

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)
на тему:

«Інтернет магазин з продажу комп'ютерної техніки»

Виконав студент групи КН-20-2
ДЬОМІН Дмитро Олексійович

Керівник роботи:
ПЕТРОСЯН Руслан Валерікович

Рецензент:
МАРЧУК Галина Вікторівна

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

Марина ГРАФ

«25» січня 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)

Здобувач вищої освіти: **ДЬОМІН Дмитро Олексійович**

Керівник роботи: **ПЕТРОСЯН Руслан Валерікович**

Тема роботи: **«Інтернет магазин з продажу комп'ютерної техніки»,**

затверджена наказом закладу вищої освіти від **«25» січня 2024 р., №22/с**

Термін здачі закінченої роботи: **«28» червня 2024 р.**

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): предмет дослідження:
технології веб-розробки, зокрема використання сучасних фреймворків та їх бібліотек.

Об'єкт дослідження: є процес розробки та впровадження інтернет-магазину.

Консультанти випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв
1			
2			
3			

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формулювання технічного завдання.	1 лютого 2024 – 9 березня 2024	Виконано
2	Проектування структури системи	10 березня 2024 – 14 квітня 2024	Виконано
3	Написання програмного коду	15 квітня 2024 – 22 квітня 2024	Виконано
4	Тестування системи	23 квітня 2024 – 24 квітня 2024	Виконано
5	Оформлення пояснювальної записки	25 квітня 2024 – 01 червня 2024	Виконано
6	Попередній захист	06 червня 2024	Виконано

Здобувач вищої освіти **Дмитро ДЬОМІН**

Керівник **Руслан ПЕТРОСЯН**

РЕФЕРАТ

Кваліфікаційна робота складається з програмного комплексу «Інтернет магазин з продажу комп'ютерної техніки» та пояснювальної записки. У роботі визначено основні завдання на розробку веб-додатку.

Метою даної роботи, на тему «Інтернет магазин з продажу комп'ютерної техніки», є дослідження та використання сучасних веб-технологій для функціонування сайту та дати можливість кінцевому користувачу з легкістю замовити комп'ютерну техніку.

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел, 1 додатка. Загальний обсяг роботи – 68 сторінок, у тому числі 57 сторінок основного тексту, 1 сторінки списку використаних джерел, 10 сторінок додатків.

В роботі наведено 45 рисунка.

Загальний обсяг роботи 68 сторінок.

КЛЮЧОВІ СЛОВА: ВЕБ-ДОДАТОК, КОМП'ЮТЕРНА ТЕХНІКА, JAVASCRIPT, MONGODB, DATABASE.

					КН.КР.Б – 122 – 24 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	«Інтернет магазин з продажу комп'ютерної техніки» Пояснювальна записка	Літ.	Арк.	Аркушів
Розроб.		Д. О. ДЬОМІН						
Керівник		Р. В. ПЕТРОСЯН					3	58
Рецензент		Г. В. МАРЧУК				Житомирська політехніка, група КН-20-2		
Н. контр.								
Зав. каф.		М.С. ГРАФ						

ABSTRACT

The qualification work consists of the software package "Online store selling computer equipment" and an explanatory note. The work defines the main tasks for the development of a web application.

The purpose of this work, on the topic "Online computer store", is to study and use modern web technologies for the functioning of the site and to enable the end user to easily order computer equipment.

The explanatory note consists of an introduction, 3 chapters, conclusions, a list of references, and 1 appendix. The total volume of the work is 68 pages, including 57 pages of the main text, 1 page of the list of references, 10 pages of appendices.

The paper contains 45 pictures.

The total volume of the work is 68 pages.

KEYWORDS: WEB APPLICATION, COMPUTER TECHNOLOGY, JAVASCRIPT, MONGODB, DATABASE.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ.....	3
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	8
1.1 Постановка задачі	8
1.2 Аналіз аналогів інтернет-магазинів з продажу комп'ютерної техніки та аксесуарів	9
1.3 Обґрунтування вибору інструментальних засобів	17
Висновки до першого розділу	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ	20
2.1 Моделювання процесу роботи інтернет магазину	20
2.2 Проектування інформаційної системи.....	21
2.3 Проектування моделі баз даних	28
2.4 Реалізація програмного коду алгоритмів роботи системи сайту.....	23
Висновки до другого розділу	42
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЙОГО ТЕСТУВАННЯ	44
3.1 Налаштування та встановлення параметрів системи	45
3.2 Інтерфейс сайта та алгоритми роботи з ним.....	30
3.3 Тестування веб-додатку.....	54
Висновки до третього розділу	55
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58

Перелік умовних скорочень

DB – база даних

ПЗ – програмне забезпечення

СУБД – система управління базами даних

API – прикладний програмний інтерфейс

ВД – веб-додаток

ПК – персональний комп'ютер

User – користувач

UI - інтерфейс

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми.

У сучасному світі швидкого розвитку цифрових технологій та інтернет-комерції, створення сайту інтернет-магазину з продажу комп'ютерної техніки є надзвичайно актуальним завданням. Зростаюча потреба у комп'ютерних пристроях та аксесуарах, підкріплена зростанням дистанційної роботи, навчання та розваг, вимагає ефективних рішень для задоволення споживацьких потреб. Інтернет-магазини забезпечують зручність та доступність, дозволяючи клієнтам вибирати, порівнювати та купувати товари, не виходячи з дому. Усе це підкреслює важливість розробки ефективного, функціонального та користувачо-орієнтованого веб-ресурсу, який відповідає сучасним вимогам ринку.

Мета кваліфікаційної роботи.

Метою кваліфікаційної роботи є розробка інтерактивного, зручного у використанні та функціонально багатого сайту інтернет-магазину з продажу комп'ютерної техніки, який забезпечить ефективну взаємодію з користувачами, підтримку процесу купівлі-продажу та сприятиме зростанню продажів і задоволеності клієнтів.

Об'єктом дослідження є процес розробки та впровадження інтернет-магазину, який спеціалізується на продажу комп'ютерної техніки та супутніх товарів.

Предметом дослідження є технології веб-розробки, зокрема використання сучасних фреймворків та бібліотек (таких як React, Next.js, Tailwind CSS), методи забезпечення користувацького досвіду (UX/UI), а також механізми інтеграції платіжних систем та управління базами даних для створення ефективного інтернет-магазину з продажу комп'ютерної техніки.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1.1 Постановка задачі

Основною метою створення веб-додатку для продажу комп'ютерної техніки є: ознайомлення з сучасними веб-технологіями по створенню та впровадження ВД. ВД розрахований на мінімальну кількість кнопок та розділів, щоб користувач не плутався під час навігації по сайту, що значно прискорить процес вибору та замовлення необхідних товарів.

В випускній роботі необхідно розробити додаток у вигляді інтернет-магазину та якомога краще спростити інтерфейс для користувача. Для реалізації поставленого завдання основними етапами є:

1. Провести аналіз ринку інтернет-магазинів з продажу комп'ютерної техніки, визначити основні тенденції та вимоги споживачів.
2. Дослідити конкурентів та визначити їх сильні та слабкі сторони.
3. Створити прототип користувацького інтерфейсу, враховуючи принципи UX/UI дизайну.
4. Забезпечити інтерактивність та зручність навігації по сайту.
5. Впровадити систему замовлень та чеків.
6. Реалізувати можливість створення особистого кабінету для треку статусів замовлення користувача.
7. Провести тестування функціональності сайту, виправити знайдені помилки та недоліки

Результатом реалізації поставленого завдання є веб-додаток з продажу комп'ютерної техніки.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз аналогів інтернет-магазинів з продажу комп'ютерної техніки та аксесуарів

Аналіз аналогів інтернет-магазинів з продажу комп'ютерної техніки та аксесуарів є важливим етапом у процесі розробки власного веб-ресурсу. Даний аналіз дозволяє вивчити існуючі рішення на ринку, оцінити їх сильні та слабкі сторони, а також визначити найкращі практики, які можна застосувати у власному проєкті. Інтернет-магазини, що спеціалізуються на продажу комп'ютерної техніки та аксесуарів, зазвичай мають широкий асортимент продукції, інтегровані платіжні системи, зручний інтерфейс користувача та інші функціональні можливості, які забезпечують зручність та ефективність процесу покупки.

Вивчення аналогів дозволяє не лише знайти конкурентні переваги, але й уникнути помилок, які вже були допущені іншими компаніями. Це також допомагає зрозуміти, які функції та сервіси є найбільш затребуваними серед користувачів, що сприяє створенню більш орієнтованого на клієнта продукту. У результаті, детальний аналіз існуючих інтернет-магазинів є необхідним для формування стратегії розвитку та впровадження інновацій, які забезпечать конкурентоспроможність нового веб-ресурсу на ринку.

Таким чином ми маємо декілька магазинів що дуже популярні в наш час, давайте їх розглянемо:

Першим буде Rozetka, оскільки на мою думку це один з самих популярних інтернет магазинів і про нього в Україні чув кожний. Основні плюси: зручний мінімалістичний інтерфейс, тех підтримка, великий асортимент товару, офлайн відділення з видачею товару.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

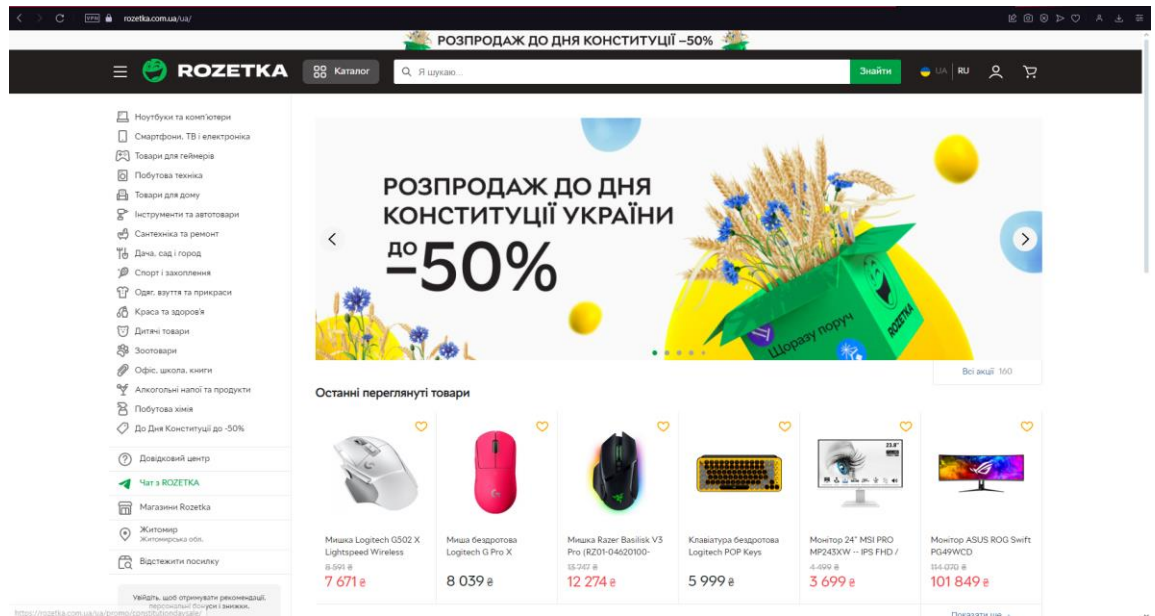


Рис. 1.1 Rozetka

На головній сторінці сайту клієнта зустрічає приємний для ока дизайн. Користувачу зручно зорієнтуватися на сайті, знайти потрібний товар в пошуку або в категоріях. На мою думку, вивіски зі знижками займають більше місця, ніж хотілось би і можуть відволікати. Також з мінусів я би підмітив перенасиченням списку категорій.

Також сайт Rozetka має мобільний додаток.

Із переваг це інтуїтивно зрозумілий інтерфейс, який полегшує навігацію по категоріям товару та пошук необхідних продуктів.

Із недоліків я б виділив технічні проблеми. В додатку часто травляються збої. Часті оновлення додатку. Клієнт відчуває дискомфорт коли намагається відкрити додаток, а він потребує оновлення, іноді по деяким причинам людина немає доступу до швидкого інтернету що б оновитися. З веб-сайтом таких проблем немає.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 1.2 Мобільний додаток Rozetka

У мобільному додатку інтерфейс представлений з мінімальною кількістю кнопок, що надає змогу користувачу швидше зорієнтуватися в магазині та знайти потрібний товар.

Наступний на огляді Зопа51, який вміщає в себе виключно товари пов'язані з комп'ютерами та аксесуарами.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

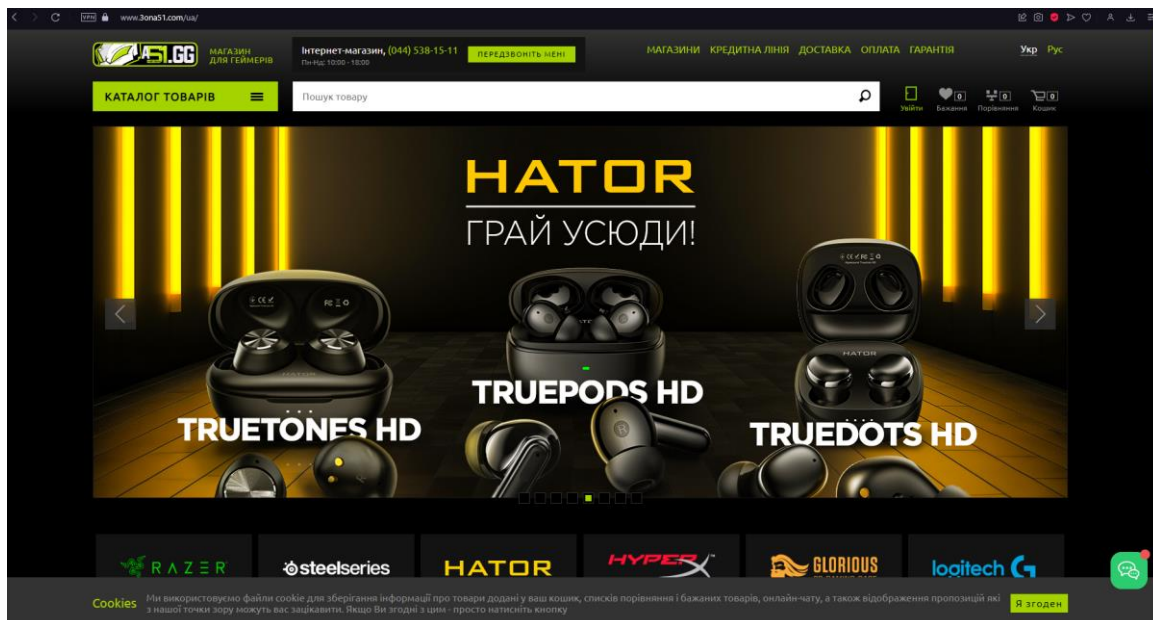


Рис. 1.2. Зона51

Дизайн цього сайту виконано в темних кольорах, що надає йому особливого індустріального стилю, але ключові кнопки при цьому видно гірше. На сайті також значний простір займає реклама акційних товарів, але за рахунок того що навігація по сайту розташована в одному місці, реклама не заважає користувачу в пошуках конкретного розділу або товару.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

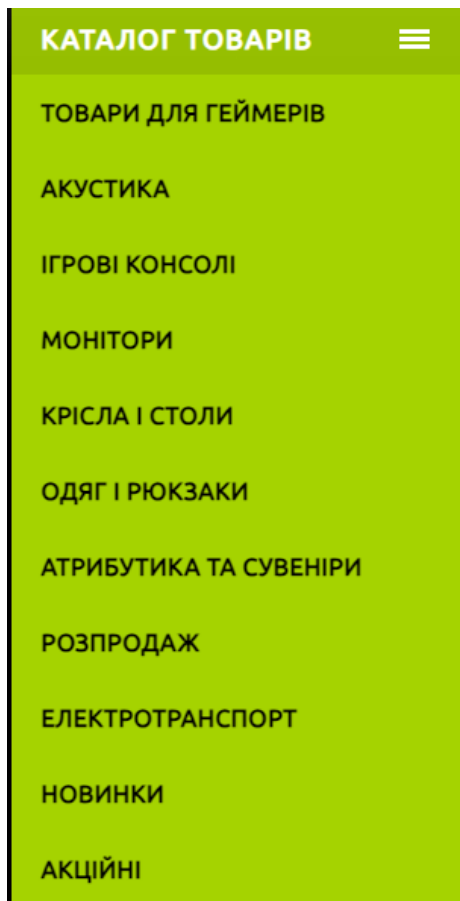


Рис. 1.3. Каталог товарів Зона51

З мінусів я б виділив неприємну для ока кольорову палітру. На деяких сторінках сайт виявився дуже темним з ледь-ледь помітними кнопками, а десь навпаки – кольори занадто кислотні.

Сайт Brain для мене поєднав сильні сторони двох попередніх конкурентів. Він має приємне оформлення, всі представлені товари на сайті пов'язані тільки з комп'ютерною технікою і тому сайт не перевантажений категоріями та також має офлайн магазини де можна в живу побачити та придбати товар, або забрати замовлення.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

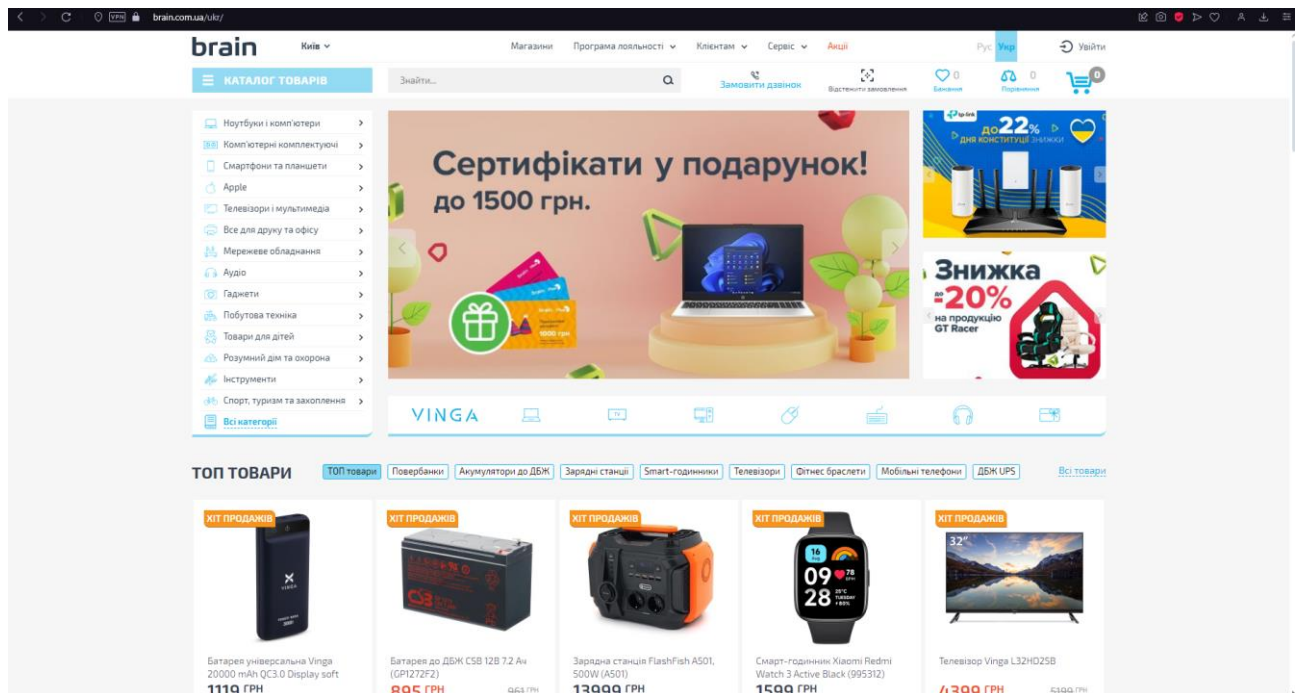


Рис. 1.4. brain

На мою думку цей сайт має найбільш клієнто орієнтованість. На ньому завжди є знижки на актуальні товари, програма лояльності та накопичення бонусів. Конкретно у сайті мінусів не виявлено, але сам магазин не має телефонного додатку, як його конкуренти.

Останнім на огляді буде вузьконаправлений інтернет магазин Elmir

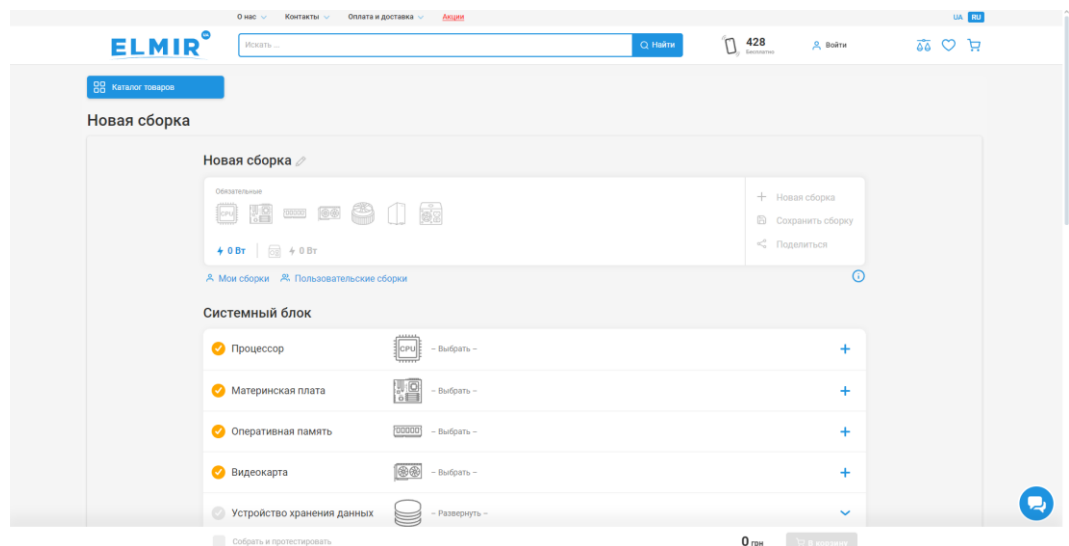


Рис. 1.5 Elmir

Цей інтернет магазин спеціалізований виключно на комп'ютерній техніці та має конфігуратор, який допоможе зібрати ПК навіть зовсім недосвідченому користувачу під його бюджет і цілі.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Системный блок					
✓	Процессор		Процессор AMD Ryzen 5 5600 s-AM4 3.5GHz/32MB Tray (100-000000927) Код товара: 1345639	4 799 грн	
✓	Материнская плата		Материнская плата GigaByte B550M AORUS Elite s-AM4 B550 Код товара: 1135724	4 399 грн	
✓	Оперативная память		Модуль памяти G.Skill DDR4 16GB 2x8GB 3200MHz Aegis (F4-3200C16D-16GIS) Код товара: 1000781	- 2 + 2 664 грн	
✓	Видеокарта		Видеокарта Asus PCI-E GeForce RTX3050 6GB DDR6 (DUAL-RTX3050-O6G) Код товара: 1515450	- 1 + 8 409 грн	
✓	Устройство хранения данных		SSD-накопитель 2.5" SATA 512GB GoodRam CX400 Gen.2 (SSDPR-CX400-512-G2) Код товара: 1139375 Добавить ещё	- 1 + 1 439 грн	
✓	Охлаждение процессора		Кулер для процессора ID-Cooling SE-214-XT ARGB Код товара: 1381975	797 грн	
✓	Корпус		Корпус 1stPlayer D3-3F2-PLUS Black 6/БП Код товара: 1311272	1 799 грн	
✓	Блок питания		Блок питания 400W Delux DLP-25D Код товара: 426234	479 грн	
Дополнительные компоненты					

Рис. 1.6 Конфігуратор ПК

Якщо ж користувач в конфігураторі вибирає компоненти які не працюють у зв'язці одне з одним, сайт повідомить про це й запропонує змінити замовлення. Ця функція відрізняє Elmir від інших сайтів-дестриб'ютерів та позитивно впливає на враження покупців. Конфігуратор ПК позбавляє зробити невірний вибір та в наслідку втрачати час та кошти на повернення товару.

✓	Процессор		Процессор AMD Ryzen 5 5600 s-AM4 3.5GHz/32MB Tray (100-000000927) Код товара: 1345639	4 799 грн	
✓	Материнская плата		Материнская плата GigaByte B550M AORUS Elite s-AM4 B550 Код товара: 1135724	4 399 грн	
✗	Оперативная память		Модуль памяти G.Skill DDR4 16GB 2x8GB 3200MHz Aegis (F4-3200C16D-16GIS) Код товара: 1000781	- 3 + 3 996 грн	
✗	Видеокарта		Видеокарта Asus PCI-E GeForce RTX3050 6GB DDR6 (DUAL-RTX3050-O6G) Код товара: 1515450	- 3 + 25 227 грн	

Рис. 1.7 Несумісність комплектуючих ПК

Отже, найпоширеніші помилки та недоліки інтернет-магазинів для покупки ПК та аксесуарів

1. Неприємний дизайн

Складна навігація: Погано структуроване меню або невпорядковані категорії можуть ускладнити пошук потрібного товару.

Застарілий або недоречний дизайн: Візуально непривабливий дизайн може відлякати покупців і створити негативне враження про магазин.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Нечитабельні шрифти та кольори: Використання невідповідних шрифтів або кольорів, які роблять текст важкочитабельним, впливають на користувацький досвід.

2. Технічні помилки

Повільне завантаження сторінок: Сайт, який повільно завантажується, може призвести до втрати клієнтів.

Збої під час оформлення замовлення: Технічні проблеми на етапі оформлення замовлення можуть відлякати клієнтів і призвести до втрати продажів.

Некоректне відображення товарів: Помилки у відображенні товарів або їх характеристик можуть ввести в оману покупців.

3. Відсутність мобільної орієнтованості та додатку

Невідповідність мобільним пристроям: Сайт, який не адаптований для мобільних пристроїв, ускладнює перегляд і покупки для користувачів смартфонів та планшетів.

Відсутність мобільного додатку: Багато користувачів надають перевагу здійсненню покупок через мобільні додатки, відсутність яких може знизити конверсію.

4. Перенасиченість сайту категоріями

Занадто багато категорій: Велика кількість категорій без чіткого структурування може заплутати користувачів і ускладнити пошук потрібних товарів.

Невпорядкованість: Категорії та підкатегорії без логічної структури можуть створити хаос на сайті.

5. Реклама займає багато місця на екрані

Надмірна кількість реклами: Велика кількість реклами може відволікати користувачів від основного контенту та товарів.

Неправильне розташування реклами: Реклама, яка заважає перегляду або перешкоджає навігації, негативно впливає на користувацький досвід.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Спливаючі вікна: Часті спливаючі вікна можуть дратувати користувачів і викликати бажання залишити сайт.

Для успішного інтернет-магазину важливо враховувати дизайн, технічну стабільність, мобільну орієнтованість, логічне структурування категорій та оптимальне розташування реклами. Виправлення цих поширених помилок допоможе створити приємний користувацький досвід і збільшити конверсію.

1.3 Обґрунтування вибору інструментальних засобів

При створенні сайту інтернет-магазину важливим етапом є вибір відповідних засобів реалізації, включаючи мови програмування, фреймворки та бази даних. Це забезпечує надійну роботу сайту, зручність його розробки та обслуговування, а також можливість масштабування у майбутньому.

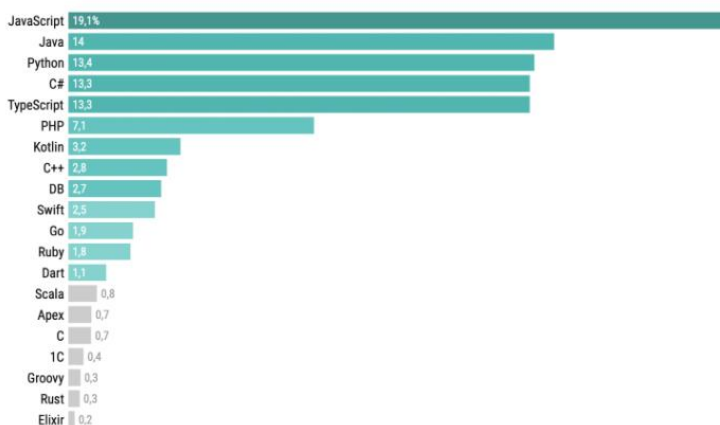


Рис. 1.5 Рейтинг мов програмування

JavaScript (та його фреймворки) є найпопулярнішою мовою програмування при створенні веб сайту інтернет магазинів.

Фреймворки JavaScript:

React: Бібліотека для створення інтерфейсів користувача, що забезпечує високу продуктивність та компонентний підхід до розробки.

Next.js: Фреймворк для React, який додає підтримку серверного рендерингу та статичної генерації сайтів, що підвищує швидкість завантаження та SEO-оптимізацію.

Vue.js: Альтернативна бібліотека для створення інтерфейсів користувача з акцентом на простоту та гнучкість.

Node.js: Середовище виконання JavaScript, яке дозволяє використовувати цю мову на серверній стороні, що забезпечує ефективну обробку запитів та високу продуктивність.

Усю отриману інформацію інтернет магазину потрібно десь зберігати. В цьому нам знадобиться MongoDB

MongoDB є однією з найпопулярніших баз даних NoSQL і має ряд переваг, які роблять її ідеальним вибором для створення інтернет-магазину:

1. Гнучка модель даних:

MongoDB зберігає дані у вигляді документів JSON-подібних форматів (BSON), що дозволяє легко зберігати та змінювати складні дані без необхідності попереднього визначення структури таблиць.

2. Масштабованість:

MongoDB підтримує горизонтальне масштабування, що дозволяє легко додавати нові вузли до кластеру бази даних для обробки зростаючих обсягів даних та підвищення продуктивності.

Висновки до першого розділу

У першому розділі кваліфікаційної роботи було здійснено комплексний підхід до розробки інтернет-магазину комп'ютерної техніки та аксесуарів, що включав постановку задачі, аналіз аналогів і вибір засобів реалізації.

В ході постановки задачі були визначені основні етапи та цілі розробки інтернет-магазину. Визначено, що необхідно створити інтерактивний, зручний у використанні та функціонально багатий веб-ресурс, який задовольнятиме потреби користувачів та сприятиме зростанню продажів. Зокрема, було окреслено завдання аналізу ринку, вибору технологій, розробки дизайну та прототипу, реалізації фронтенд та бекенд частин, інтеграції платіжних систем, тестування, оптимізації та підтримки сайту.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Аналіз аналогів інтернет-магазинів дозволив вивчити поточні рішення на ринку, виявити їхні сильні та слабкі сторони, а також визначити найкращі практики, які можна застосувати у власному проекті. Було досліджено інтерфейси користувача, функціональні можливості, асортимент товарів, платіжні системи та інші ключові аспекти успішних інтернет-магазинів. Це дало змогу сформулювати чітке розуміння вимог до майбутнього веб-ресурсу та знайти конкурентні переваги.

На основі аналізу ринку було обрано найпопулярніші мови програмування та фреймворки для створення інтернет-магазину. Було визначено, що JavaScript (зокрема, React і Next.js) є оптимальним вибором для фронтенд-розробки завдяки своїй популярності, продуктивності та зручності. Для бекенд-розробки було обрано Node.js, який забезпечує ефективну обробку запитів. Крім того, було розглянуто переваги використання MongoDB як бази даних, яка забезпечує гнучкість, масштабованість, високу продуктивність, реплікацію та інтеграцію з JavaScript.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТІВАННЯ САЙТУ ІНТЕРНЕТ-МАГАЗИНУ КОМП'ЮТЕРНОЇ ТЕХНІКИ

2.1 Визначення варіантів використання та структури системи

При розробці інтернет-магазину комп'ютерної техніки та аксесуарів важливо визначити різні вимоги до проекту, які включають бізнес-вимоги, функціональні, нефункціональні та інші специфікації. Це допоможе забезпечити повноту і послідовність реалізації проекту.

Бізнес-вимоги

1. Цілі проекту:

Збільшення продажів комп'ютерної техніки та аксесуарів через онлайн-канал.

Забезпечення зручного та безпечного процесу купівлі для клієнтів.

Підвищення рівня задоволеності клієнтів через якісне обслуговування та підтримку.

2. Цільова аудиторія:

Приватні особи, які шукають комп'ютерну техніку для особистого використання.

Малі та середні підприємства, які потребують комп'ютерного обладнання для бізнес-цілей.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Комерційні вимоги:

Підтримка різних способів оплати, включаючи кредитні картки, PayPal та інші електронні платіжні системи.

Здійснення маркетингових кампаній, включаючи знижки, акції та програми лояльності.

Підтримка багатомовності для обслуговування клієнтів з різних регіонів.

Функціональні вимоги

1. Реєстрація та автентифікація користувачів:

Можливість створення облікового запису користувача з використанням електронної пошти та пароля.

Вхід до системи за допомогою облікових даних або соціальних мереж (Google, Facebook).

2. Каталог товарів:

Відображення каталогу товарів з фільтрацією та сортуванням за різними параметрами (ціна, бренд, категорія).

Можливість перегляду детальної інформації про товар, включаючи опис, характеристики, зображення та відгуки користувачів.

3. Кошик та замовлення:

Додавання товарів до кошика та редагування кількості товарів у кошику.

Оформлення замовлення з вказанням адреси доставки, вибором способу оплати та підтвердженням замовлення.

4. Адміністративна панель:

Управління товарами, включаючи додавання, редагування та видалення товарів.

Перегляд та обробка замовлень, управління статусами замовлень.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Управління користувачами, включаючи редагування даних користувачів та контроль доступу.

5. Пошук:

Реалізація пошуку товарів за ключовими словами.

Завдяки аналізу варіантів використання та структури системи наведено діаграму її використання залежно від статусу користувача.(Рис.2.1)

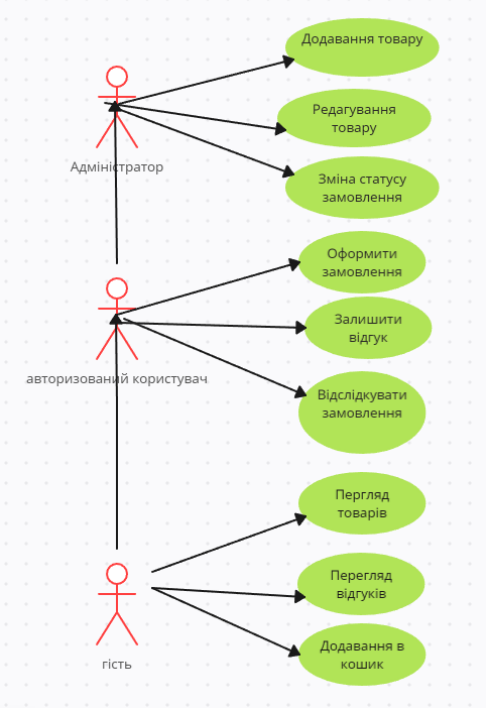


Рис. 2.1 Діаграма варіантів використання додатку

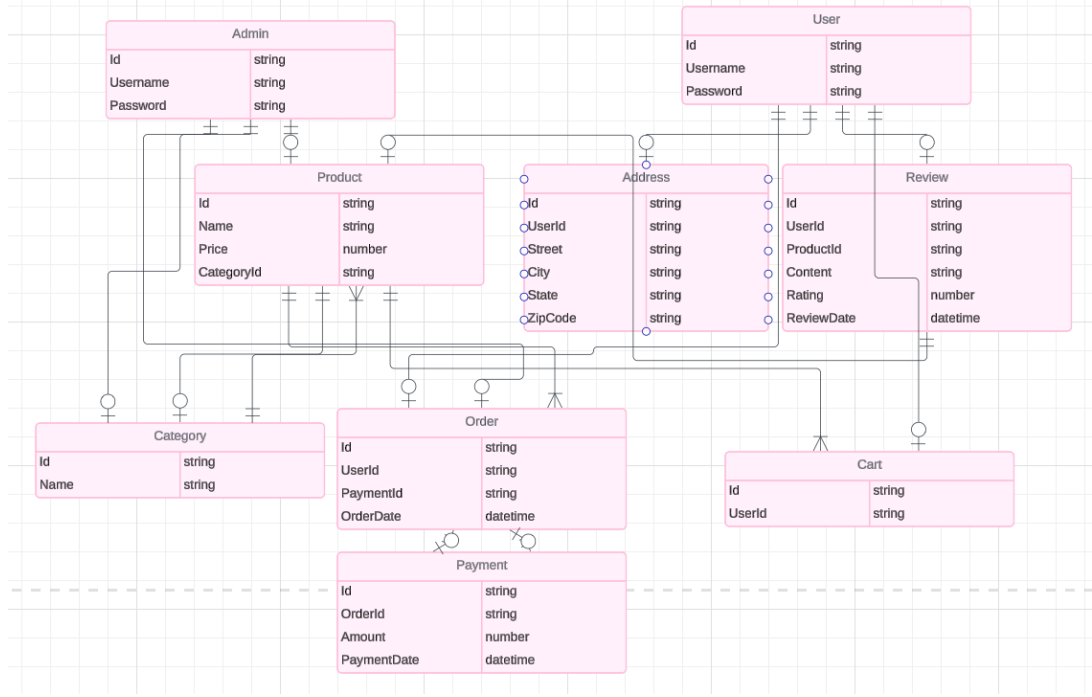


Рис. 2.2 Діаграма сутностей веб додатку

Ця діаграма допомагає візуалізувати взаємозв'язки між різними сутностями в системі та визначити їх основні атрибути та відносини.

User: Користувач, який може мати замовлення (Order), кошик (Cart), відгуки (Review) та адреси (Address).

Product: Товар, який може належати до категорії (Category) та бути частиною замовлення (Order) або кошика (Cart).

Order: Замовлення, яке містить товари (Product) та пов'язане з оплатою (Payment).

Cart: Кошик, який містить товари (Product).

Category: Категорія товарів, до якої належать продукти (Product).

Review: Відгук, який пише користувач (User) про продукт (Product).

Payment: Оплата, пов'язана із замовленням (Order).

Admin: Адміністратор, який керує товарами (Product), категоріями (Category) та замовленнями (Order).

Address: Адреса, яку має користувач (User).

2.2 Розробка бази даних для веб сайту

Розробка бази даних для веб-сайту з використанням MongoDB може значно спростити процес створення та підтримки сучасних веб-додатків, забезпечуючи високу продуктивність і масштабованість.

Для того щоб створити базу даних та під'єднати її до проекту, спочатку потрібно зареєструватися на сайті MongoDB. Сайт пропонує пройти швидку реєстрацію через Google, ми скористаємося цим (Рис.2.3)

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		



Log in to your account

Don't have an account? [Sign Up](#)



Or with email and password

Email Address

Next

Рис.2.3 Реєстрація на MongoDB

Наступним кроком є налаштування IP-адреси, оскільки сервер у нас локальний, це немає значення та сайт пропонує вибрати мій поточний IP.(Рис.2.4)

Також важливим налаштуванням є дозвіл доступ з різних IP-адрес до додатку. На локальний сервер цілком безпечно дозволити під'єднання з усіх можливих адрес.(Рис.2.5)

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

Add entries to your IP Access List

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the [Network Access Page](#).

IP Address

Description

IP Access List

Description

102.219.208.38/32

My IP Address

Рис. 2.4 Додавання IP-адреси

Add IP Access List Entry

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more](#).

Access List Entry:

Comment:



This entry is temporary and will be deleted in

6 hours

Рис.2.5 Надання дозвілу на під'єднання

IP Address	Comment	Status	Actions
102.219.208.38/32 (includes your current IP address)	My IP Address	Active	EDIT DELETE
0.0.0.0/0 (includes your current IP address)		Active	EDIT DELETE

Рис.2.6 Перевірка доступу

Тепер можна створювати базу даних для нашого сайту інтернет магазину з продажу комп'ютерної техніки. Для цього потрібно тільки натиснути кнопку CreateDataBase і потім в ній створити потрібні нам колекції.(Рис.2.7)

Рис.2.7 Create Database

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

База даних інтернет магазину має п'ять колекцій

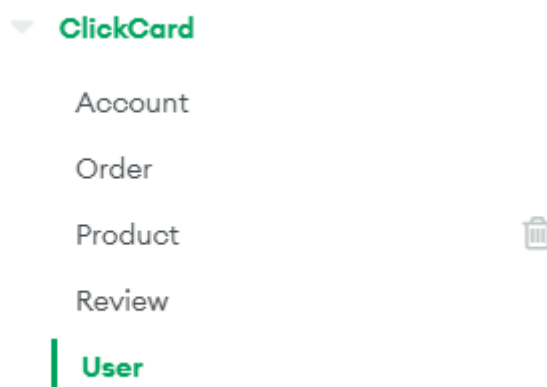


Рис.2.8 Колекції бази даних MongoDB

User (Користувач):

- Колекція, що містить дані про користувачів веб-сайту.
- Кожен документ (дані про користувача) може містити поля, такі як ім'я, електронна пошта, пароль (хеш), роль (наприклад, адміністратор або звичайний користувач), інформація про профіль тощо.

Product (Продукт):

- Колекція, що містить дані про продукти, які продаються на веб-сайті.
- Кожен документ (інформація про продукт) може включати назву продукту, опис, ціну, категорію, зображення, відгуки користувачів тощо.

Review (Відгук):

- Колекція, що зберігає відгуки користувачів про продукти.
- Кожен документ може містити інформацію, таку як ідентифікатор продукту, оцінка (зірки), текстовий відгук, ім'я автора відгуку, дату тощо.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Account (Обліковий запис):

- Колекція, що містить інформацію про облікові записи користувачів.
- Це може включати дані про вхідні дані (логін, пароль), налаштування безпеки, інформацію про статус активності облікового запису тощо.
- **Order (Замовлення):**
- Колекція, що зберігає дані про замовлення користувачів.
- Кожен документ може містити інформацію, таку як ідентифікатор замовлення, товари, кількість, вартість, статус (нове, оброблюється, виконане), адреса доставки, дані користувача, який зробив замовлення, тощо.

Ці колекції в MongoDB дозволяють ефективно зберігати та управляти даними для різних сутностей веб-сайту, забезпечуючи гнучкість і швидкодію завдяки документно-орієнтованій моделі бази даних.

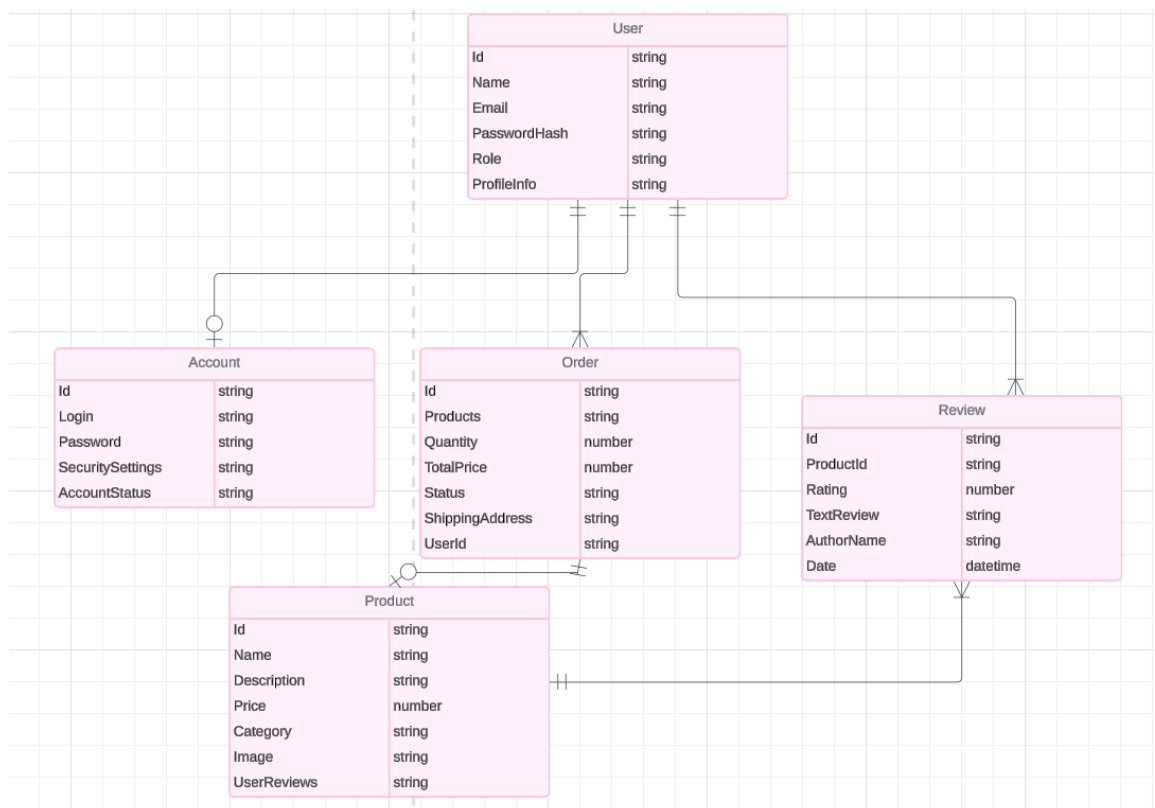


Рис.2.9 Діаграма бази даних

Завершальним кроком є під’єднання створеної бази даних до проекту.

Для цього потрібно зайти у розділ Connect на сайті ДБ MongoDB, натиснути MongoDB for VS Code та отримати посилання, яке будемо використовувати у наших налаштуваннях проекту.

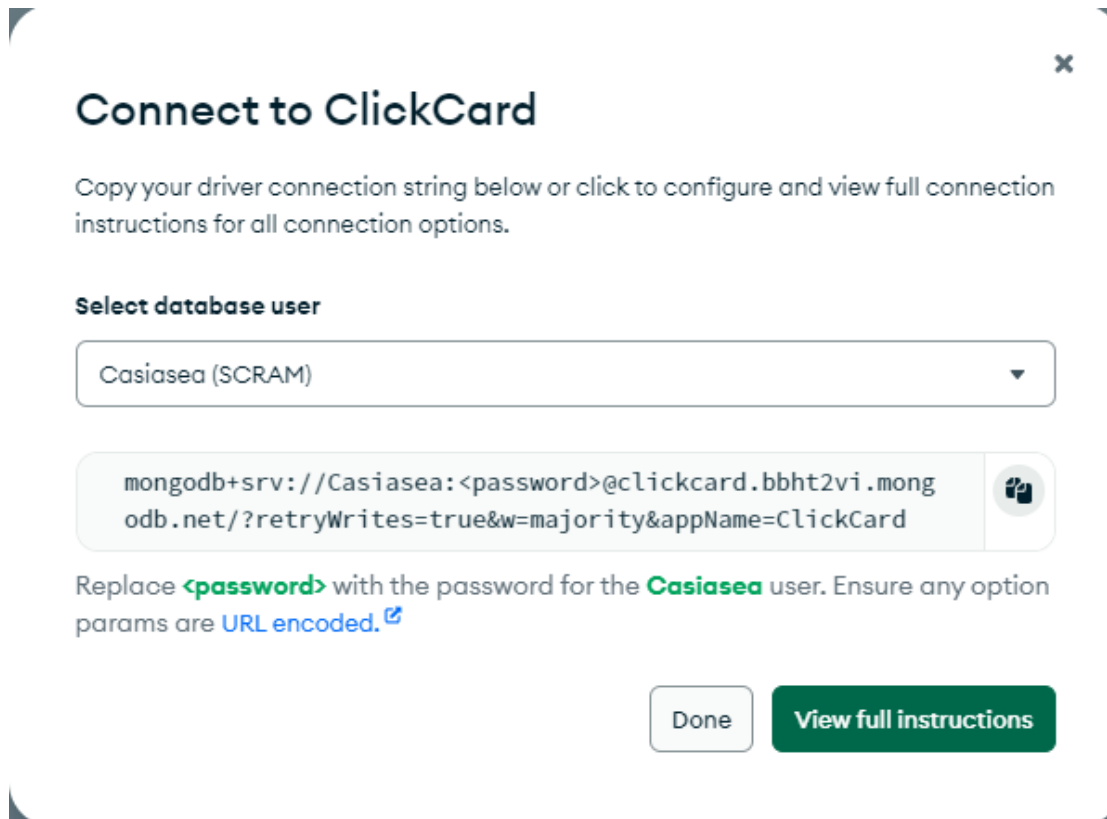


Рис.2.10 Отримання коду для з’єднання з БД

Тепер отриманий код вставляємо у створений файл конфігуратор в програмного коді. ВАЖЛИВО замість <password> потрібно ввести свій пароль при реєстрації бази даних

```
1 DATABASE_URL = "mongodb+srv://Casiasea:j8TYx3IHAIr985Lx@clickcard.bbht2vi.mongodb.net/ClickCard"
```

Рис.2.11 Під’єднання БД до проекту

2.3 Проектування та реалізація алгоритмів роботи системи

Веб-додаток дозволяє користувачам зареєструватись і увійти до своїх облікових записів. Після цього кожен зареєстрований користувач може створювати заявки на покупку товарів. Після заповнення відповідної форми, дані заявки заносяться в базу даних. Адміністратор має можливість керувати цими заявками, приймаючи або відхиляючи їх. Вартість і кількість компонентів

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

розраховуються автоматично на формі. Після успішного продажу вибрані товари відправляються на вказану користувачем адресу.

Щодо проектування та реалізації алгоритмів для веб-сайту зі збору та продажу ПК, розглянемо ключові етапи та можливості системи, як описано у діаграмі діяльності.

Головна сторінка

1. Користувач переходить на головну сторінку сайту.
2. Тут користувач може оглядати доступні товари.

Реєстрація та авторизація

1. Якщо користувач ще не зареєстрований, він створює свій обліковий запис, заповнивши просту форму.
2. Після успішної реєстрації користувач авторизується в системі.
3. У разі невдачі авторизації користувач повертається до форми реєстрації.

Створення замовлення

1. Після входу в систему користувач може скласти замовлення, обираючи компоненти для свого майбутнього ПК.
2. Після цього він здійснює оплату за замовлення.

Отримання замовлення

1. Після успішної оплати користувач отримує своє замовлення.
2. Якщо користувач не авторизувався, він також може переглянути своє замовлення.

Залишити відгук

1. Зареєстрований користувач має можливість залишити відгук про товар або послугу після отримання замовлення.

Профіль адміністратора

1. Адміністратор може переглядати й відстежувати замовлення.
2. Він аналізує статистику роботи системи.
3. Адміністратор може додавати й редагувати товари.
4. Він також має можливість переглядати коментарі, що залишили користувачі.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		


```

    for (let i = arr.length - 1; i > 0; i--) {
      const j = Math.floor(Math.random() * (i + 1));
      [arr[i], arr[j]] = [arr[j], arr[i]];
    }
    return arr;
  };

  const randomizedProducts = randomizeArray(products);

  return (
    <div className="p-8">
      <Container>
        <div>
          <HomeBanner />
        </div>
        <div className="grid grid-cols-2 sm:grid-cols-3 lg:grid-cols-4 xl:grid-
cols-5 2xl:grid-cols-6 gap-8">
          {randomizedProducts.map((product: any) => (
            <ProductCard data={product} key={product.id} />
          ))}
        </div>
      </Container>
    </div>
  );
}

```

Цей код визначає асинхронний React-компонент HomePage, який завантажує продукти на основі переданих параметрів пошуку та відображає їх у вигляді карток на головній сторінці. Якщо продукти не знайдено, відображається повідомлення "Виберіть категорію". Перед відображенням продукти перемішуються для випадкового порядку. Основні компоненти, такі як Container,

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

HomeBanner, і ProductCard, використовуються для структуризації та відображення інформації про продукти.

Далі за алгоритми веб сайту користувачу рекомендовано авторизуватися вже в створений аккаунт або зареєструвати новий. Також на сайті реалізована можливість увійти через Google.

```
interface RegisterFormProps {
  currentUser: SafeUser | null;
}

const RegisterForm: React.FC<RegisterFormProps> = ({ currentUser }) => {
  const [isLoading, setIsLoading] = useState(false);
  const { register, handleSubmit, formState: { errors } } =
useForm<FieldValues>({
  defaultValues: { name: "", email: "", password: "" },
});
  const router = useRouter();

  useEffect(() => {
    if (currentUser) {
      router.push("/cart").then(() => router.refresh());
    }
  }, [currentUser, router]);

  const onSubmit: SubmitHandler<FieldValues> = async (data) => {
    setIsLoading(true);
    try {
      await axios.post("/api/register", data);
      toast.success("Аккаунт створено");
      const callback = await signIn("credentials", { email: data.email, password:
data.password, redirect: false });
    }
  }
```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (callback?.ok) {
  router.push("/cart").then(() => {
    router.refresh();
    toast.success("Вхід виконано");
  });
} else if (callback?.error) {
  toast.error(callback.error);
}
} catch {
  toast.error("Помилка");
} finally {
  setIsLoading(false);
}
};

if (currentUser) {
  return <p className="text-center">Авторизація...</p>;
}

return (
  <>
    <Heading title="Вхід в особистий кабінет" />
    <Button outline label="Continue with Google" icon={AiOutlineGoogle}
onClick={() => signIn("google")} />
    <hr className="bg-slate-300 w-full h-px" />
    {[ "name", "email", "password" ].map((field, idx) => (
      <Input
        key={idx}
        id={field}

```

```

        label={field === "name" ? "ім'я" : field === "email" ? "Email" :
"пароль"}
        disabled={isLoading}
        register={register}
        errors={errors}
        required
        type={field === "password" ? "password" : "text"}
    />
  )))
  <Button label={isLoading ? "Завантаження" : "Зареєструватися"}
onClick={handleSubmit(onSubmit)} />
  <p className="text-sm">
    Вже є аккаунт? <Link className="underline"
href="/login">Увійти</Link>
  </p>
</>
);
};

```

export default RegisterForm;

Цей код визначає React-компонент RegisterForm, який обробляє реєстрацію нового користувача. Якщо користувач вже авторизований, він перенаправляється на сторінку кошика. Компонент включає форму для введення імені, електронної пошти та паролю, а також кнопку для реєстрації або авторизації через Google. Після успішної реєстрації виконується авторизація і перенаправлення до кошика. Компонент також відображає повідомлення про помилки або успіх реєстрації.

Аналогічно, якщо користувач вже зареєстрований він має змогу увійти в аккаунт та також автоматично сторінка перенапривить його до кошика.

```
interface LoginFormProps {
```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    currentUser: SafeUser | null;
  }

const LoginForm: React.FC<LoginFormProps> = ({ currentUser }) => {
  const [isLoading, setIsLoading] = useState(false);
  const { register, handleSubmit, formState: { errors } } =
useForm<FieldValues>({
    defaultValues: { email: "", password: "" },
  });
  const router = useRouter();

  useEffect(() => {
    if (currentUser) {
      router.push("/cart").then(() => router.refresh());
    }
  }, [currentUser, router]);

  const onSubmit: SubmitHandler<FieldValues> = async (data) => {
    setIsLoading(true);
    const callback = await signIn("credentials", { ...data, redirect: false });
    setIsLoading(false);
    if (callback?.ok) {
      router.push("/cart").then(() => {
        router.refresh();
        toast.success("Ви вже увійшли");
      });
    } else if (callback?.error) {
      toast.error(callback.error);
    }
  };
};

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (currentUser) {
  return <p className="text-center">Авторизація...</p>;
}

return (
  <>
    <Heading title="Увійти в ClickCard" />
    <Button outline label="Увійти через Google" icon={AiOutlineGoogle}
onClick={() => signIn("google")} />
    <hr className="bg-slate-300 w-full h-px" />
    {[ "email", "password"].map((field, idx) => (
      <Input
        key={idx}
        id={field}
        label={field === "email" ? "Email" : "Пароль"}
        disabled={isLoading}
        register={register}
        errors={errors}
        required
        type={field === "password" ? "password" : "text"}
      />
    ))}
    <Button label={isLoading ? "Завантаження" : "Логін"}
onClick={handleSubmit(onSubmit)} />
    <p className="text-sm">
      Не маєте аккаунту? <Link className="underline"
href="/register">Реєстрація</Link>
    </p>
  </>
)

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

```
);  
};
```

```
export default LoginForm;
```

Цей код визначає React-компонент LoginForm, який обробляє вхід користувача. Якщо користувач вже авторизований, його перенаправляють до кошика. Форма включає поля для введення електронної пошти та паролю, а також кнопку для входу або авторизації через Google. Після успішного входу користувач перенаправляється до кошика, і відображаються відповідні повідомлення про успіх або помилки.

Отже, авторизований користувач тепер має змогу передивитися товар та “покласти” його у кошик. За це відповідає наступний код :

```
const CartCount = () => {  
  const { cartTotalQty } = useCart();  
  const router = useRouter();  
  
  return (  
    <div  
      className="relative cursor-pointer"  
      onClick={() => router.push("/cart")}  

```

В кошику він також може відредагувати товари, змінити їх кількість до моменту оплати.

```
interface ItemContentProps {  
  item: CartProductType;  
}
```

```
const ItemContent: React.FC<ItemContentProps> = ({ item }) => {  
  const { handleRemoveProductFromCart, handleCartQtyIncrease,  
    handleCartQtyDecrease } = useCart();
```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

return (
  <div className="grid grid-cols-5 text-xs md:text-sm gap-4 border-t-[1.5px]
border-slate-200 py-4 items-center">
    <div className="col-span-2 flex gap-2 md:gap-4">
      <Link href={` /product/${item.id} `}>
        <div className="relative w-[70px] aspect-square">
          <Image      src={item.selectedImg.image}      alt={item.name}      fill
className="object-contain" />
        </div>
      </Link>
      <div className="flex flex-col justify-between">
        <Link href={` /product/${item.id} `}>{truncateText(item.name)}</Link>
        <div>{item.selectedImg.color}</div>
        <button   className="text-slate-500   underline"   onClick={()   =>
handleRemoveProductFromCart(item)}>
          Видалити
        </button>
      </div>
    </div>
    <div className="justify-self-center">{formatPrice(item.price)}</div>
    <div className="justify-self-center">
      <SetQuantity
        cartCounter
        cartProduct={item}
        handleQtyIncrease={() => handleCartQtyIncrease(item)}
        handleQtyDecrease={() => handleCartQtyDecrease(item)}
      />
    </div>
    <div className="justify-self-end font-semibold">
      {formatPrice(item.price * item.quantity)}
    </div>
  </div>
)

```

```

        </div>
    </div>
);
};

export default ItemContent;

const CartClient: React.FC<CartClientProps> = ({ currentUser }) => {
    const { cartProducts, handleClearCart, cartTotalAmount } = useCart();

    const router = useRouter();

    if (!cartProducts || cartProducts.length === 0) {
        return (
            <div className="flex flex-col items-center">
                <div className="text-2xl">Кошик пустий</div>
                <div>
                    <Link
                        href={"/"}
                        className="
text-slate-500 flex items-center gap-1
mt-2
"
                    >
                        <MdArrowBack />
                        <span>Почати покупки</span>
                    </Link>
                </div>
            </div>
        );
    }
}

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Цей код визначає React-компонент ItemContent, який відображає деталі продукту в кошику. Компонент включає зображення продукту, назву, колір, ціну, кількість та загальну вартість. Користувач може змінювати кількість продукту або видаляти його з кошика за допомогою відповідних функцій.

Фінальним етапом оформлення замовлення клієнт має визначитися зі способом оплати та оплатити свої покупки

```
<div className="text-sm flex flex-col gap-1 items-start">
  <div className="flex justify-between w-full text-base font-semibold">
    <span>Загальна сума</span>
    <span>{formatPrice(cartTotalAmount)}</span>
  </div>
  <p className="text-slate-500">
    Вартість доставки буде визначена під час оформлення замовлення.
  </p>
  <Button
    label={currentUser ? "Перейти до перевірки" : "Увійдіть для перевірки"}
    outline={!currentUser}
    onClick={() => {
      router.push(currentUser ? "/checkout" : "/login");
    }}
  />
</div>

<form onSubmit={handleSubmit} id="payment-form">
  <div className="mb-6">
    <Heading title="Введіть свої дані для завершення замовлення" />
  </div>
  <h2 className="font-semibold mb-2">Адреса доставки</h2>
  <AddressElement
```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

```

options={{
  mode: "shipping",
  allowedCountries: ["US", "KE"],
}}
/>
<h2 className="font-semibold mt-4 mb-2">Оплата</h2>
<PaymentElement id="payment-element" options={{ layout: "tabs" }} />
<div className="py-4 text-center text-slate-700 text-xl font-bold">
  Всього: {formattedPrice}
</div>
<Button
  label={isLoading ? "Обробка" : "Оплатити"}
  disabled={isLoading || !stripe || !elements}
  onClick={() => {}}
/>
</form>

```

Сайт ознайомлює клієнта з кінцевою сумою, вартістю доставки, робить останню перевірку чи користувач не вийшов з кабінету та його адреса доставки вірна і пропонує оплатити замовлення або продовжити покупки.

Після того як клієнту прийшов його товар, він може залишити відгук про нього та поставити оцінку від 1 до 5, де 5 це максимальний бал.

```

return (
  <div className="flex flex-col gap-2 max-w-[500px]">
    <Heading title="Ваш відгук" />
    <Rating onChange={(event, newRating) => setCustomValue('rating',
newRating)} />
    <Input
      id="review"
      label="Ваш коментар"
      disabled={isLoading}

```

```

        register={register}
        errors={errors}
        required
    />
);

    <Button label={isLoading ? "Завантаження" : 'Оцініть товар'}
onClick={handleSubmit(onSubmit)}/>
</div> );

```

Цей код відображає форму для залишення відгуку на товар. Форма включає заголовок, компонент для оцінювання (рейтинг) та поле для введення коментаря. Поле для введення коментаря вимикається, якщо форма знаходиться в стані завантаження.

Висновки до другого розділу

Отже, другий розділ проектування та реалізації алгоритмів системи є важливим етапом у створенні інформаційної системи для збору та продажу комп'ютерів. Він дозволяє детально розглянути і реалізувати ключові аспекти, необхідні для забезпечення її успішної роботи:

1. **Глибоке розуміння вимог:** Аналіз вимог дозволяє чітко визначити функціональні та нефункціональні вимоги до системи. Це включає в себе потреби користувачів, бізнес-процеси, які система повинна підтримувати, та технічні обмеження.
2. **Ефективне проектування бази даних:** Розробка структури бази даних (наприклад, використання MongoDB) є важливою для забезпечення швидкодії та надійності системи. Це включає проектування схеми даних, вибір типів даних та оптимізацію запитів.
3. **Розробка ефективних алгоритмів:** Використання оптимальних алгоритмів для управління замовленнями, обробки транзакцій, а також для реалізації інших бізнес-логік системи є ключовим для забезпечення її ефективності та швидкодії.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

4. **Вибір технологій і інструментів:** Правильний вибір технологій і інструментів розробки дозволяє забезпечити гнучкість, масштабованість та безпеку системи. Це можуть бути фреймворки для фронтенду і бекенду, бібліотеки для управління станом, інструменти для тестування та інші.

5. **Інтерфейс користувача:** Розробка інтуїтивно зрозумілого і зручного інтерфейсу є ключовим аспектом для задоволення потреб користувачів. Це включає дизайн, навігацію, взаємодію з елементами інтерфейсу та зручність використання.

Всі ці аспекти спільно допомагають створити функціональну, ефективну та високоякісну інформаційну систему, яка відповідає потребам користувачів і бізнес-вимогам у сфері продажу комп'ютерів.

.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РОЗРОБКА ІНТЕРФЕЙСУ, ТЕСТУВАННЯ БАЗИ ДАНИХ ТА ІНТЕРФЕЙСУ

3.1 Налаштування та встановлення параметрів системи

Процес встановлення та налаштування параметрів веб-сайту для збору та продажу персональних комп'ютерів включає роботу з різними компонентами, кожен з яких вимагає індивідуального розгортання та налаштування. Для візуалізації взаємозв'язків між цими компонентами використовується діаграма розгортання, що відображає структуру системи та процеси їх взаємодії.

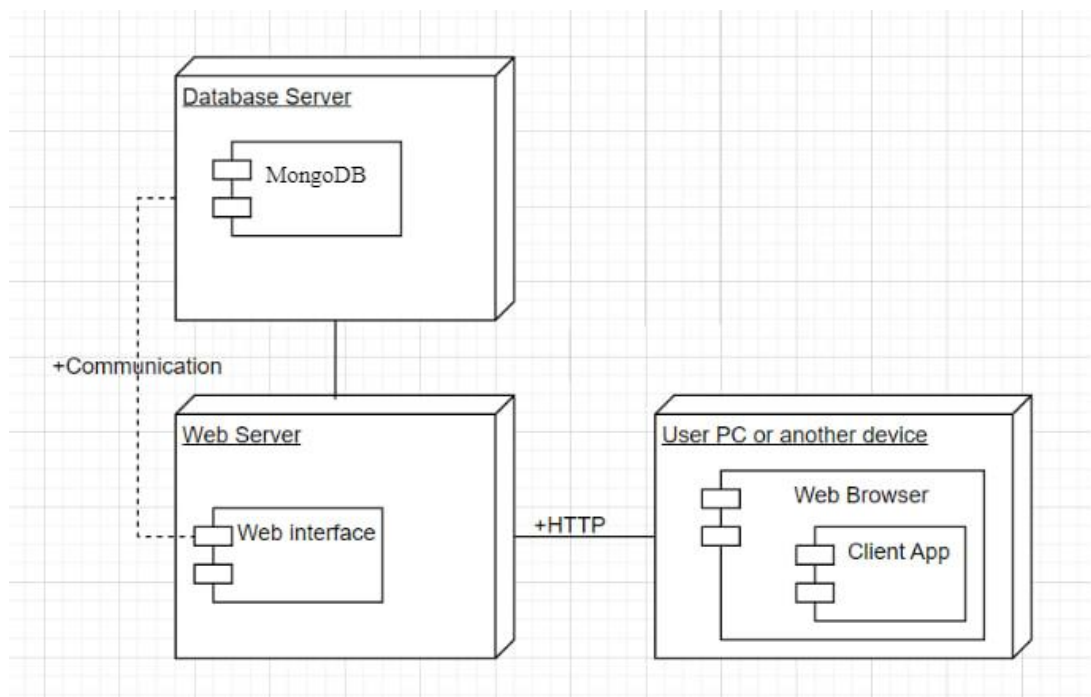


Рис.3.1 Діаграма розгортання веб-додатку для обміну цифровими активами

Ця схема розгортання ілюструє систему, що складається з трьох основних компонент: веб-сервера, сервера баз даних і клієнтського пристрою. Взаємодія клієнтського пристрою з веб-сервером здійснюється через протокол HTTP, який забезпечує обмін даними у форматі запит-відповідь на основі стандартів TCP, забезпечуючи надійну передачу даних без втрат. Взаємодія між веб-сервером і сервером баз даних відбувається через протокол TCP/IP, що забезпечує стабільний та неперервний обмін інформацією.

3.2 Інтерфейс сайту та алгоритм роботи з ним

Коли користувач потрапляє на сайт він бачить головну сторінку сайту, де розташовані рекомендовані товари (спочатку новинки). Також він бачить навігацію по сайту.

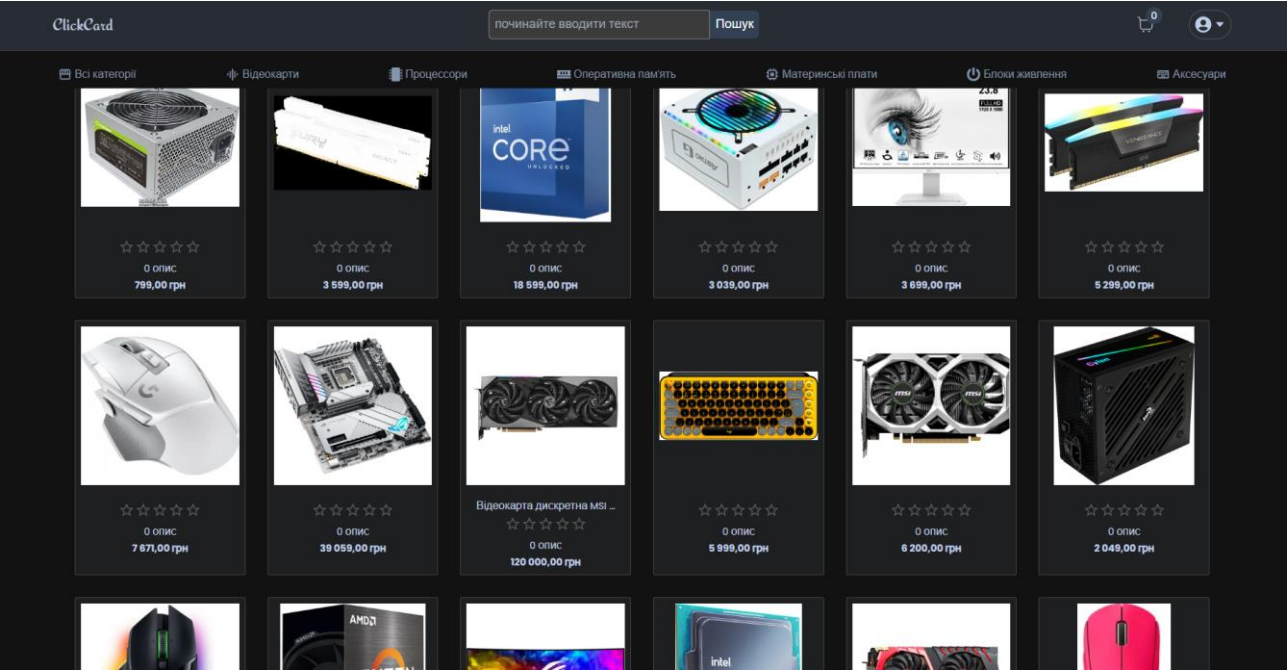


Рис. 3.2. Головна сторінка сайту

Прокроливши всі товари, користувач потрапляє в куточок споживача, де може ознайомитися з соц мережами та зворотнім зв'язком сайту

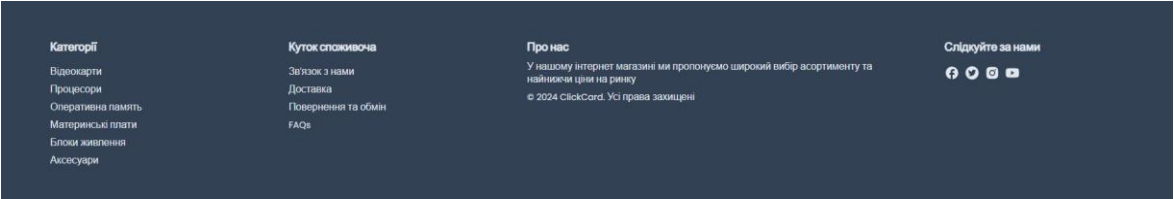


Рис.3.3 Куточок споживача

Вмотивований зробити покупки користувач має змогу зареєструватися або увійти в обліковий запис.Після успішної авторизації на сайті підтягується ім'я користувача та його аватар

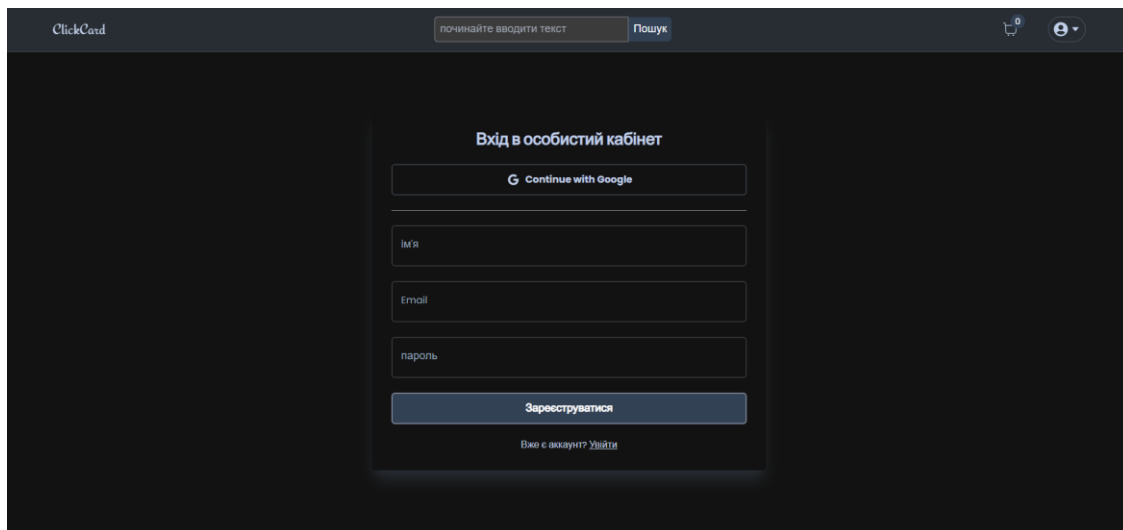


Рис 3.4 Форма реєстрації

При реєстрації у користувача є вибір : пройти реєстрацію заповнюючи поля “Ім’я”, “email” та “пароль” або продовжити через Google.

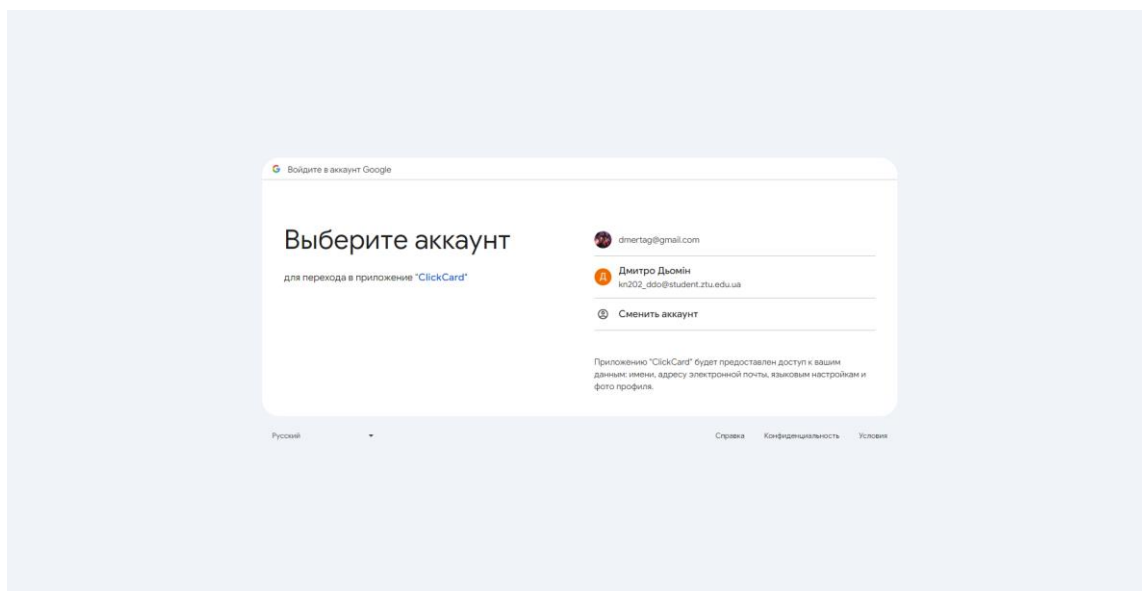


Рис 3.5 Вхід через Google

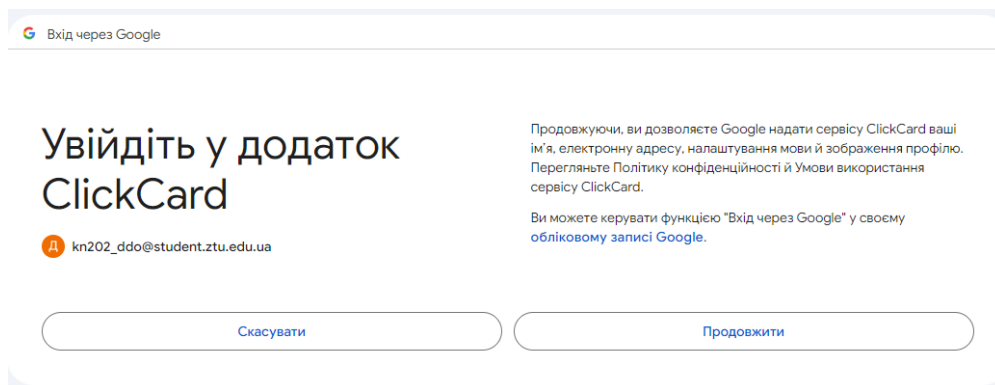


Рис 3.6 Підтвердження авторизації через Google

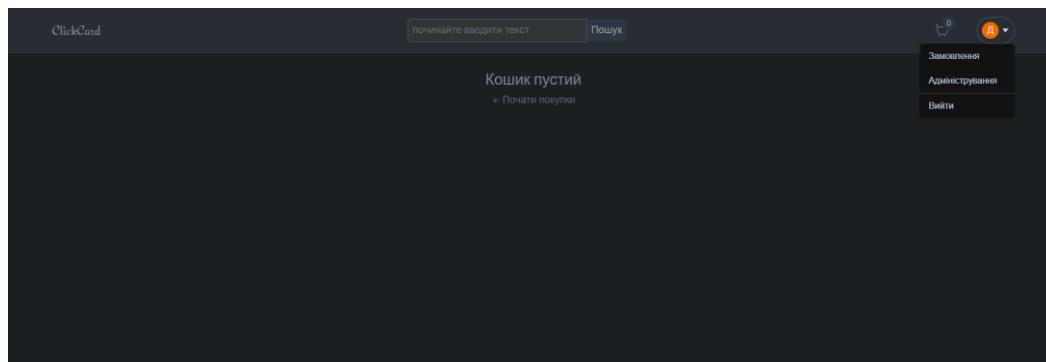


Рис 3.7 Успішний вхід

Авторизований користувач стає нашим клієнтом та має вільний доступ до кошика та списку своїх замовлень. Після чого він може вибрати по категоріям, рейтингу, або навіть кольору потрібний товар та додати його в кошик.

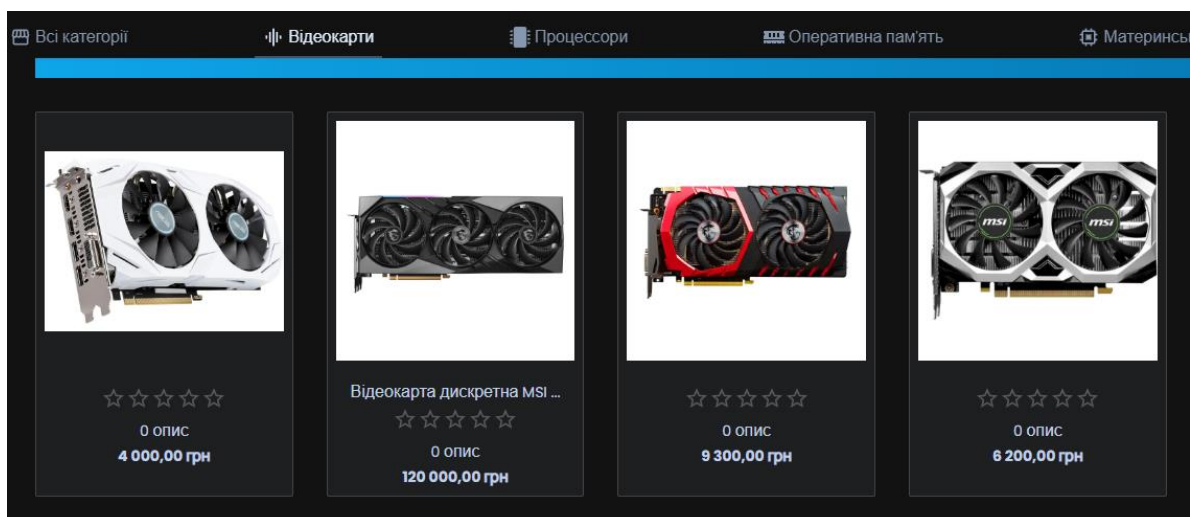


Рис 3.8 Вибір категорії

Після вибору категорії, клієнт обирає сподобавшийся йому товар, змією колір та вибирає потрібну кількість товару для замовлення

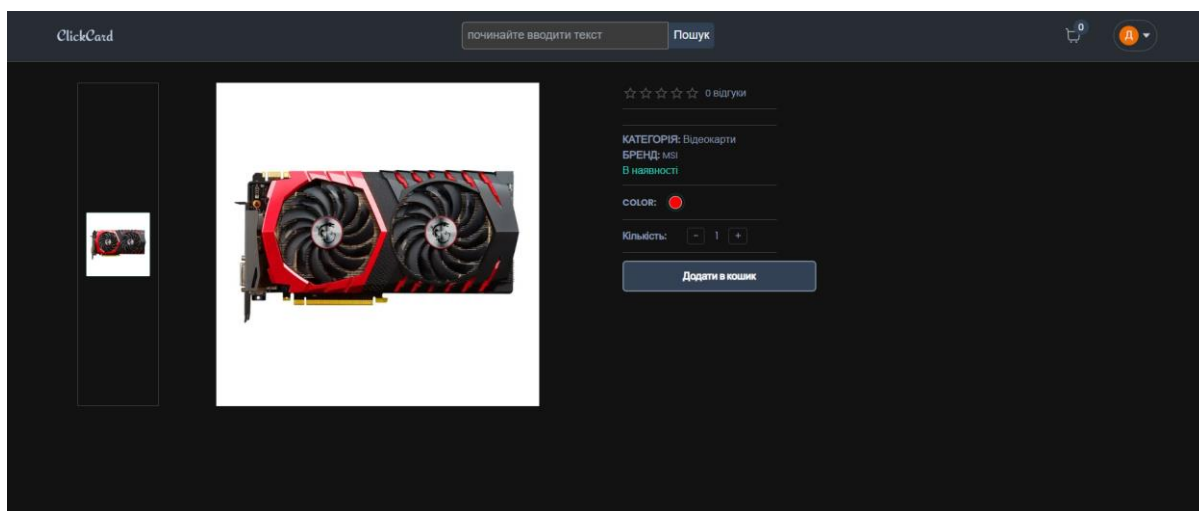


Рис. 3.9 Вибір товару

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

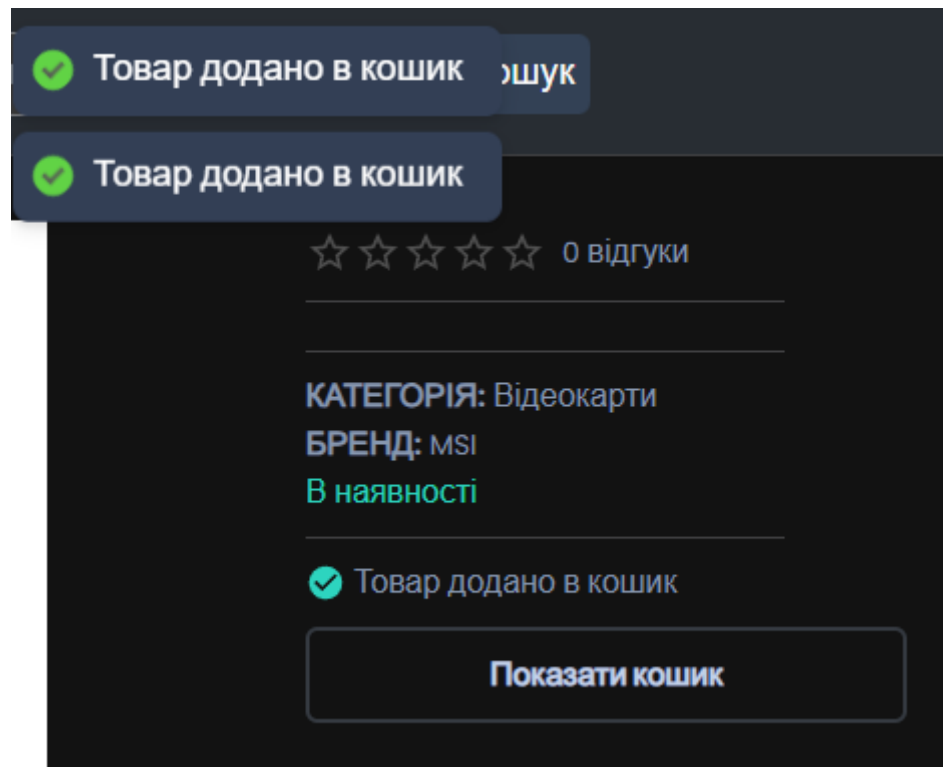


Рис 3.10 Додавання декількох товарів в кошик

Після того як клієнт визначився зі своїм замовленням він може переглянути його в кошику, оплатити та отримати його за бажаною адресою

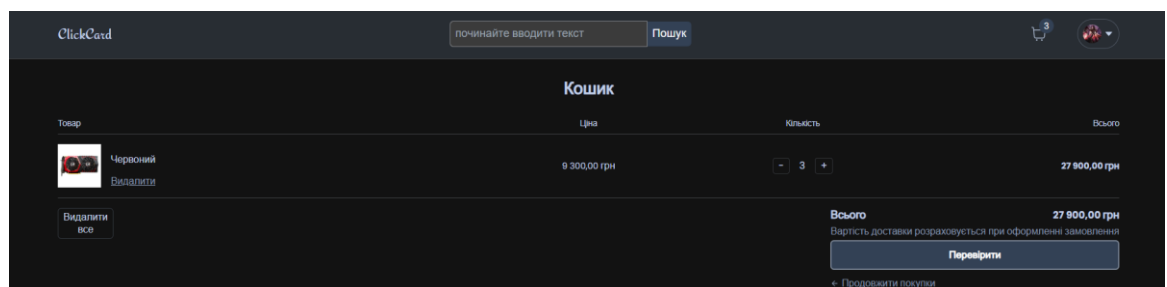


Рис.3.11 Кошик

У випадку, якщо клієнт передумав купувати товар, він може видалити його з кошика, або видалити одразу всі товари та очистити кошик

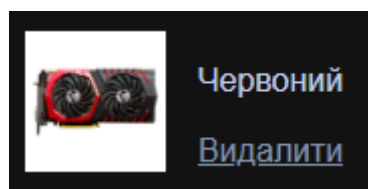


Рис. 3.12 Кнопка видалення 1 товару

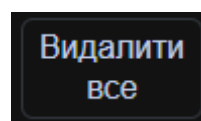


Рис. 3.13 Кнопка видалення всіх товарів з кошику

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

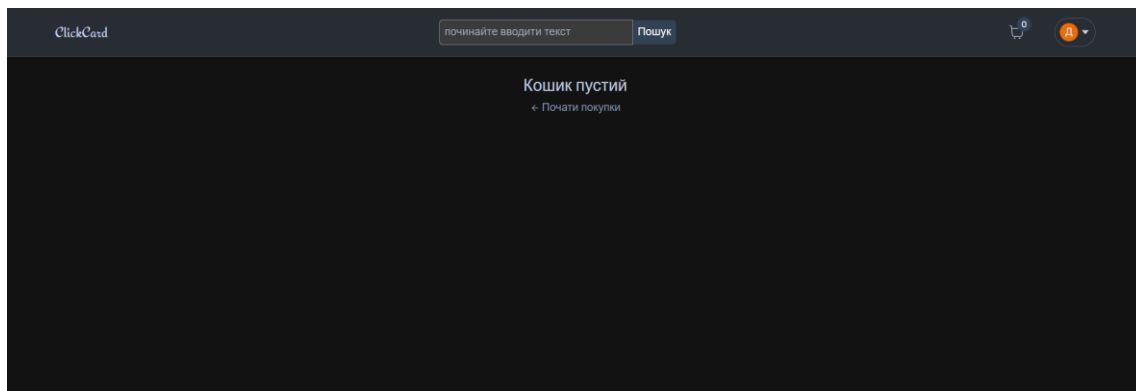


Рис. 3.14 Очищений кошик

Якщо споживач визначився з товаром у кошику і остаточно готовий його замовити, він натискає кнопку перевірити, яка перевіряє чи товар ніхто не купив пока клієнт вагався та він є на залишку.

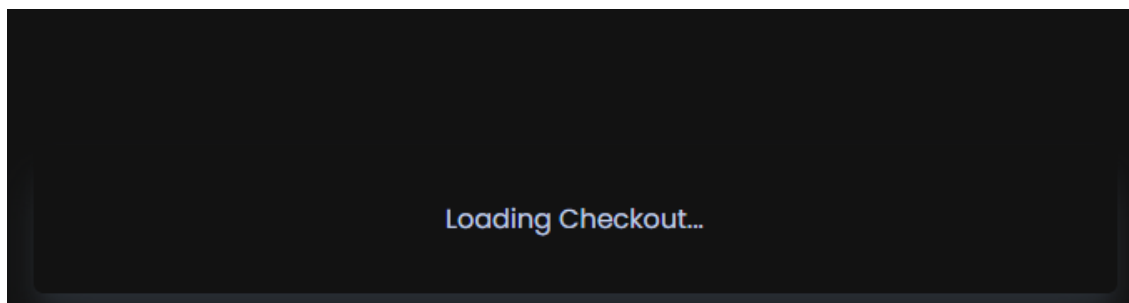


Рис. 3.15 Перевірка залишку товару на складі

Якщо перевірка успішна, клієнт потрапляє на вікно з оплатою та доставкою. Йому потрібно заповнити поля Ім'я, Прізвище, Країна, Адреса та поля рахунків банківської карти.

Рис.3.16 Оплата товару

Деталі замовлення			
Order ID: 667ddc09577b3d3c3fed5b42			
Сума: 2 790 000,00 грн			
Статус платежу:			
Статус доставки:			
Дата: 17 minutes ago			
Замовлено товару			
Товар	Ціна	QTY	Всього
 Червоний	9 300,00 грн	3	\$27900.00

Рис.3.17 Відслідковування статусу доставки

Після того як клієнт оплатив своє замовлення, в роботу вступає адміністратор сайту, котрий може переглянути замовлення, внести корективи та зв’язатися з клієнтом

Присвоєння статусу Адміністратор відбувається через базу даних MongoDB шляхом зміни значення “USER” на “ADMIN” у колекції User

User	<pre>role : "USER" _id: ObjectId('667c5cdb5d66af9510ef9d65') name : "Throw down your fears" email : "dmertag@gmail.com" emailVerified : null image : "https://lh3.googleusercontent.com/a/ACg8ocK_emRGxqCUYVC2soGHCpyA0n-Po8..." createdAt : 2024-06-26T18:24:27.252+00:00 updatedAt : 2024-06-26T18:24:27.252+00:00 role : "ADMIN"</pre>
------	--

Рис.3.18 Зміна ролі користувача

Доречі, тут можна передивитися список усіх зареєстрованих користувачів, їх ім’я, прізвище, поштову адресу, дату реєстрації та аватар.

При отриманні статусу Адміністратор користувач має доступ до статистики, де відображається інформація про кількість товарів на сайті, замовлень, оплачених та не оплачених замовлень, проданих товарів. Також є графік продажів по днях і можна дізнатися який день тижня самий прибутковий.

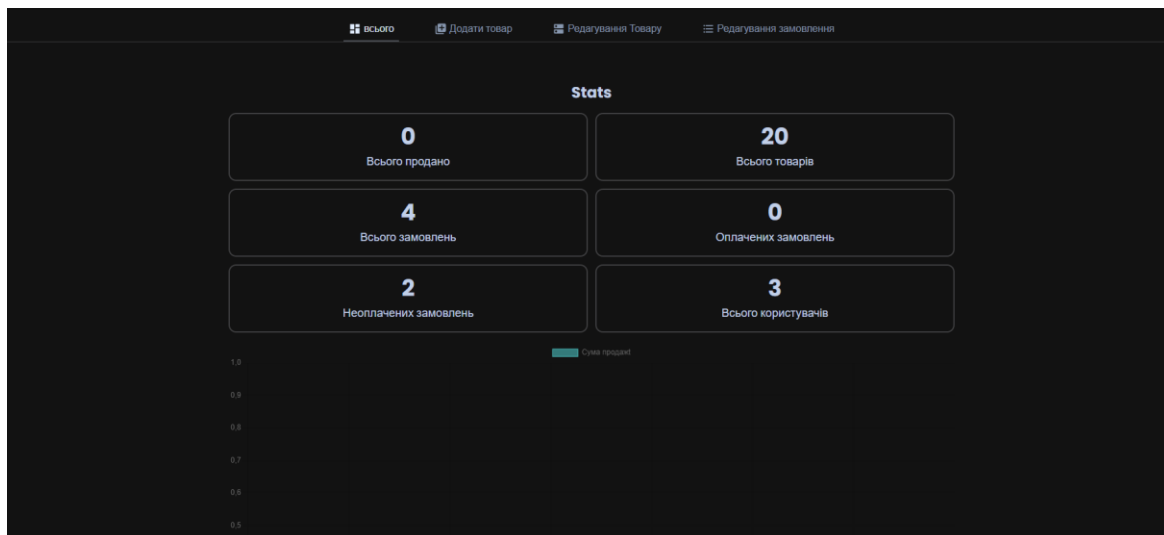


Рис.3.19 Статистика сайту

Для змоги перегляду та редагування замовлень клієнтів, адміністратор має перейти у вкладку “Редагування замовлення”

Керування замовленнями							
☐	ID	Ім'я покупця	сума(UAH)	Статус платежу	Статус доставки	дата	Дія
☐	667ddc09577b3d3c3fea5b42		27 900,00 грн				🔍 ✓ 👁
☐	667ddc09577b3d3c3fea5b41		27 900,00 грн				🔍 ✓ 👁
☐	667c68b25d66af9510ef9d69		240 000,00 грн	в очікуванні 🕒			🔍 ✓ 👁
☐	667c68b25d66af9510ef9d68		240 000,00 грн	в очікуванні 🕒			🔍 ✓ 👁
Rows per page: 9 ▾ 1-4 of 4 < >							

Рис.3.20 Зміна статусу замовлень

Керування замовленнями							
☐	ID	Ім'я покупця	сума(UAH)	Статус платежу	Статус доставки	дата	Дія
☐	667ddc09577b3d3c3fea5b42		27 900,00 грн		доставлено ✓		🔍 ✓ 👁

Рис.3.21 Успішна доставка замовлення

Також робота адміністратора полягає в постійному оновленні товару на сайті. Він має право додавати або видаляти товар з магазину, змінювати наявність на складі та обирати колір і категорію пристрою

Додати товар

Назва
Відеокарта Asus PCI-Ex GeForce RTX 3060 ROG Strix Gaming OC V2 LI

ціна
17468

бренд
Asus

OC Mode: 1912 МГц

Gaming Mode: 1882 МГц

Обсяг пам'яті

12 ГБ

Система охолодження

Axial-tech Fan Design

☐ Немає в наявності

Оберіть категорію

Всі категорії

Відеокарти

Процесори

Оперативна пам'ять

Материнські плати

Блоки живлення

Рис.3.22 Додавання товару на сайт

Для завантаження зображення товару потрібно вибрати колір та з'явиться можливість додавання фото

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Виберіть доступні кольори продукту та завантажте їх зображення.
Ви повинні завантажити зображення для кожного кольору, інакше ваш вибір кольору буде проігноровано.

☐ Білий	☒ Чорний	392376348.webp	Відміна
☐ Срібний	☐ Сірий		
☐ Червоний	☐ Золотий		
☐ Синій	☐ Графіт		

Додати товар

Рис.3.23 Додавання зображення товару

Редагувати товар						
☐	ID	Назва	ціна(UAH)	Категорія	Бренд	В наявності
☐	667c60e25d66af9510ef9d67		120 000,00 грн			Немає в наявності
☐	667c8a265d66af9510ef9d6a		4 000,00 грн			В наявності ✓

Рис.3.24 Зміна статусу товару

Товар успішно завантажений на сайт. Його можна знайти по категорії, кольору, або ціні. По замовчуванню у товару немає відгуків, так як його ще ніхто не купував

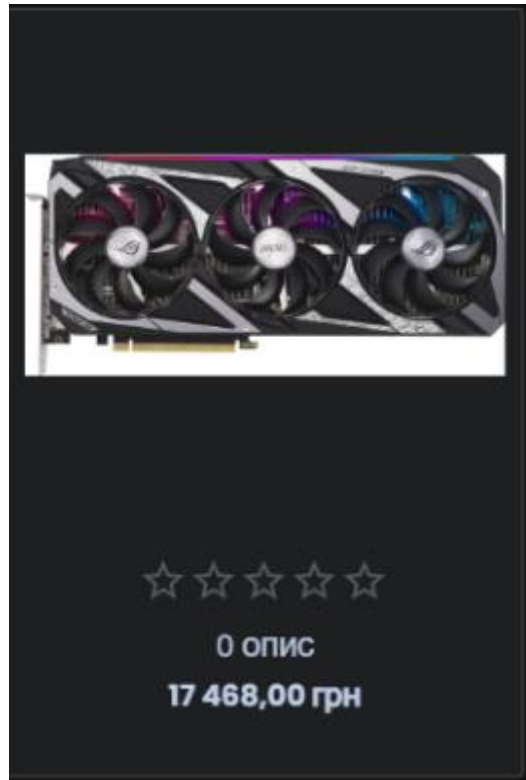


Рис.3.25 Доданий товар

3.3 Тестування роботи сайту

Тестування роботи сайту є критично важливим етапом для забезпечення його надійності, ефективності і відповідності вимогам користувачів. Ось деякі типи тестів, які можуть бути застосовані для сайту зі збору та продажу персональних комп'ютерів:

1. Функціональні тести:

Тести інтерфейсу користувача (UI): Перевірка коректності та зручності взаємодії з елементами веб-інтерфейсу, включаючи перевірку кнопок, форм, меню та інших елементів.

Тести функціональності: Перевірка виконання основних функцій, таких як реєстрація користувачів, створення замовлень, обробка платежів і т.д.

2. Тести продуктивності:

Тести навантаження: Вимірювання, як система веде себе при підвищеній навантаженості, наприклад, при одночасному доступі великої кількості користувачів.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

Тести швидкодії: Вимірювання часу відклику сервера на запити користувачів, перевірка оптимізації завантаження сторінок і т.д.

3. Тести безпеки:

Тести на вразливості: Перевірка на наявність потенційних вразливостей, таких як SQL-ін'єкції, XSS-атаки, перехоплення сесій і т.д.

Тести автентифікації і авторизації: Перевірка правильності реалізації процесів входу в систему та доступу до різних функціональних можливостей.

4. Тести сумісності:

Тести на різних пристроях і браузерах: Перевірка коректності відображення і роботи сайту на різних пристроях (комп'ютери, планшети, мобільні телефони) та в різних браузерах (Chrome, Firefox, Safari, Edge і т.д.).

5. Тести відновлення і відновлення даних:

Тести резервного копіювання: Перевірка правильності процесів резервного копіювання і відновлення даних в разі аварійної ситуації або втрати даних.

Ці типи тестів допомагають забезпечити якість роботи сайту, його стабільність і безпеку, а також зручність взаємодії з користувачами під час різних умов експлуатації.

Висновки до третього розділу

У цьому розділі було розглянуто процес налаштування та встановлення параметрів системи для веб-сайту. Цей етап важливий для забезпечення стабільної та ефективної роботи системи, а також для готовності до подальших етапів розробки та тестування.

Був розглянутий інтерфейс веб-сайту для користувачів і адміністраторів, а також основні алгоритми його роботи. Цей розділ дозволяє глибше розібратися в процесах взаємодії з користувачем та забезпечує важливий функціонал для веб-сайту з продажу комп'ютерів.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

Тестування є ключовим етапом у визначенні якості і надійності веб-сайту. В цьому розділі були розглянуті різні типи тестів. Ці етапи тестування сприяють підвищенню надійності і безпеки сайту, а також забезпечують зручність взаємодії з користувачами в різних умовах. Це є важливим кроком у створенні і підтримці високоякісного веб-сайту для продажу комп'ютерів, забезпечуючи його ефективну роботу і відповідність вимогам користувачів.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було створено сайт з продажу комп'ютерної техніки та аксесуарів на мові програмування JavaScript

Було проведено аналіз актуальних даних щодо використання інформаційних технологій у сфері інтернет-магазинів комп'ютерної техніки. Результати аналізу дозволили визначити оптимальні технології та підходи для розробки проекту.

Реалізовано інтерфейс користувача і адміністративного інтерфейсу інтернет-магазину комп'ютерної техніки, що відповідають сучасним UX/UI стандартам. Інтерфейси розроблені з урахуванням зручності користувачів та потреб адміністраторів.

Розроблено концептуальні та технічні аспекти інтернет-магазину, включаючи архітектурні рішення, вибір технологій для фронтенду та бекенду, структуру бази даних, а також визначення функціональних вимог і особливостей проекту.

Були проведені тести функціональності інтерфейсів користувача та адміністративної панелі, а також тестування бази даних для забезпечення надійності і стабільності роботи системи. Тести включали перевірку роботи основних функцій сайту, обробку і збереження даних, а також сценарії взаємодії з користувачами та адміністраторами. Результати тестування підтвердили відповідність системи вимогам проекту і виявили та виправили потенційні проблеми з функціональністю і продуктивністю.

В цілому, ця робота демонструє розуміння процесів розробки веб-проектів інтернет-магазинів, зокрема важливість аналізу потреб користувачів, ефективного проектування і реалізації програмних рішень, а також високого стандарту у тестуванні та забезпеченні якості продукту.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. React.js [Електронний ресурс]. Режим доступу до ресурсу:
<https://uk.reactjs.org/docs/getting-started.html>
2. CSS [Електронний ресурс]. Режим доступу до ресурсу:
<https://devdocs.io/css/>
3. MongoDB [Електронний ресурс]. Режим доступу до ресурсу:
<https://cloud.mongodb.com/>
4. Draw io [Електронний ресурс]. Режим доступу до ресурсу:
<https://app.diagrams.net/>
5. Rozetka [Електронний ресурс] – Режим доступу до ресурсу:
<https://rozetka.com.ua/ua/>.
6. Освітній портал [Електронний ресурс] – Режим доступу до ресурсу:
<https://learn.ztu.edu.ua>.
7. VS Code [Електронний ресурс] – Режим доступу до ресурсу:
<https://code.visualstudio.com/download>.
8. Dou [Електронний ресурс] – Режим доступу до ресурсу:
<https://dou.ua/lenta/articles/language-rating-2023>
9. Zona51 [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.Zona51.com>
10. Elmir [Електронний ресурс] – Режим доступу до ресурсу:
<https://elmir.ua>
11. Brain [Електронний ресурс] – Режим доступу до ресурсу:
<https://brain.com.ua/ukr/>
12. Prisma [Електронний ресурс]. Режим доступу до ресурсу:
<https://www.prisma.io/docs/orm/overview/databases/sqlite>

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

Код авторизації на сайті

```

interface LoginFormProps {
  currentUser: SafeUser | null;
}

const LoginForm: React.FC<LoginFormProps> = ({ currentUser }) => {
  const [isLoading, setIsLoading] = useState(false);
  const { register, handleSubmit, formState: { errors } } =
useForm<FieldValues>({
  defaultValues: {
    email: "",
    password: "",
  },
});

const router = useRouter();

useEffect(() => {
  if (currentUser) {
    router.push("/кошик");
    router.refresh();
  }
}, [currentUser]);

const onSubmit: SubmitHandler<FieldValues> = async (data) => {
  setIsLoading(true);
  try {
    const callback = await signIn("облікові дані", { ...data, redirect: false });
    if (callback?.ok) {

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        router.push("/кошик");
        router.refresh();
        toast.success("Ви вже увійшли");
    }
    if (callback?.error) {
        toast.error(callback.error);
    }
} catch (error) {
    toast.error("Помилка");
} finally {
    setIsLoading(false);
}
};

if (currentUser) {
    return <p className="text-center">Авторизація...</p>;
}

return (
    <>
        <Heading title="увійти в ClickCard" />
        <Button
            outline
            label="Увійти через Google"
            icon={AiOutlineGoogle}
            onClick={() => {
                signIn("google");
            }}
        />
        <hr className="bg-slate-300 w-full h-px" />

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<Input
  id="email"
  label="Email"
  disabled={isLoading}
  register={register}
  errors={errors}
  required
/>

<Input
  id="password"
  label="Пароль"
  disabled={isLoading}
  register={register}
  errors={errors}
  required
  type="password"
/>

<Button
  label={isLoading ? "Завантаження" : "Логін"}
  onClick={handleSubmit(onSubmit)}
/>

<p className="text-sm">
  Не маєте аккаунту?{" "}
  <Link className="underline" href="/Реєстрація">
    Реєстрація
  </Link>
</p>
</>

);

};

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг футера

```

const Footer = () => {
  const currentYear = new Date().getFullYear();

  return (
    <footer className="bg-slate-700 text-slate-200 text-sm mt-16">
      <Container>
        <div className="flex flex-col md:flex-row justify-between pt-16 pb-8">
          <FooterSection title="Категорії">
            <Link href="#">Відеокарти</Link>
            <Link href="#">Процесори</Link>
            <Link href="#">Оперативна пам'ять</Link>
            <Link href="#">Материнські плати</Link>
            <Link href="#">Блоки живлення</Link>
            <Link href="#">Акcesуари</Link>
          </FooterSection>
          <FooterSection title="Куток споживача">
            <Link href="#">Зв'язок з нами</Link>
            <Link href="#">Доставка</Link>
            <Link href="#">Повернення та обмін</Link>
            <Link href="#">FAQs</Link>
          </FooterSection>
          <div className="w-full md:w-1/3 mb-6 md:mb-0">
            <h3 className="text-base font-bold mb-2">Про нас</h3>
            <p className="mb-2">
              У нашому інтернет-магазині ми пропонуємо широкий вибір
              асортименту та найнижчі ціни на ринку.
            </p>
            <p>&copy; {currentYear} ClickCard. Усі права захищені.</p>

```

```

    </div>
    <FooterSection title="Слідкуйте за нами">
      <SocialLink href="#" icon={<MdFacebook size={24} />} />
      <SocialLink href="#" icon={<AiFillTwitterCircle size={24} />} />
      <SocialLink href="#" icon={<AiFillInstagram size={24} />} />
      <SocialLink href="#" icon={<AiFillYoutube size={24} />} />
    </FooterSection>
  </div>
</Container>
</footer>
);
};

const FooterSection = ({ title, children }) => (
  <div className="w-full md:w-1/4 mb-6 md:mb-0">
    <h3 className="text-base font-bold mb-2">{title}</h3>
    {children}
  </div>
);

const SocialLink = ({ href, icon }) => (
  <Link href={href} className="inline-block">
    {icon}
  </Link>
);

```

Лістинг кошика

```

const CartClient: React.FC<CartClientProps> = ({ currentUser }) => {
  const { cartProducts, handleClearCart, cartTotalAmount } = useCart();
  const router = useRouter();

  if (!cartProducts || cartProducts.length === 0) {
    return (
      <div className="flex flex-col items-center">
        <div className="text-2xl">Кошик порожній</div>
        <div>
          <Link
            href={"/"}
            className="text-slate-500 flex items-center gap-1 mt-2"
          >
            <MdArrowBack />
            <span>Почати покупки</span>
          </Link>
        </div>
      </div>
    );
  }

  return (
    <div>
      <Heading title="Кошик" center />
      <div className="grid grid-cols-5 text-xs gap-4 pb-2 items-center mt-8">
        <div className="col-span-2 justify-self-start">Товар</div>
        <div className="justify-self-center">Ціна</div>
        <div className="justify-self-center">Кількість</div>
      </div>
    </div>
  );
};

```

```

    <div className="justify-self-end">Всього</div>
  </div>
  <div>
    {cartProducts.map((item) => (
      <CartItem key={item.id} item={item} />
    ))}
  </div>
  <div className="border-t-[1.5px] border-slate-200 py-4 flex justify-between gap-4">
    <div className="w-[90px]">
      <Button
        label="Очистити кошик"
        onClick={handleClearCart}
        small
        outline
      />
    </div>
    <div className="text-sm flex flex-col gap-1 items-start">
      <div className="flex justify-between w-full text-base font-semibold">
        <span>Всього</span>
        <span>{formatPrice(cartTotalAmount)}</span>
      </div>
      <p className="text-slate-500">
        Вартість доставки розраховується при оформленні замовлення
      </p>
      <Button
        label={currentUser ? "Перевірити" : "Увійдіть, щоб перевірити"}
        outline={!currentUser}
        onClick={() =>
          currentUser ? router.push("/checkout") : router.push("/login")}
      >
    </div>
  </div>

```

```

    }
  />
  <Link
    href={"/"}
    className="text-slate-500 flex items-center gap-1 mt-2"
  >
    <MdArrowBack />
    <span>Продовжити покупки</span>
  </Link>
</div>
</div>
</div>
);
};

```

```

const CartItem: React.FC<{ item: CartProductType }> = ({ item }) => {
  const { handleRemoveProductFromCart, handleCartQtyIncrease,
handleCartQtyDecrease } = useCart();

```

```

  return (
    <div className="grid grid-cols-5 text-xs md:text-sm gap-4 border-t-[1.5px]
border-slate-200 py-4 items-center">
      <div className="col-span-2 justify-self-start flex gap-2 md:gap-4">
        <Link href={`\product/${item.id}`}>
          <div className="relative w-[70px] aspect-square">
            <Image
              src={item.selectedImg.image}
              alt={item.name}
              fill
              className="object-contain"

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        />
      </div>
    </Link>
    <div className="flex flex-col justify-between">
      <Link href={` /product/${item.id}`}>{truncateText(item.name)}</Link>
      <div>{item.selectedImg.color}</div>
      <div className="w-[70px]">
        <button className="text-slate-500 underline" onClick={() =>
handleRemoveProductFromCart(item)}>
          Видалити
        </button>
      </div>
    </div>
  </div>
  <div className="justify-self-center">{formatPrice(item.price)}</div>
  <div className="justify-self-center">
    <SetQuantity
      cartCounter={true}
      cartProduct={item}
      handleQtyIncrease={() => handleCartQtyIncrease(item)}
      handleQtyDecrease={() => handleCartQtyDecrease(item)}
    />
  </div>
  <div className="justify-self-end font-semibold">
    {formatPrice(item.price * item.quantity)}
  </div>
</div>
);
};

```