email+'</td>'+
25              '<td>'+value.mobile+'</td>'+
26              '<td>'+value.address1+'<br>'+value.address2+'</td>'+
27              '</tr>';
28
29          });
30
31          $("#customer_list").html(customersHTML);
32
33        }else if(resp.status == 303){
34
35        }
36
37      }
38    });
39  }
40 }
```

Рисунок 3.16 – Функція отримання даних користувача з бази даних

- Перевірка унікальності електронної пошти. В системі перевіряється, чи є введена електронна пошта унікальною (не зареєстрованою раніше).
- Захист пароля зображений на рис. 3.17. Пароль зазвичай хешується (за допомогою, наприклад, MD5 або SHA-256), перш ніж зберігатися в базі даних, для забезпечення безпеки.

```

login.php
1 <?php
2 include "db.php";
3
4 session_start();
5
6 #Login script is begin here.
7 #If user given credential matches successfully with the data available in database then we will echo string login_success
8 #login_success string will go back to called Anonymous funtion $("#login").click()
9 if(isset($_POST["email"]) && isset($_POST["password"])){
10     $email = mysqli_real_escape_string($con,$_POST["email"]);
11     $password = md5($_POST["password"]);
12     $sql = "SELECT * FROM user_info WHERE email = '$email' AND password = '$password'";
13     $run_query = mysqli_query($con,$sql);
14     $count = mysqli_num_rows($run_query);
15     //if user record is available in database then $count will be equal to 1
16     if($count == 1){
17         $row = mysqli_fetch_array($run_query);
18         $_SESSION["uid"] = $row["user_id"];
19         $_SESSION["name"] = $row["first_name"];
20         $ip_add = getenv("REMOTE_ADDR");
21         //we have created a cookie in login_form.php page so if that cookie is available means user is not login
22         if (isset($_COOKIE["product_list"])) {
23             $p_list = stripslashes($_COOKIE["product_list"]);
24             //here we are decoding stored json product list cookie to normal array
25             $product_list = json_decode($p_list,true);
26             for ($i=0; $i < count($product_list); $i++) {
27                 //After getting user id from database here we are checking user cart item if there is already product is listed or
28                 $verify_cart = "SELECT id FROM cart WHERE user_id = $_SESSION[uid] AND p_id = ".$product_list[$i];
29                 $result = mysqli_query($con,$verify_cart);
30                 if(mysqli_num_rows($result) < 1){
31                     //if user is adding first time product into cart we will update user_id into database table with valid id
32                     $update_cart = "UPDATE cart SET user_id = '$_SESSION[uid]' WHERE ip_add = '$ip_add' AND user_id = -1";
33                     mysqli_query($con,$update_cart);
34                 }else{
35                     //if already that product is available into database table we will delete that record
36                     $delete_existing_product = "DELETE FROM cart WHERE user_id = -1 AND ip_add = '$ip_add' AND p_id = ".$product_li
37                     mysqli_query($con,$delete_existing_product);

```

Рисунок 3.17 – Валідація користувача

2. *Авторизація користувача.* Користувачі, які вже зареєстровані на веб-сайті, можуть увійти за допомогою свого облікового запису, це зображено на рис.3.19.

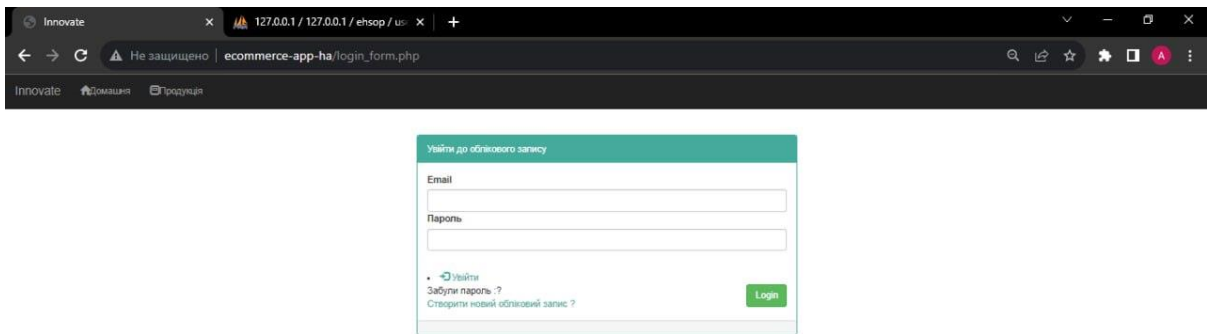


Рисунок 3.19 – Приклад форми входу для користувача

Процес авторизації включає такі етапи:

- Форма авторизації. Користувач вводить свою електронну пошту та пароль у форму на веб-сайті.
- Перевірка даних. Система перевіряє, чи існують введені дані в базі даних і чи вони вірні.
- Створення сесії наведено на рис. 3.20. Якщо авторизація успішна, сервер створює сесію для користувача, яка дозволяє відстежувати його вхід на веб-сайт.

```

login.php
1 <?php
2 include "db.php";
3
4 session_start();
5
6 #Login script is begin here
7 #If user given credential matches successfully with the data available in database then we will echo string login_success
8 #login_success string will go back to called Anonymous funtion $("#login").click()
9 if(isset($_POST["email"]) && isset($_POST["password"])){
10     $email = mysqli_real_escape_string($con,$_POST["email"]);
11     $password = md5($_POST["password"]);
12     $sql = "SELECT * FROM user_info WHERE email = '$email' AND password = '$password'";
13     $run_query = mysqli_query($con,$sql);
14     $count = mysqli_num_rows($run_query);
15     //if user record is available in database then $count will be equal to 1
16     if($count == 1){
17         $row = mysqli_fetch_array($run_query);
18         $_SESSION["uid"] = $row["user_id"];
19         $_SESSION["name"] = $row["first_name"];
20         $ip_add = getenv("REMOTE_ADDR");
21         //we have created a cookie in login_form.php page so if that cookie is available means user is not login
22         if (isset($_COOKIE["product_list"])) {
23             $p_list = stripslashes($_COOKIE["product_list"]);
24             //here we are decoding stored json product list cookie to normal array
25             $product_list = json_decode($p_list,true);
26             for ($i=0; $i < count($product_list); $i++) {
27                 //After getting user id from database here we are checking user cart item if there is already product is listed or
28                 $verify_cart = "SELECT id FROM cart WHERE user_id = $_SESSION[uid] AND p_id = ".$product_list[$i];
29                 $result = mysqli_query($con,$verify_cart);
30                 if(mysqli_num_rows($result) < 1){
31                     //if user is adding first time product into cart we will update user_id into database table with valid id
32                     $update_cart = "UPDATE cart SET user_id = '$_SESSION[uid]' WHERE ip_add = '$ip_add' AND user_id = -1";
33                     mysqli_query($con,$update_cart);
34                 }else{
35                     //if already that product is available into database table we will delete that record
36                     $delete_existing_product = "DELETE FROM cart WHERE user_id = -1 AND ip_add = '$ip_add' AND p_id = ".$product_list[$i];
37                     mysqli_query($con,$delete_existing_product);

```

Рисунок 3.20 – Системна логіка перевірки даних користувача

Наведені процеси необхідні для забезпечення безпеки даних користувачів та для надання доступу до особистого кабінету на веб-сайті. Вони дозволяють веб-сайту ідентифікувати користувачів і взаємодіяти з ними відповідно до їхніх прав та ролей.

### 3.3.2. Додавання товарів до кошика

Додавання товарів до кошика – це важлива функція для будь-якого інтернет-магазину. Нижче наведено детальний опис логіки і коду для цього процесу.

#### *Логіка додавання товарів до кошика*

1. Користувач переглядає товари на веб-сайті та обирає певний товар для покупки, як зображено на рис. 3.21.

2. Після вибору товару користувач натискає кнопку "Додати у кошик"(рис. 3.21).

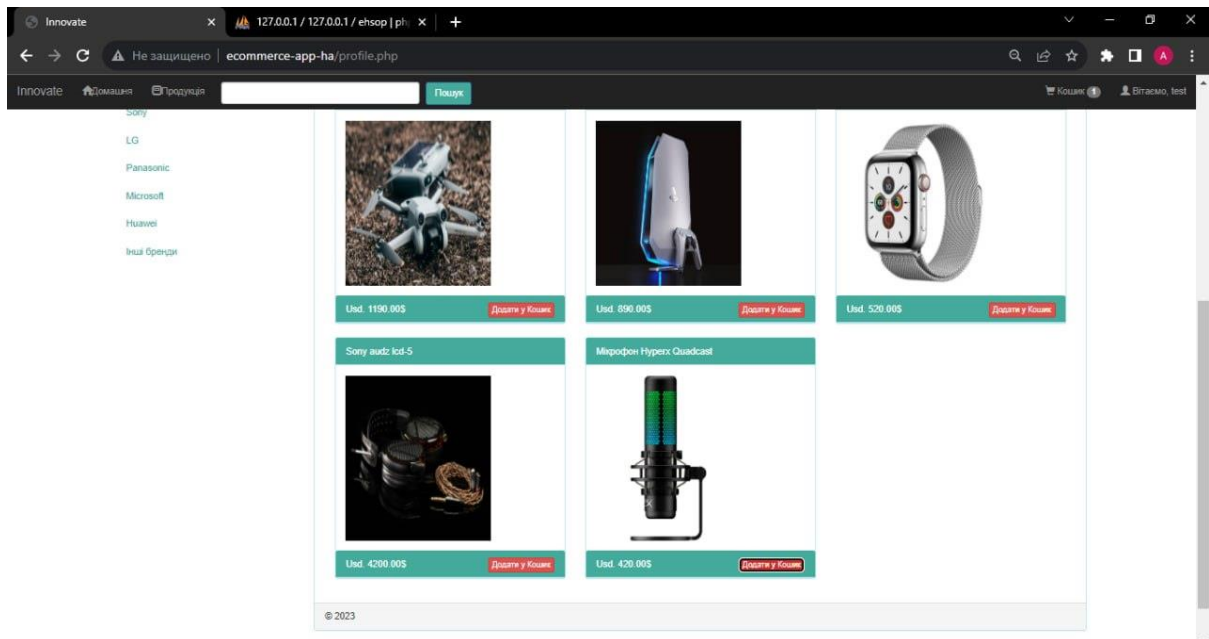


Рисунок 3.21 – Обрання товару і додавання до кошику

3. Інформація про доданий товар (назва, ціна, зображення тощо) додається до кошика користувача.

4. Якщо користувач увійшов в систему (має сеанс), інформація про товар зберігається в базі даних та прив'язується до його облікового запису.

5. Якщо користувач не увійшов в систему (гость), інформація про товар зберігається на основі його IP-адреси, і обліковий запис створюється для нього, зображено на рис. 3.22.

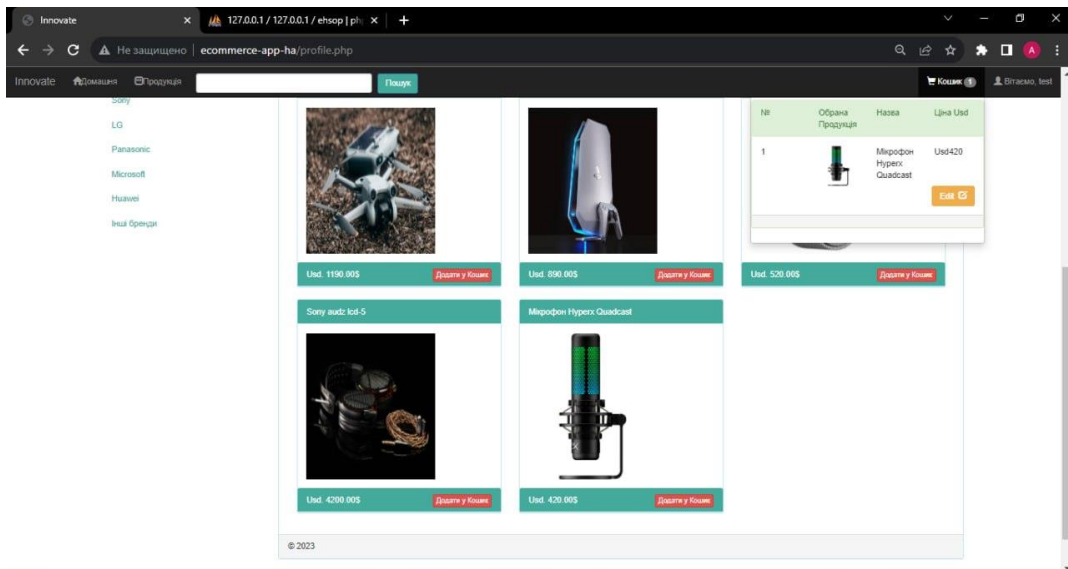


Рисунок 3.22 – Приклад доданого товару до кошику для зареєстрованого користувача test

6. У випадку успішного додавання товару до кошика, користувач отримує повідомлення про успішне додавання товару.

### *Приклад PHP-коду*

У PHP-кодi логiка реалiзована таким чином:

1. Перевiряється, чи користувач додав товар до кошика за допомогою кнопки "Додати у кошик", зображено на рис. 3.23.

```

165
166
167 if (isset($_POST["addToCart"])) {
168
169     $p_id = $_POST["prodId"];
170
171
172
173     if (isset($_SESSION["uid"])) {
174
175         $user_id = $_SESSION["uid"];
176
177         $sql = "SELECT * FROM cart WHERE p_id = '$p_id' AND user_id = '$user_id'";
178         $run_query = mysqli_query($con, $sql);
179         $count = mysqli_num_rows($run_query);
180         if ($count > 0) {
181             echo "
182                 <div class='alert alert-warning'>
183                     <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
184                     <b>Додано, додати ще щось?</b>
185                 </div>
186             "; //not in video
187         } else {
188             $sql = "INSERT INTO `cart`
189                 (`p_id`, `ip_add`, `user_id`, `qty`)
190                 VALUES ('$p_id','$ip_add','$user_id','1')";
191             if (mysqli_query($con, $sql)) {
192                 echo "
193                     <div class='alert alert-success'>
194                         <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
195                         <b>Додано..!</b>
196                     </div>
197                 ";
198             }
199         }
200     } else {
201         $sql = "SELECT id FROM cart WHERE ip_add = '$ip_add' AND p_id = '$p_id' AND user_id = -1";

```

Рисунок 3.23 – Перевірка додавання товару до кошику

2. Якщо користувач увійшов в систему, інформація про товар додається до таблиці "cart" у базі даних, прив'язана до його облікового запису.

3. Якщо користувач не увійшов в систему, інформація про товар зберігається на основі IP-адреси, зображено на рис. 3.24 та рис. 3.25.



```

if (isset($_SESSION["uid"])) {
    // Отримання ID користувача, який увійшов в систему
    $user_id = $_SESSION["uid"];

    // Отримання ID товару, який додається до кошика
    $p_id = $_POST["proId"];

    // Перевірка, чи товар вже існує в кошику користувача
    $sql = "SELECT * FROM cart WHERE p_id = '$p_id' AND user_id = '$user_id'";
    $run_query = mysqli_query($con, $sql);
    $count = mysqli_num_rows($run_query);

    if ($count > 0) {
        // Товар вже є в кошику користувача, виконайте необхідну обробку
    } else {
        // Додавання товару до кошика користувача
        $sql = "INSERT INTO `cart` (`p_id`, `ip_add`, `user_id`, `qty`) VALUES ('$p_id', '$ip_add', '$user_id', '1')";
        if (mysqli_query($con, $sql)) {
            // Отримано успішну відповідь від сервера
        } else {
            // Обробка помилки додавання товару
        }
    }
}
}

```

Рисунок 3.24 – Перевірка користувача

```

if (!isset($_SESSION["uid"])) {
    // Отримання ID товару, який додається до кошика
    $p_id = $_POST["proId"];

    // Перевірка, чи товар вже існує в кошику гостя (за IP-адресою)
    $sql = "SELECT id FROM cart WHERE ip_add = '$ip_add' AND p_id = '$p_id'";
    $query = mysqli_query($con, $sql);

    if (mysqli_num_rows($query) > 0) {
        // Товар вже є в кошику гостя, виконайте необхідну обробку
    } else {
        // Додавання товару до кошика гостя
        $sql = "INSERT INTO `cart` (`p_id`, `ip_add`, `user_id`, `qty`) VALUES ('$p_id', '$ip_add', '$user_id', '1')";
        if (mysqli_query($con, $sql)) {
            // Отримано успішну відповідь від сервера
        } else {
            // Обробка помилки додавання товару
        }
    }
}
}

```

Рисунок 3.25 – Логіка роботи якщо товар додає гість

Також у розробленому додатку є можливість видаляти товари з кошика та оновлювати кількість товарів у кошику за допомогою відповідних запитів.

### 3.3.3. Оформлення замовлення

У розділі оформлення замовлення веб-сайту ми забезпечуємо користувачів можливістю вибору товарів та оплати їх. Цей розділ включає два основних етапи: сторінку оформлення замовлення та обробку успішної оплати. Для цього використовуються PHP та MySQL для взаємодії з базою даних і створення необхідних форм та звітів.

Розглянемо детальніше файл `customer\_order.php` (сторінку оформлення замовлення) зображений на рис. 3.26.

```

customer_order.php
30 <li><a href="index.php"><span class="glyphicon glyphicon-modal-window"></span>Продукція</a></li>
31 </ul>
32 </div>
33 </div>
34 <p><br/></p>
35 <p><br/></p>
36 <p><br/></p>
37 <div class="container-fluid">
38
39 <div class="row">
40 <div class="col-md-2"></div>
41 <div class="col-md-8">
42 <div class="panel panel-default">
43 <div class="panel-heading"></div>
44 <div class="panel-body">
45 <h1>Детали замовлення</h1>
46 <hr/>
47 <?php
48 include_once("db.php");
49 $user_id = $_SESSION["uid"];
50 $orders_list = "SELECT o.order_id,o.user_id,o.product_id,o.qty,o.trx_id,o.p_status,p.product_title,p.produc
51 $query = mysqli_query($con,$orders_list);
52 if (mysqli_num_rows($query) > 0) {
53 while ($row=mysqli_fetch_array($query)) {
54
55 <div class="row">
56 <div class="col-md-6">
57 
59 <div class="col-md-6">
60 <table>
61 <tr><td>Назва продукції</td><td><b><?php echo $row["product_title"]; ?></b></td></tr>
62 <tr><td>Вартість </td><td><b><?php echo CURRENCY." ".$row["product_price"]; ?></b></td></tr>
63 <tr><td>Кількість</td><td><b><?php echo $row["qty"]; ?></b></td></tr>
64 <tr><td>ID транзакції</td><td><b><?php echo $row["trx_id"]; ?></b></td></tr>
65 </table>
66 </div>
67 </div>

```

Рисунок 3.26 – Сторінка оформлення замовлень користувачів

У цьому файлі створюється сторінка оформлення замовлення, на якій користувач може переглянути вибрані ним товари та вказати необхідну інформацію для замовлення.

Алгоритм роботи:

1. Виконується перевірка: чи користувач авторизований (в нашому випадку, за допомогою сесій). Якщо користувач не авторизований, він перенаправляється на сторінку авторизації.

2. Виконується підключення необхідних зовнішніх ресурсів, таких як CSS-стилі та JavaScript-скрипти.

3. У верхній частині сторінки створюється навігаційне меню для переходу на інші сторінки вашого веб-сайту.

4. Наступним кроком йде відображення інформації про товари, які користувач вибрав для замовлення. Цю інформацію «витагується» з бази даних, об'єднуючи дані з таблиці `orders` і `products`.

5. Вивід інформації про товари у вигляді списку, де вказана назва товару, ціна, кількість та інші деталі, зображено на рис. 3.27 та рис. 3.28.

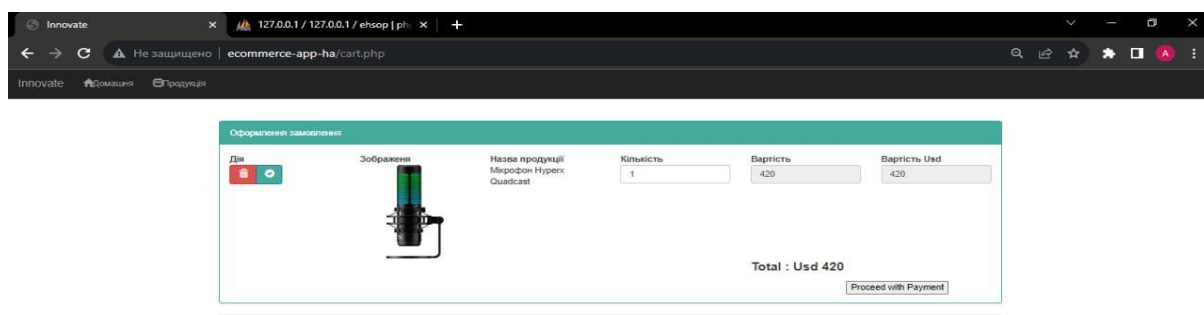
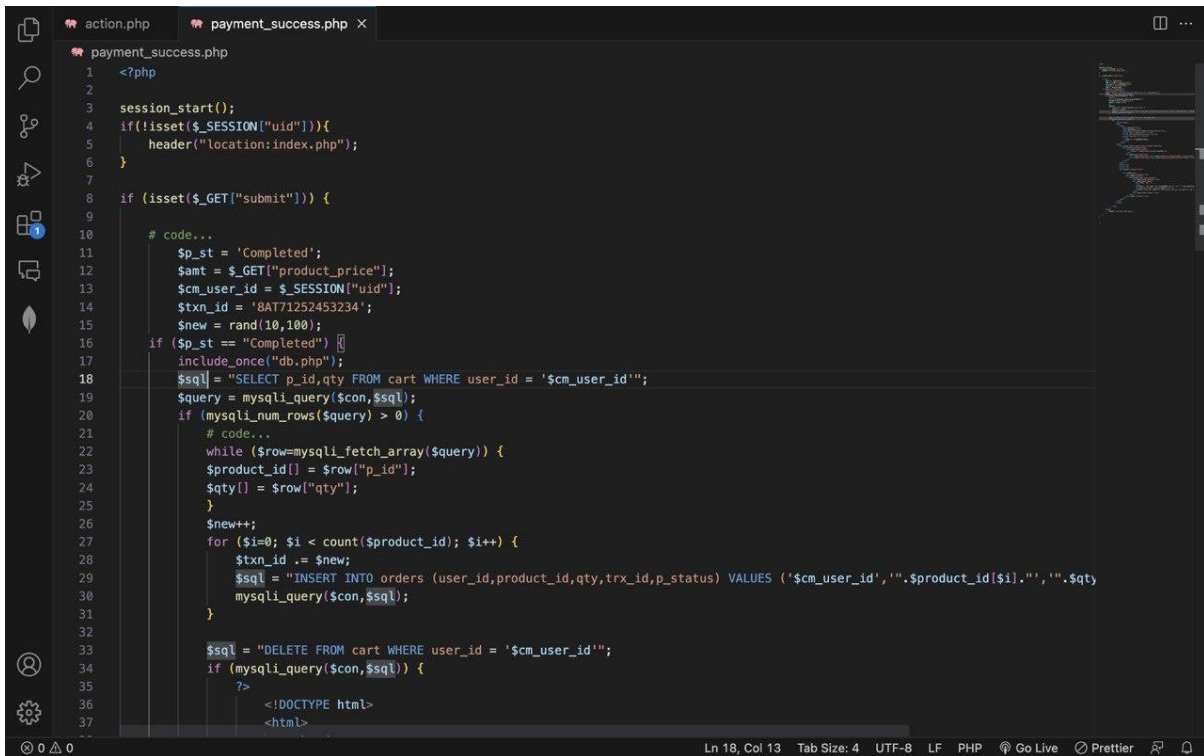


Рисунок 3.27 – Приклад реалізації сторінки замовлення у розробленому додатку

6. На цій сторінці також включаємо кнопку для оплати, яка веде до платіжного шлюзу.

Обробка платежів виконується наступним чином:



```

1 <?php
2
3 session_start();
4 if(!isset($_SESSION["uid"])){
5     header("location:index.php");
6 }
7
8 if (isset($_GET["submit"])) {
9
10     # code...
11     $p_st = 'Completed';
12     $amt = $_GET["product_price"];
13     $cm_user_id = $_SESSION["uid"];
14     $txn_id = 'BAT71252453234';
15     $new = rand(10,100);
16
17     if ($p_st == "Completed") {
18         include_once("db.php");
19         $sql = "SELECT p_id,qty FROM cart WHERE user_id = '$cm_user_id'";
20         $query = mysqli_query($con,$sql);
21         if (mysqli_num_rows($query) > 0) {
22             # code...
23             while ($row=mysqli_fetch_array($query)) {
24                 $product_id[] = $row["p_id"];
25                 $qty[] = $row["qty"];
26             }
27             $new++;
28             for ($i=0; $i < count($product_id); $i++) {
29                 $txn_id .= $new;
30                 $sql = "INSERT INTO orders (user_id,product_id,qty,trx_id,p_status) VALUES ('$cm_user_id','.$product_id[$i].','$qty'";
31                 mysqli_query($con,$sql);
32             }
33             $sql = "DELETE FROM cart WHERE user_id = '$cm_user_id'";
34             if (mysqli_query($con,$sql)) {
35                 ?>
36                 <!DOCTYPE html>
37                 <html>

```

Рисунок 3.28 – Програмна реалізація транзакцій

Цей файл використовується для обробки та підтвердження успішної оплати.

1. Перевіряємо, чи користувач авторизований. Якщо користувач не авторизований, перенаправляємо його на сторінку авторизації.

2. Використовуємо параметри, які передаються через URL, для отримання інформації про успішну оплату, зокрема `txn\_id`, `product\_price`, тощо.

3. Оновлюємо базу даних, змінюючи статус оплати на "Completed", зберігаючи інформацію про транзакцію та виконуючи інші необхідні дії для оформлення замовлення.

4. Після додавання замовлення до бази даних відображаємо користувачу повідомлення про успішну оплату та надаємо інформацію про замовлення, включаючи ідентифікатор транзакції, та перенаправляємо його до списку продуктів, це зображено на рис. 3.29.

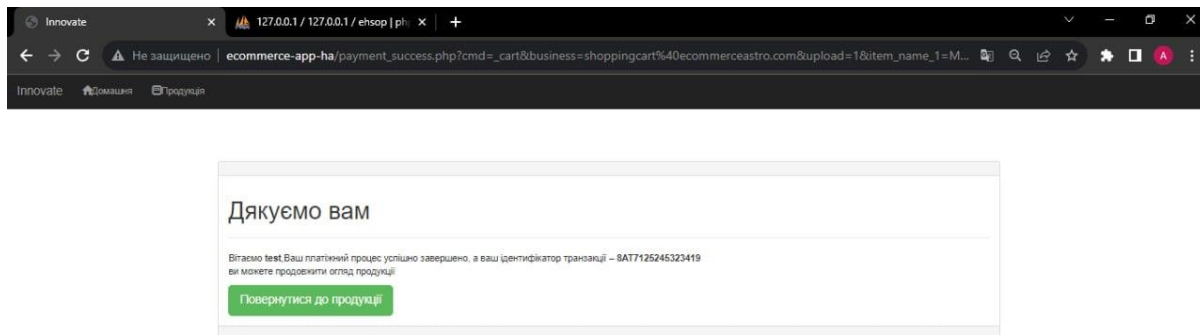


Рисунок. 3.29 – Успішна транзакція

Загалом, ці два файли спільно працюють, щоб забезпечити користувачам можливість оформити та оплатити свої замовлення на веб-сайті.

### 3.4. Оптимізація веб-сайту по його комерціалізації

#### Підвищення швидкості завантаження сторінок

Робота відзначається науковою новизною завдяки розробці та застосуванню нових підходів до оптимізації веб-сайтів інтернет-магазинів. Пропонується низка інноваційних методик, які дозволяють підвищити

продуктивність веб-сайту, враховуючи сучасні тенденції у споживчій поведінці та технологічній динаміці.

1. *Асинхронне завантаження ресурсів.* Однією з ключових методик, що робить розроблений підхід науково новим, є використання асинхронної загрузки статичних ресурсів, таких як стилі, зображення та скрипти. Це дозволяє значно прискорити завантаження сторінок і зменшити час очікування користувачів.
2. *Оптимізація зображень.* Ще однією інноваційною методикою є застосування технік стиснення та використання форматів з високою продуктивністю, таких як WebP, для оптимізації завантаження зображень. Це сприяє швидкому завантаженню сторінок і поліпшує загальний користувацький досвід.

### **Адаптивний дизайн та оптимізація для мобільних пристроїв**

З метою підвищення комерційної ефективності та задоволення потреб користувачів, було розроблено та вдосконалено веб-сайт за допомогою таких науково нових методик:

3. *Адаптивний дизайн.* Однією з ключових інновацій є використання сучасних методик CSS Grid та Flexbox для створення адаптивного дизайну. Цей підхід гарантує оптимальне відображення веб-сайту на різних розмірах екранів, що важливо для користувачів, які використовують мобільні пристрої.
4. *Мобільна оптимізація.* Важливим аспектом роботи є забезпечення доступності та коректного відображення сайту на мобільних пристроях. Використання сучасних технологій адаптивного дизайну та підтримки адаптивного веб-дизайну, щоб забезпечити відмінний користувацький досвід для мобільних користувачів.

## Кешування та покращення взаємодії з базою даних

Було вдосконалено спосіб взаємодії користувачів з веб-сайтом та зменшили навантаження на сервер завдяки:

5. *Використання CDN.* Підхід включає розміщення статичних ресурсів, таких як зображення та стилі, на CDN. Це сприяє підвищенню швидкості завантаження веб-сайту в різних регіонах та забезпечує кращий користувацький досвід.
6. *Оптимізація Базу Даних.* Наукова робота включає оптимізацію структури бази даних та вдосконалення SQL-запитів з метою зменшення часу доступу до інформації. Це дозволяє підвищити продуктивність та відповідати потребам користувачів інтернет-магазину.

Ці наукові новації і методики забезпечують роботі унікальність і роблять її важливим внеском у сферу оптимізації веб-сайтів інтернет-магазинів, враховуючи сучасні тенденції та потреби користувачів. Результати і практичні вдосконалення відзначаються науковою новизною та спрямовані на підвищення комерційної ефективності веб-сайтів.

### 3.4.1. Пошукова оптимізація (SEO)

У світі веб-розвитку і відповідному підвищенні видимості сайту в пошукових системах існує кілька ключових методів, які допомагають досягти цієї мети. Розглянемо деякі з них:

1. *Аналіз ключових слів.* Важливо провести дослідження ключових слів, які найбільше підходять для нашого бізнесу. Ці ключові слова слід використовувати в заголовках, мета-описах, контенті сторінок та URL-адресах, щоб підвищити видимість сайту в пошукових системах.

*2. Покращення швидкості завантаження та оптимізація зображень.*

Зменшення розміру зображень на сайті та використання кешування може позитивно вплинути на час завантаження сторінок, що важливо для користувачів і рейтингу в пошукових системах.

*3. Структура веб-сайту.* Необхідно створити логічну структуру веб-сайту з правильними маршрутами та посиланнями. В цьому випадку, для полегшення індексації вмісту застосовують мікророзмітку.

*4. Соціальні медіа та контент-маркетинг.* Розробка цікавого контенту спрямована на створення цікавого та корисного матеріалу, який привертає аудиторію та збільшує нашу експертизу у відповідній галузі. Використання соціальних медіа для реклами та спілкування з аудиторією.

*5. Аналіз та моніторинг.* Для покращення результатів та ефективності веб-сайту важливо використовувати інструменти веб-аналітики для моніторингу трафіку, конверсій та інших показників. Аналіз результатів оптимізації дозволяє нам вдосконалювати стратегію. Крім цього, регулярне проведення А/В-тестів є необхідним для визначення, які конкретні зміни на сайті призводять до покращення конверсії. Отримані дані використовуються для подальшої оптимізації веб-сайту.

*6. Клієнтська підтримка та відносини.* Для покращення якості обслуговування клієнтів і залучення їх знову, важливо забезпечити швидкий та ефективний зв'язок з ними через різні канали, такі як чат, телефон, електронна пошта та соціальні медіа. Крім цього, розгляньте можливість впровадження програми лояльності та надання різних переваг постійним клієнтам для збільшення їх лояльності і підтримки відносин.

*7. Розширення бізнесу та реклама.* Для досягнення більшого успіху та збільшення продажів можна розглянути наступні можливості:

Розширення асортименту товарів або послуг. Ця стратегія спрямована на привертання більше клієнтів, які шукають різноманітні варіанти і рішення.



Рекламні кампанії. Використовуючи різні канали, такі як Google Ads, соціальні медіа та контекстна реклама, ви можете привертати нових клієнтів та збільшувати свою видимість на ринку.

Співпраця та афіліатський маркетинг. Це відкриває можливість співпрацювати з іншими брендами або запустити афіліатські програми з метою розширення вашої аудиторії та збільшення обсягу продажів.

## 3.5 Тестування

### 3.5.1. Функціональне тестування

Функціональне тестування спрямоване на перевірку правильності роботи функцій та функціональних можливостей розробленої системи.

*Тести та тестові дані:*

1. Реєстрація користувача:
  - Введення правильних даних користувача та перевірка, чи відбувається успішна реєстрація, зображено на рис.3.30.

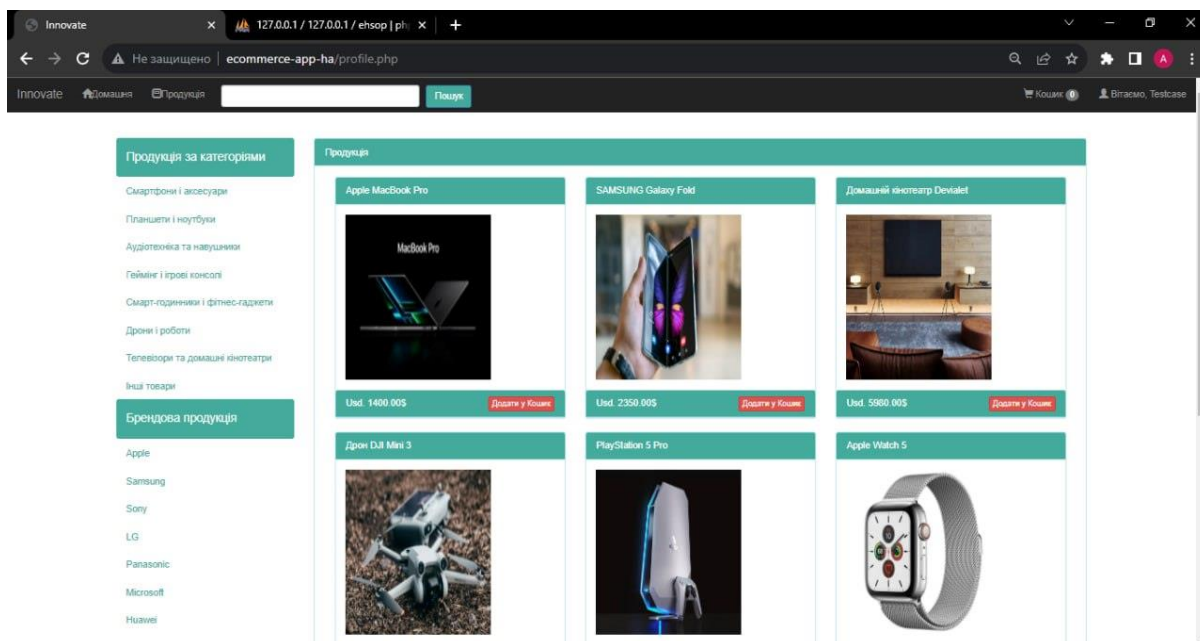


Рисунок 3.30 – Успішна реєстрація користувача

- Введення неправильного паролю та перевірка, чи відображається повідомлення про помилку, зображено на рис. 3.31.

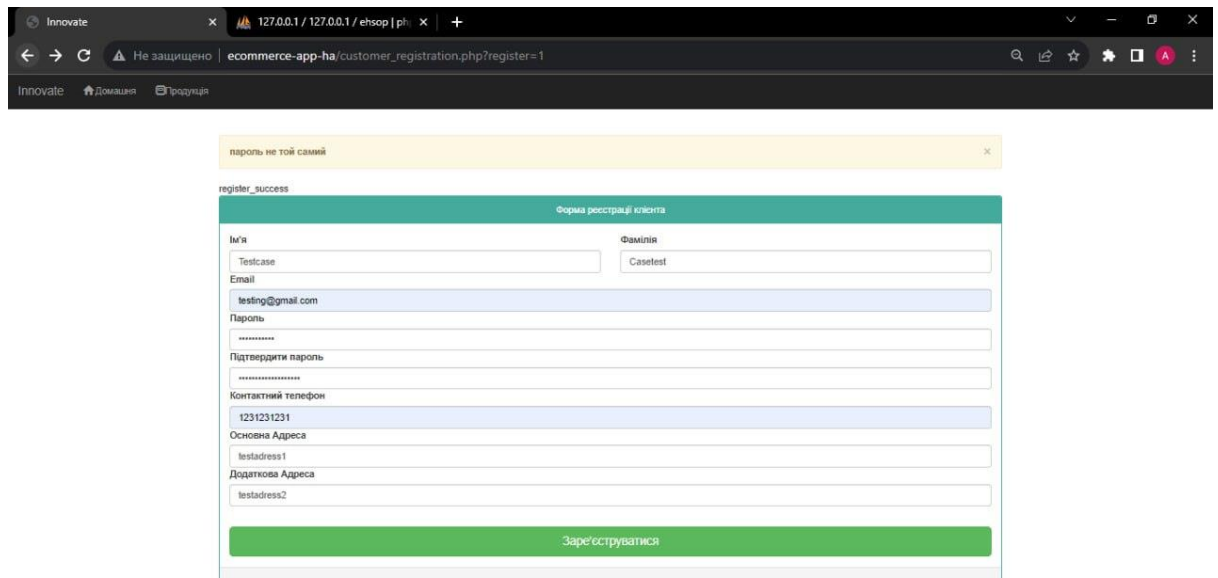


Рисунок 3.31 – Перевірка валідації

2. Вхід користувача:
  - Введення правильних облікових даних користувача та перевірка, чи можна увійти в систему, зображено на рис. 3.32.

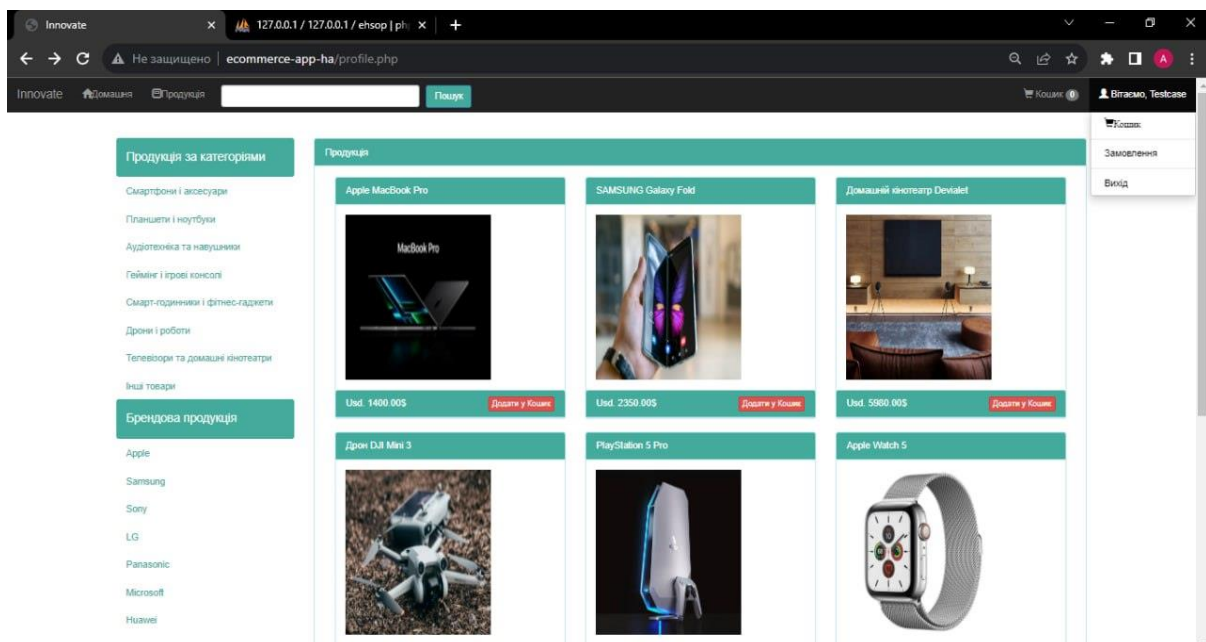


Рисунок 3.32 – Успішний логін користувача

- Введення неправильного паролю та перевірка, чи відображається повідомлення про помилку, зображено на рис. 3.33.



Рисунок 3.33 – Валідація входу незарєстрованого користувача

### 3. Додавання товару до кошика:

- Додавання товару до кошика та перевірка, чи він відображається в кошику, зображено на рис.3.34.

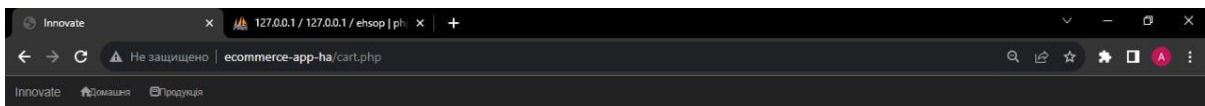


Рисунок 3.34 – Додавання та відображення

#### 4. Оформлення замовлення:

- Заповнення форми для оформлення замовлення та перевірка, чи зберігаються введені дані, зображено на рис. 3.35.

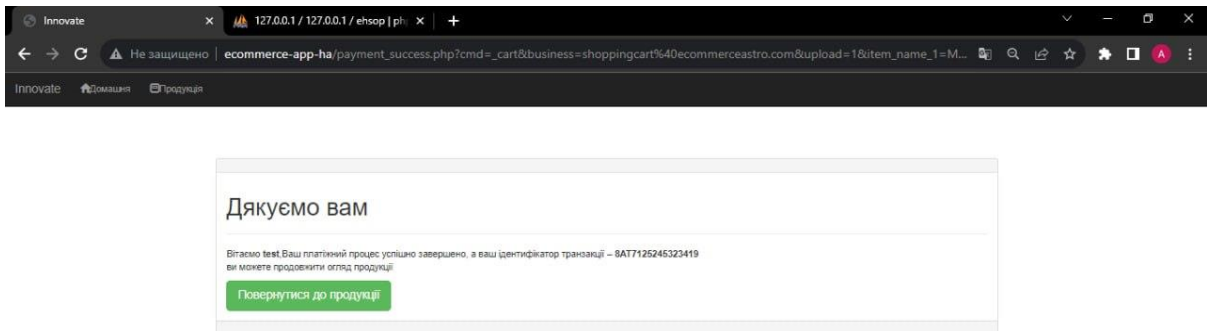


Рисунок 3.35 – Перевірка оформлення замовлення

#### 5. Пошук товару:

- Введення запиту в поле пошуку та перевірка, чи відображаються відповідні результати пошуку, зображено на рис. 3.36.

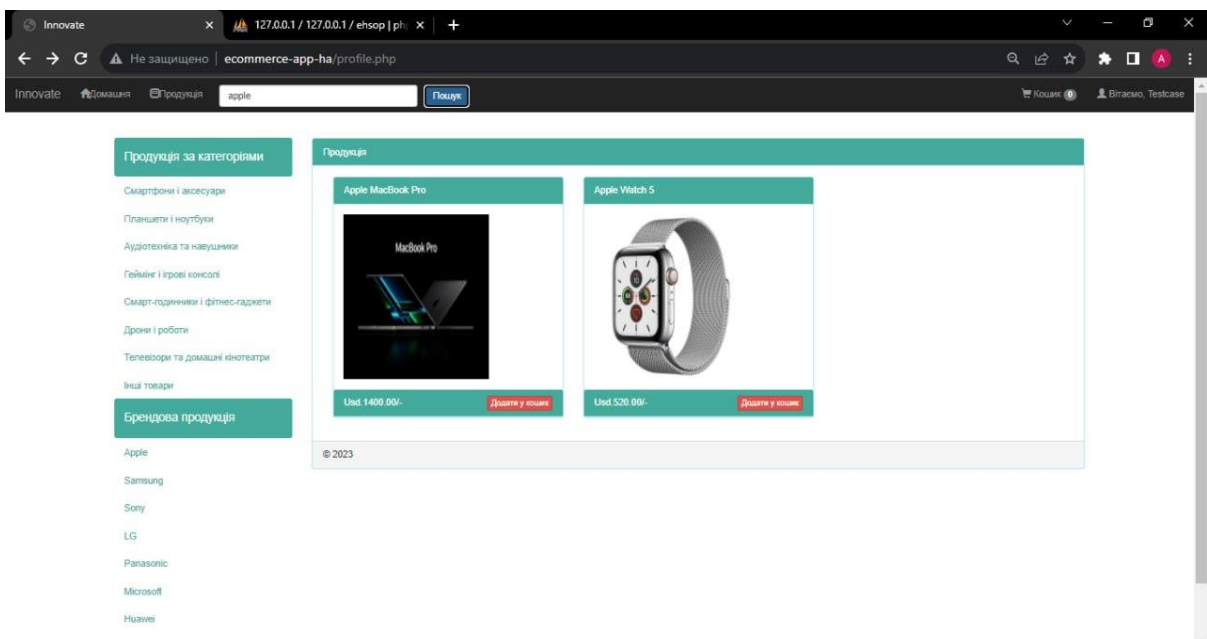


Рисунок 3.36 – Пошук відображує техніку apple

#### 6. Видалення товару з кошика:

- Видалення товару з кошика та перевірка, чи оновлюється кошик правильно, зображено на рис. 3.37 та на рис. 3.38.

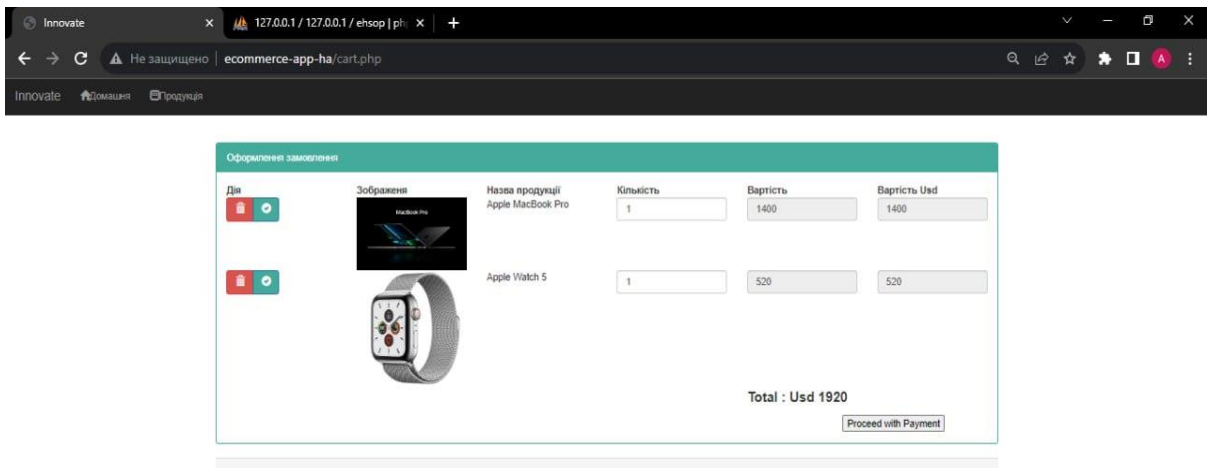


Рисунок 3.37 – Товари до видалення

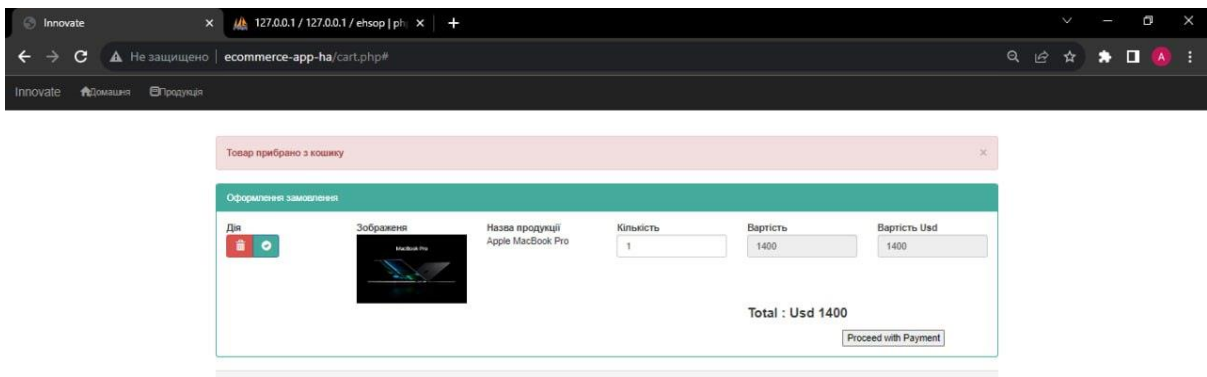


Рисунок 3.38 – Коректне видалення товару

7. Зміна кількості товару в кошику:
  - Зміна кількості товару в кошику та перевірка, чи відображається правильна кількість та вартість замовлення, зображено на рис. 3.39.

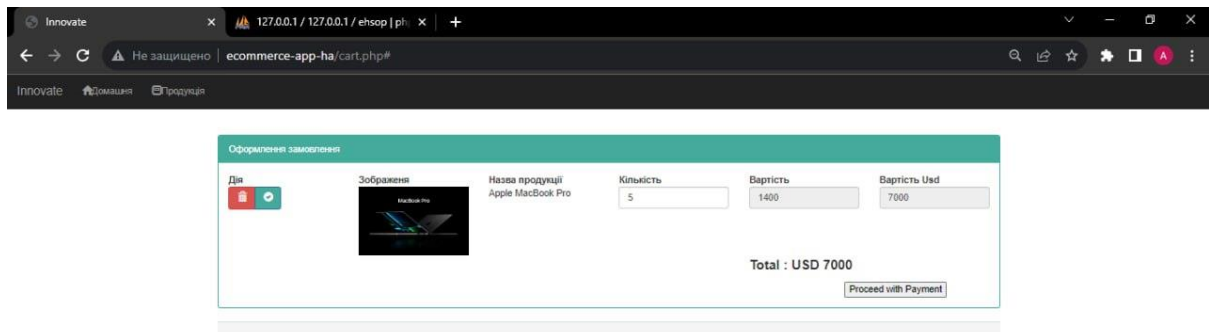


Рисунок 3.39 – Зміна кількості товарів

В результаті проведених дій, всі функції повинні працювати правильно, і користувач повинен мати можливість виконувати операції без помилок.

## Висновки до розділу

Проведено порівняльний аналіз тестування веб-сайту без впровадження заходів по його оптимізації, та сайту, в якому застосувалася оптимізація.

У структурі веб-сайту без застосування оптимізації система меню та навігації не були досконалыми, що ускладнювало пошук необхідної інформації для клієнтів. Функціонал сайту також мав обмеження, зокрема, відсутність зручної системи реєстрації та авторизації користувачів, що може бути стримуючим фактором для покупців. Швидкість завантаження сайту не відповідала сучасним стандартам, що могло призвести до втрати клієнтів. Також SEO-показники були низькими, що обмежувало видимість веб-сайту в пошукових системах.

У структурі веб-сайту із застосування запропонованих заходів, описаних у розділі 3, помітні поліпшення стали очевидними. Технічні аспекти були оптимізовані для забезпечення максимальної продуктивності. Структура веб-сайту стала більш логічною та зручною для користувачів, що полегшило їм пошук необхідної інформації. Функціонал сайту був розширений, забезпечуючи зручну систему реєстрації та авторизації користувачів, що позитивно вплинуло на їхній досвід. Оптимізація швидкості завантаження та поліпшення SEO-показників допомогли залучити більше відвідувачів та підвищити видимість веб-сайту в пошукових системах.

Отже, впроваджені заходи дійсно призвели до покращення комерційної ефективності веб-сайту, забезпечивши зручний та безпечний досвід для користувачів та підвищивши його конкурентоспроможність на ринку.

## ВИСНОВКИ

В кваліфікаційній роботі був проведений аналіз електронної комерції в Україні. Визначено, що наявність веб-сайту є ключовою складовою успіху у сфері електронної комерції, а оптимізація цього веб-сайту дозволить підвищити комерційну ефективність.

На основі проведеного аналізу було запропонована інформаційна модель системи, яка дозволила покращити функціонал, дизайн та інтерфейс, а також забезпечити безпеку інформації.

Вибір технологічного стеку був вирішальним кроком у розробці веб-сайту, і він включав вибір серверного середовища, фреймворку для веб-розробки та засобів безпеки. Розробка структури веб-сайту та реалізація функціоналу, такого як реєстрація користувачів, додавання товарів до кошика та оформлення замовлення, показали свою ефективність. Завдяки використанню CDN та оптимізації бази даних було вдосконалено спосіб взаємодії користувачів з веб-сайтом та зменшено навантаження на сервер. Запропоновані асинхронне завантаження ресурсів та оптимізація зображень дозволили підвищити продуктивність веб-сайту та швидкість завантаження сторінок. Було досягнуто підвищення комерційної ефективності веб-сайту за рахунок покращення таких технічних характеристик як швидкість завантаження, підвищення SEO-показників та мобільної оптимізації.

Таким чином, наукова новизна кваліфікаційної роботи базується на впровадженні структурованих підходів до оптимізації веб-сайту інтернет-магазину, що покращують якість і результати роботи.

Результати дослідження свідчать, що розроблений веб-сайт є конкурентоспроможним на ринку і здатним привертати клієнтів.

Перспективи подальших досліджень можуть бути спрямовані на розширення функціоналу, вдосконалення оптимізації та вивчення інших



аспектів електронної комерції, зокрема впливу маркетингових стратегій на комерційну ефективність інтернет-магазинів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Корогод Г.О., Дерев'янку О.О. Оптимізація ВЕБ-сайтів по підвищенню комерційної ефективності інтернет-магазинів // Інформаційні технології в науці, виробництві та підприємстві: Збір. наук. праць молодих вчених, аспірантів, магістрів кафедри комп'ютерних наук та технологій / загал.наук.ред. В.Ю.Щербань – К.:ТОВ "Фастбінд Україна". – 2023. – с.135-137.
2. Комерційна ефективність України [Електронний ресурс] – Режим доступу до ресурсу:  
[https://uz.ligazakon.ua/ua/magazine\\_article/EA016163#:~:text=У%202021%20році%20український%20ринок,країн%20Центральної%20та%20Східної%20Європи.](https://uz.ligazakon.ua/ua/magazine_article/EA016163#:~:text=У%202021%20році%20український%20ринок,країн%20Центральної%20та%20Східної%20Європи.)
3. Сергієнко О. В. Ефективність веб-сайту як інструменту рекламування продукції підприємства / О. В. Сергієнко, О. Л. Гунька // Ефективна економіка. — 2015. — № 12. — Режим доступу: <http://www.economy.nayka.com.ua/?op=1&z=4620>.
4. Щербань Н. І. Розробка веб-сайту інтернет-магазину: технологічний аспект / Н. І. Щербань // Системні дослідження та інформаційні технології. — 2016. — № 2. — С. 116–122.
5. Муравйова І. О. Розробка ефективного веб-сайту для інтернет-магазину / І. О. Муравйова, О. В. Сулова // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я. — 2017. — № 1 (11). — С. 69–76.
6. Гусарова Т. М. Маркетинг інтернет-магазину: теоретичний аспект / Т. М. Гусарова // Вісник Харківського національного університету імені В. Н. Каразіна. Серія "Економічна". — 2018. — № 35. — С. 179–183.

7. Рибак О. В. Оптимізація веб-сайтів як інструмент підвищення комерційної ефективності інтернет-магазину / О. В. Рибак, А. І. Мізинцева // Економічний аналіз. — 2019. — № 28. — С. 69–74.
8. Слободянюк В. В. Технічна оптимізація інтернет-магазину на платформі WordPress з метою збільшення продажів / В. В. Слободянюк, О. В. Палій // Вісник КНТУ. Серія: Економіка, право, менеджмент. — 2020. — № 2. — С. 176–181.
9. Попович Н. М. Вплив SEO-оптимізації на ефективність інтернет-магазину / Н. М. Попович, В. О. Клименко // Вісник Харківського національного університету імені В. Н. Каразіна. Серія "Економічна". — 2021. — № 46. — С. 185–192.
10. Дубніцька І. М. Методи підвищення ефективності інтернет-магазину / І. М. Дубніцька, Л. М. Лукашевич // Збірник наукових праць КНУТД. Серія: Економіка та менеджмент. — 2022. — № 2. — С. 16–22.
11. Медведєва І. В. Інтеграція соціальних медіа в інтернет-магазині: можливості та переваги / І. В. Медведєва, О. М. Поліщук // Економіка та управління підприємствами: сучасний стан та перспективи розвитку. — 2023. — № 1. — С. 32–39.
12. Гурьянов О. В. Основи веб-розробки: навчальний посібник / О. В. Гурьянов, О. І. Барабаш. — К.: КНЕУ, 2016. — 240 с.
13. Іванов С. О. Розробка інтернет-магазину на платформі OpenCart / С. О. Іванов, Т. В. Журавльова // Моделювання та інформаційні технології в економіці. — 2017. — № 3. — С. 30–36.
14. Коваленко М. І. Проектування інтерфейсу користувача для веб-сайту інтернет-магазину / М. І. Коваленко, О. С. Шевченко // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я. — 2018. — № 2 (14). — С. 58–64.

15. Міщенко О. В. Способи залучення клієнтів до інтернет-магазину / О. В. Міщенко, Т. А. Пономаренко // Економіка і суспільство. — 2019. — № 19. — С. 134–139.
16. Попова І. В. Аналіз методів оптимізації швидкості завантаження веб-сайту / І. В. Попова, Ю. В. Шаповалова // Ефективна економіка. — 2020. — № 12. — С. 76–82.
17. Сідорова А. С. Вплив кольорового рішення веб-сайту на рівень конверсії / А. С. Сідорова, Т. М. Марченко // Вісник Харківського національного університету імені В. Н. Каразіна. Серія "Психологія". — 2021. — № 72. — С. 47–53.
18. Ткаченко Д. В. Аналіз та вибір платформи для розробки інтернет-магазину / Д. В. Ткаченко, Н. О. Бондарчук // Системи обробки інформації. — 2022. — № 5. — С. 22–29.
19. Щербань Ю. О. Оптимізація структури веб-сайту для поліпшення користувацького досвіду / Ю. О. Щербань, О. В. Гребень // Інформаційні технології та комп'ютерна інженерія. — 2023. — № 1 (53). — С. 45–51.
20. Adams, T. (2016). *Web Development & Design Foundations with HTML5*. Pearson — С. 45-67.
21. Dewsbury, G., & Ride, P. (2017). *HTML5 and CSS3 All-in-One For Dummies*. Wiley — С. 112-135.
22. McFarland, D. (2019). *JavaScript & jQuery: The Missing Manual*. O'Reilly Media — С. 78-94.
23. Duckett, J. (2020). *HTML and CSS: Design and Build Websites*. Wiley— С. 23-45.
24. Myers, E., & Hettiarachchi, S. (2021). *Pro PHP Security: From Application Security Principles to the Implementation of XSS Defenses*. Apress — С. 155-178.

25. Smith, A., & Collingridge, G. (2022). The Ultimate Guide to E-commerce SEO. *Search Engine Journal* — С. 220-245.
26. Kissane, E., & Whalin, T. (2016). *Web Performance Daybook Volume 2*. O'Reilly Media — С. 75-92.
27. Greig, D. (2017). The Ultimate Guide to Getting Started with SEO. *Moz*— С. 50-71.
28. Poynter, D., & Bollmeyer, S. (2018). *Conversion Optimization: The Art and Science of Converting Prospects to Customers*. Wiley — С. 102-128.
29. Maughan, D., & Cattell, A. (2019). *Google Analytics Breakthrough: From Zero to Business Impact*. Wiley — С. 95-118.
30. Schwartz, B. (2020). A Guide to HTTP/2 Server Push. *Smashing Magazine* — С. 34-56.
31. Patel, N. (2021). *The Complete Guide to SEO*. QuickSprout — С. 88-109.
32. Стефаненко, Д. О. (2017). Розробка веб-сайтів та веб-додатків: навчальний посібник. Видавництво "Ін Юре" — С. 67-89.
33. Манжос, О. І. (2018). Інтернет-маркетинг: підручник. Київський університет. Сторінки: 42-64.
34. Довгаль, О. М., & Гордійчук, О. І. (2019). Мобільні додатки для бізнесу: навчальний посібник. Центр учбової літератури — С. 75-97.
35. Лесів, А. О. (2021). Інтернет-магазин: розробка та просування. Київський університет — С. 110-132.
36. Томашевський, В. С. (2016). Веб-розробка для початківців: навчальний посібник. Київський університет — С. 25-48.
37. Бойчук, О. В. (2017). Оптимізація та просування веб-сайтів. Видавництво "Академія" — С. 82-104.
38. Реброва, Т. І. (2018). Реклама та продвиження в Інтернеті: підручник. Центр учбової літератури — С. 67-88.
39. Петренко, С. М. (2019). Веб-аналітика та її використання: практичний посібник. Видавництво "Символ-Плюс" — С. 45-67.

40. Сімонов, О. П. (2020). Розробка мобільних додатків для Android: навчальний посібник. Київський університет — С. 78-99.

## ДОДАТКИ