

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

## Кваліфікаційна робота

на здобуття ступеня вищої освіти «Бакалавр»

«Розробка маркетингу з використанням технологій React та Node.js»

(тема кваліфікаційної роботи українською мовою)

«Development of a Marketplace Using React and Node.js Technologies»

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

Коломієць Євген Олександрович

(прізвище, ім'я, по-батькові здобувача)

Керівник к.т.н., доцент Фразе-Фразенко О.О.

(науковий ступінь, вчене звання, прізвище, ініціали)



(підпис)

Рецензент к.т.н., доцент Щербіна Ю.В.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:  
Протокол засідання кафедри  
Інформаційних технологій

№ 1 від 09 червня 2024 р.

Завідувачка кафедри

  
(підпис) КАЗАКОВА Надія  
(прізвище, ім'я)

Захищено на засіданні ЕК № 13  
протокол № 18 від 20 червня 2024 р.

Оцінка відмінно / A / 90  
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

  
(підпис) КОПИЧЕНКО Іван  
(прізвище, ім'я)

Одеса 2024

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	5
ВСТУП .....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ.....	9
1.1 Дослідження предмета та особливостей онлайн комерції .....	9
1.2 Аналіз існуючих рішень .....	12
1.3 Аналіз технологій великих інтернет-магазинів .....	15
1.4 Постановка завдання для розробки.....	16
2 ВИБІР АРХІТЕКТУРИ СИСТЕМИ ТА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ .....	18
2.1 Вибір мови програмування та менеджера пакетів .....	18
2.2 Вибір засобів для серверної розробки .....	22
2.3 Вибір технологій для клієнтської частини .....	23
2.4 Вибір архітектури мережевого WEB-ресурсу .....	25
2.5 Вибір системи управління базами даних .....	30
2.6 Вибір системи управління версіями та середовище розробки.....	34
3 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ .....	38
3.1 Проектування інтернет-магазину за допомогою методології функціонального моделювання SADT .....	38
3.2 Проектування мережевого WEB-ресурсу за допомогою методології Workflow Diagramming .....	44
3.3 Проектування мережевого WEB-ресурсу за допомогою методології потоків даних DFD .....	47
3.4 Проектування бази даних системи.....	49
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ .....	55
4.1 Керівництво додатком користувача-клієнта системи .....	55
4.2 Реалізація додатку користувача-адміністратора системи .....	62
ВИСНОВКИ.....	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	67

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

HTML – Hyper Text Markup Language – мова гіпертекстової розмітки, використовується для створення структури веб-сторінок.

CSS – Cascading Style Sheets – каскадні таблиці стилів, використовуються для визначення зовнішнього вигляду веб-сторінок.

JavaScript (JS) – динамічна, об'єктно-орієнтована мова програмування, використовується для створення інтерактивних елементів на веб-сторінках.

MySQL – система управління базами даних (СУБД), часто використовується для зберігання та управління даними в веб-додатках.

AJAX – Asynchronous JavaScript and XML – технологія, що дозволяє асинхронно взаємодіяти з сервером, не перезавантажуючи всю сторінку.

API – Application Programming Interface – інтерфейс програмування додатків, який дозволяє взаємодіяти між програмами чи сервісами.

CMS – Content Management System – система управління контентом, призначена для створення та редагування веб-сайтів без необхідності програмування.

SSL – Secure Sockets Layer – протокол захищеної передачі даних в мережі Інтернет, що забезпечує шифрування комунікації між веб-сервером і браузером.

DNS – Domain Name System – система доменних імен, яка перетворює зрозумілі для людини доменні імена в IP-адреси комп'ютерів.

FTP – File Transfer Protocol – протокол передачі файлів, який використовується для передачі файлів між комп'ютерами у мережі.

SEO – Search Engine Optimization – оптимізація для пошукових систем, набір стратегій та технік, що допомагають підвищити видимість веб-сайту в результатах пошуку.

CDN – Content Delivery Network – мережа доставки контенту, яка забезпечує швидку доставку веб-контенту за допомогою розподілених серверів.

MVC – Model-View-Controller – шаблон проектування програмного забезпечення, що розділяє додаток на три основні компоненти: модель, представлення та контролер.

UX – User Experience – користувацький досвід, який описує взаємодію користувача з веб-сайтом або додатком та його задоволення від цієї взаємодії.

## ВСТУП

Інтернет-торгівля в Україні стрімко розвивається і не втрачає динаміки. За підсумками минулого року, наша країна випередила всі інші європейські країни за темпами зростання інтернет-продажів, які склали 35%. Інтернет давно став невід'ємною частиною нашого життя, і багато підприємців, місцевих бізнесів, а також великі компанії вже активно працюють на цьому ринку. Це відкриває можливості не тільки для місцевої торгівлі, але й для виходу на національний та навіть світовий ринки. Проте, твердження, що інтернет-комерція повністю витісняє традиційну, є лише частково правильним: високі витрати на оренду, утримання персоналу, економічна криза та інші фактори впливають на класичну торгівлю. Однак більшість ритейлерів намагаються створити багатоканальні системи збуту, і в деяких галузях відсутність інтернет-магазинів може стати стратегічним недоліком.

Інтернет-магазин – це платформа в мережі, де відбувається безпосередній продаж товарів споживачам, як юридичним, так і фізичним особам, з урахуванням доставки. Всі процеси – від розміщення інформації про товар до оформлення замовлення і укладення угоди – проходять в інтернеті на сайті магазину.

Веб-ресурси мають багато переваг перед традиційними магазинами, зокрема:

1. Зручність: Інтернет-магазини працюють цілодобово, що дозволяє робити покупки у будь-який час, навіть вночі або у вихідні.
2. Широкий вибір товарів: Інтернет-магазини можуть пропонувати більший асортимент товарів, оскільки вони не обмежені фізичним простором. Ви можете знайти товари з усього світу, не виходячи з дому.
3. Порівняння цін і вибір: В Інтернеті легко порівнювати ціни на товари у різних магазинах, швидко знаходячи найкращі пропозиції без необхідності відвідувати багато магазинів.

4. Зручність доставки: Багато інтернет-магазинів пропонують швидку доставку прямо до вашого дому або офісу, що робить процес покупки ще зручнішим.
5. Відгуки і рейтинги: В Інтернеті можна переглядати відгуки та рейтинги товарів, що допомагає зробити обдуманий вибір перед покупкою.
6. Економія часу і зусиль: Покупки в Інтернеті дозволяють економити час і зусилля, оскільки не потрібно їхати до магазину, шукати місце для паркування чи стояти в чергах.
7. Зручність оплати: Багато інтернет-магазинів пропонують різні способи оплати, включаючи кредитні картки, електронні платіжні системи і навіть післяплату, що робить процес оплати максимально зручним.
8. Пошук знижок і акцій: В Інтернеті можна легко знайти знижки, акції та промокоди, що дозволяє економити гроші на покупках.

Метою кваліфікаційної роботи є створення інтернет-магазину, який буде використовуватися для надання послуг у сфері продажу техніки.

Для досягнення цієї мети необхідно вирішити наступні задачі:

- Дослідити процеси організації інтернет-торгівлі.
- Провести аналіз аналогічних сайтів для розробки технічного завдання.
- Вибрати та налаштувати інструменти реалізації.
- Визначити структуру даних та спроектувати веб-сайт.
- Реалізувати додатки та підготувати інструкції для користувачів.

Кваліфікаційна робота містить 67 сторінок, 36 рисунків, 10 таблиць, 12 джерел посилання.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

## 1.1 Дослідження предмета та особливостей онлайн комерції

Ситуація у сучасному економічному житті змушує бізнес ставити інформаційні технології частіше за все до своїх сервісів для підвищення якісних та конкурентних характеристик.

Інтернет-магазин являє собою необхідну частину сучасного комерційного світу. По-перше, ці компанії забезпечують зручність та відкритість для клієнтів. Інтернет-магазини працюють цілодобово, що надає можливість приймати рішення про покупку кожного дня незалежно від часу та місця, де це зроблено. У сучасному світі, де люди рідко ходять за покупками через зайнятість, це особливо важливо. По-друге, інтернет-магазини мають можливість пропонувати широкий вибір товарів та послуг без обмежень магазину. Асортимент товарів і послуг найчастіше є досить великим і може задовольнити вимоги практично будь-якого клієнта. По-третє, широкий географічний охоплюючий магазини надасть можливість працювати з країнами по цілому світі, що розширить спектр для клієнтів (рис. 1.1).

Більше того, інтернет-магазини надають можливість детального порівняння та ознайомлення з будь-яким товаром або послугою. Багато платформ ставлять комусь вагу, надаючи можливість порівнювати товари та послуги з різних веб-ресурсів провайдерів, рахуючи таким чином кошти, та вибирати акції. Крім того, багато магазинів пропонують документацію та рекламу товарів та послуг, відгуки та рейтинги покупців, які свідчать та оцінюють відомості про покупця.

На таких платформах споживач може забезпечити зручний вибір та мінімізувати ризик обміну. Крім того, важливим є витрати на плату та доставку, які часто оплачуються інтернет-магазинами. Широкий спектр місцевого застосування робить варіанти оплати максимально зручними для клієнта, включаючи кредитні картки, електронні гроші та післяплату.

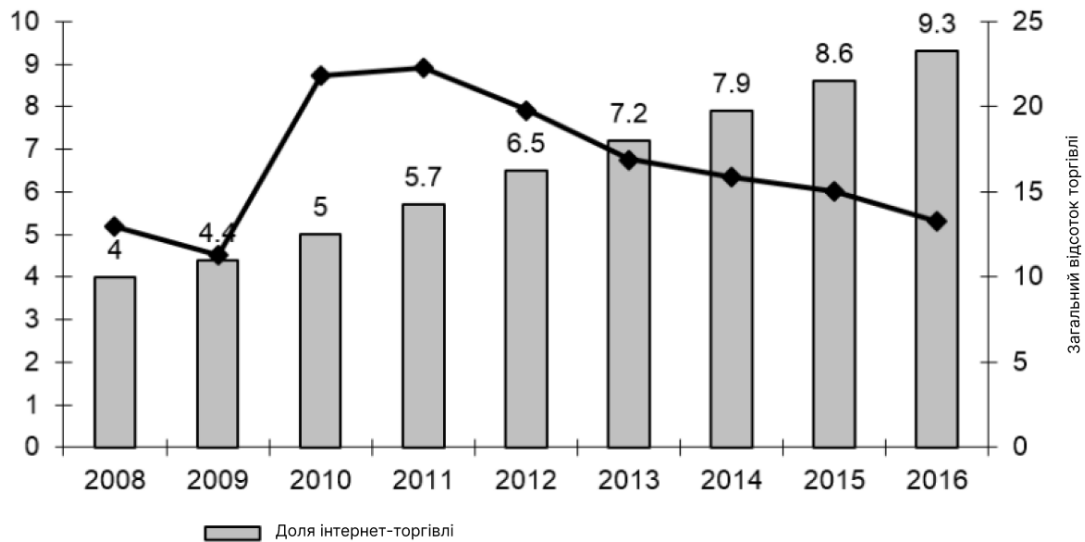


Рисунок 1.1 – Темпи росту інтернет-торгівлі (за даними ресурсу [www.reclamare.ua](http://www.reclamare.ua))

Щодо доставки, багато магазинів пропонують можливості швидкої та вірної доставки до ванни, що є додатковим надбавленням до зручності покупок онлайн. Наприкінці зручність, доступність, великий вибір та послуги обслуговування застосовуються до інтернет-магазинів, що робить їх привабливими для сучасних споживачів і незамінними для бізнесу в умовах постійної конкуренції.

Крім того, інтернет-магазини дозволяють підприємствам скоротити витрати, пов'язані з орендою приміщень, найманням персоналу та будь-які інші витрати, пов'язані з традиційними магазинами. Це, у свою чергу, сприяє низькій вартості ведення бізнесу, що відображається на конкурентоспроможних цінах на товари та послуги для споживачів.

Зі збільшенням кількості покупок, які здійснюються в Інтернеті, фірми можуть розширити свою аудиторію та розширити свої рекламні та маркетингові можливості. За допомогою цільової реклами та персоналізованих пропозицій вони з більшою ймовірністю охоплюють клієнтів, що допомагає їм ефективніше продавати свої товари та послуги. Таким чином, інтернет-магазини не тільки зручні для споживачів, але й вигідні для бізнесу.



Е-комерція вже багато років займає важливе місце в житті майже кожного українця. Незважаючи на сучасні виклики та події, електронна комерція продовжує зростати як кількісно, так і якісно.

За даними European Business Association, ринок е-комерції в Україні стабільно зростає та розвивався до лютого 2022 року, з річним ростом у 20-30% починаючи з 2018 року. Вже весною 2023 року, обсяг електронної комерції в Україні повернувся на довоєнний рівень та продовжив позитивну динаміку [12].

Український e-commerce має свої особливі тенденції та правила, що продиктовані в тому числі нашими воєнними реаліями та викликами.

Перш за все варто зазначити, що в Україні прийнятий та успішно функціонує Закон «Про електронну комерцію» №675-VIII від 03 вересня 2015 року [11]. Даний законодавчий акт встановлює загальні правила та особливості здійснення електронної торгівлі в Україні. Законом визначені вимоги до продавця. Зокрема під час своєї діяльності та у разі поширення комерційного електронного повідомлення, продавець зобов'язаний забезпечити прямий, простий, стабільний доступ інших учасників відносин у сфері електронної комерції до інформації про:

- повне найменування юридичної особи або прізвище, ім'я, по батькові фізичної особи-підприємця;
- адресу реєстрації;
- ідентифікаційний номер;
- адресу електронної пошти та/або адреса інтернет-магазину;
- відомості про ліцензію (серія, номер, строк дії та дата видачі), якщо господарська діяльність підлягає ліцензуванню;
- інформацію щодо включення податків у розрахунок вартості товару, роботи, послуги;
- у разі доставки товару - інформація про вартість доставки.

## 1.2 Аналіз існуючих рішень

Для аналізу було обрано декілька сайтів зі схожою тематикою. Усіх їх об'єднує схожий функціонал та, як для інтернет магазинів, специфіка товару.

Загалом було розглянуто:

- Rozetka;
- Comfy;
- Citrus;

Одним із таких сайтів являється ROZETKA, яка представлена на рисунку 1.2.

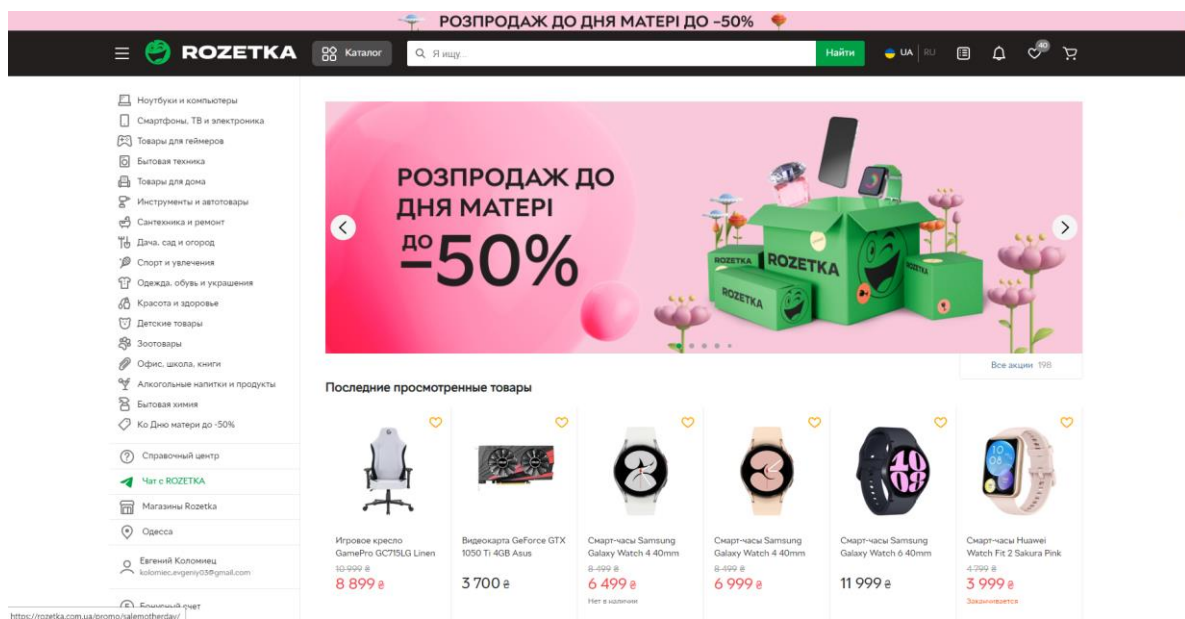


Рисунок 1.2 – Головна сторінка інтернет-магазину Rozetka

Даний інтернет-магазин має великий вибір товарів різного напрямлення. Для прикладу можна взяти: побутова хімія, дитячі товари, техніка, меблі та багато іншого. Також має комфорт на широку систему фільтрації товарів та цільового пошуку, система рекомендацій та створення «списків бажаного». Хоча багато товарів продаються від імені самого інтернет-магазину, все ж там є функціонал що дозволяє реалізовувати свої товари на ринку іншим

продавцям. Є одним із реалізаторів а також представляє деякі бренди на ринку України, наприклад Lenovo, Asus, Xiaomi і т.д. На сайті можна лишити відгук після покупки товару та задати питання для продавця, для уточнення даних про товар.

З мінусів можна виділити велику націнку як для інтернет-магазину – багато товарів можна знайти значно з меншою ціною на інших ресурсах, але Rozetka гарантує повернення товару в перші 14 днів після купівлі, якщо товар не задовільнив покупця.

Схожу тематику та призначення має сайт Comfy, його головна сторінка представлена на рисунку 1.3. Даний інтернет-магазин має великий вибір техніки та товарів призначених для неї, побутових речей та меблів. На відміну від Rozetka має функціонал для порівняння товарів між собою, що облегшує вибір товару. Ідентично має регулярні знижки, акції та дозволяє підприємцям реалізовувати свій товар в інтернеті. Також можна лишати відгук на товар, але не можна лишити питання для продавця, є можливість лише подзвонити на «гарячу лінію» для уточнення деталей.

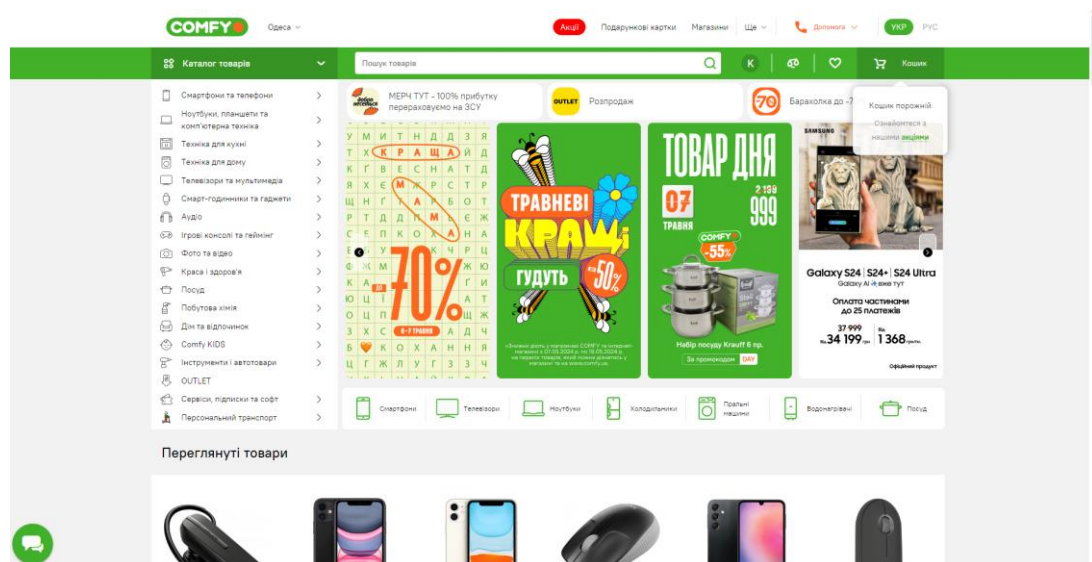


Рисунок 1.3 – Головна сторінка інтернет-магазину Comfy

Як і попередні сайти, що були проаналізовані Citrus (рисунок 1.4) має схожий функціонал та реалізацію багатьох функцій, але можна виділити, що

Citrus має голосовий пошук по всьому каталогу. Даний сайт має більш динамічний але лаконічний дизайн. Також дозволяє лишити відгук, та як і Comfy реалізовує уточнення даних через технічну підтримку.

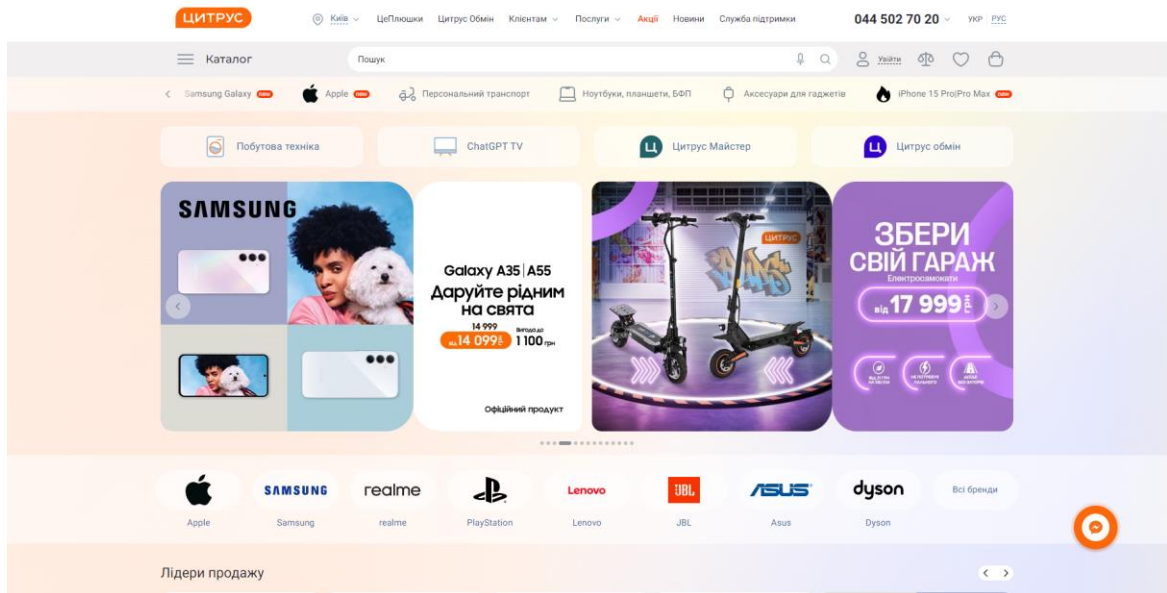


Рисунок 1.4 – Головна сторінка інтернет-магазину Citrus

Розглянемо детальніше інформацію про сайти та проведемо загальний аналіз (табл.1.1).

Таблиця 1.1 – Аналіз розглянутих сайтів для продажу техніки

Критерії	Rozetka	Comfy	Citrus
Детальна інформація про товар	+	+	+
Сучасний дизайн	+	+	+
Зворотній зв'язок	+	+	+
Швидкість доставки	+	+	-
Детальність фільтрів	+	-	-
Можливість лишати питання до товару	+	-	-
Видима цінова політика	+	+	+
Можливість реалізації для підприємців свого товару	+	-	-

### 1.3 Аналіз технологій великих інтернет-магазинів

У сучасному світі великі інтернет-магазини стали важливими гравцями на глобальному ринку, значно впливаючи на спосіб, у який люди здійснюють покупки. Вони пропонують широкий асортимент товарів та послуг, забезпечуючи зручність та швидкість процесу придбання. Основою успішної роботи таких магазинів є використання передових технологій, які дозволяють не тільки ефективно управляти бізнесом, але й забезпечувати високий рівень задоволеності клієнтів. У цьому розділі ми проведемо детальний аналіз технологій, які застосовуються найбільшими гравцями на ринку електронної комерції, такими як ROZETKA, Comfy, Citrus та інші. Розглянемо, які саме інноваційні рішення використовують ці компанії для оптимізації своїх процесів, забезпечення безпеки транзакцій, покращення обслуговування клієнтів та підвищення конкурентоспроможності. Цей аналіз допоможе нам зрозуміти, які технологічні підходи є найбільш ефективними та можуть бути використані для розробки інтернет-магазинів різного масштабу. Технології та їх порівняння наведені в табл. 1.2.

Таблиця 1.2 – Аналіз технологій розглянутих сайтів

Критерії	Rozetka	Comfy	Citrus
FrontEnd технології	Angular.js	Vue.js	React, Next.js
BackEnd технології	Node.js, TypeScript	PHP	Node.js
Інші сервіси	Hotjar, Google Analytics, Cloudflare Browser Insights	Google Analytics	Open Graph, Module Federation

## 1.4 Постановка завдання для розробки

Дипломний проект призначений для того, щоб користувач мав можливість вибрати з каталогу товар або декілька, оцінити його характеристики та придбати. Інтернет-магазин повинен бути реалізований як веб-сайт, доступний в Інтернеті. Сайт повинен поділятися на дві частини – клієнтську та серверну.

Додаток повинен мати наступні функціональні можливості:

- Створення бренду та типу товару товару;
- Створення товару та його характеристик;
- Реалізація реєстрації, авторизації та перевірки на токен;
- Зберігання даних користувачів та товару в БД;
- Фільтрація даних за типом та брендом;
- Перегляд характеристик товару;
- Додавання в кошик;
- Придбання товару;
- Розділення інтерфейсу (Pagination);

Сайт повинен поділятися на серверну та користувацьку частину, також він має бути інтерфейс для панелі адміністратора.

Серверна частина взаємодіє з БД та за допомогою патерну проектування MVC (model view controller) передає данні з контролера в клієнтську частину (рис. 1.4).

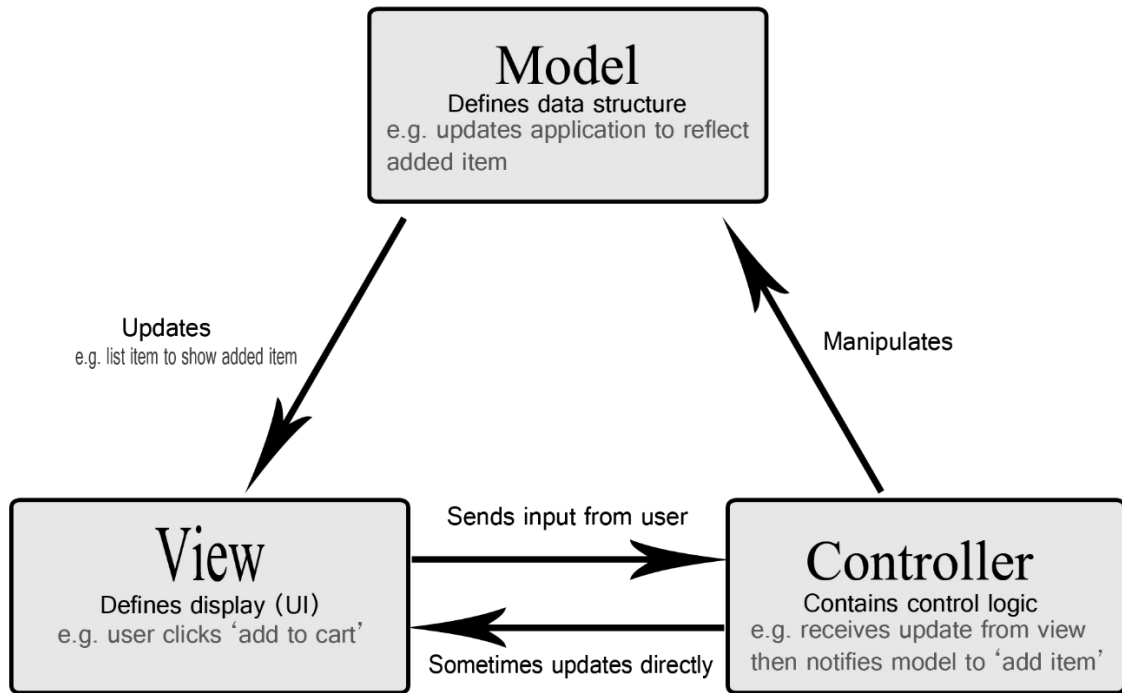


Рисунок 1.4 – Структура патерну MVC

Користувачський інтерфейс веб-сайту повинен бути зрозумілим, інтуїтивно представляти структуру розміщеної інформації та забезпечувати швидкий і логічний перехід між розділами та сторінками. Адміністративна частина сайту має надавати можливість створення та відправки даних про новостворені бренди, типи та товари.

Для досягнення мети кваліфікаційної роботи необхідно вирішити такі задачі:

- Дослідити процеси організації інтернет-торгівлі.
- Провести аналіз аналогічних сайтів для розробки технічного завдання.
- Вибрати та налаштувати інструменти реалізації.
- Визначити структуру даних та спроектувати веб-сайт.
- Реалізувати додатки та оптимізувати їх.

## **2 ВИБІР АРХІТЕКТУРИ СИСТЕМИ ТА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ**

### **2.1 Вибір мови програмування та менеджера пакетів**

У сучасному світі веб-технологій створення та підтримка веб-сторінок є важливим аспектом діяльності будь-якої компанії чи організації. Основою для розробки веб-сторінок є мова розмітки HTML (HyperText Markup Language), яка забезпечує структурування контенту та взаємодію з користувачами. HTML виступає невід'ємною частиною веб-розробки, дозволяючи створювати від простих статичних сторінок до складних динамічних веб-додатків. Його можливості забезпечують інтеграцію текстових, графічних та мультимедійних елементів, а також форм для взаємодії з користувачами, що робить його універсальним інструментом для розробників.

HTML є основною мовою розмітки, яка використовується для створення веб-сторінок та веб-додатків. Його основна мета полягає у структуризації контенту на веб-сторінках за допомогою тегів, які вказують браузерам, як відображати текст, зображення та інші елементи. Від моменту свого створення HTML став невід'ємною частиною інтернету, забезпечуючи основу для веб-контенту.

У сучасному світі веб-технологій, крім створення та підтримки веб-сторінок, важливим аспектом є їхній візуальний вигляд та привабливість для користувачів. Основою для оформлення та стилізації веб-сторінок є мова каскадних таблиць стилів CSS (Cascading Style Sheets), яка забезпечує візуальне представлення контенту та взаємодію з користувачами. CSS виступає невід'ємною частиною веб-розробки, дозволяючи створювати від простих статичних сторінок до складних динамічних веб-додатків з вражаючим дизайном. Його можливості забезпечують інтеграцію різноманітних стилів, графічних елементів, анімацій та адаптивного дизайну, що робить його універсальним інструментом для розробників.



CSS є основною мовою стилів, яка використовується для оформлення веб-сторінок та веб-додатків. Його основна мета полягає у визначенні стилів для HTML-елементів, що дозволяє контролювати кольори, шрифти, макети, відступи та інші аспекти вигляду веб-сторінок. Від моменту свого створення CSS став невід'ємною частиною інтернету, забезпечуючи можливість створення красивих та адаптивних веб-інтерфейсів, які покращують користувацький досвід.

Наразі перед web-розробником стоїть безліч різноманітних завдань – від створення інтерактивних розважальних сайтів до серйозних бізнес-проектів, які вимагають підвищеної надійності та захисту від несанкціонованого доступу. Для їх реалізації необхідно правильно підбирати інструменти – мови програмування, фреймворки та бібліотеки. Сьогодні доступно багато мов програмування, і перевага кожної з них проявляється лише в контексті конкретного завдання. Вибір мови залежить від знань програміста та їхньої достатності для реалізації проекту.

Мова програмування – формальна система знаків, призначена для розробки комп'ютерних програм. Мова програмування визначає набір команд, синтаксису та семантичних правил, що визначають зовнішній вигляд програми та дії, які виконує виконавець (комп'ютер) під її контролем [7].

На наступному графіку показано популярність мов програмування для розробки веб сайтів (рис. 2.1).

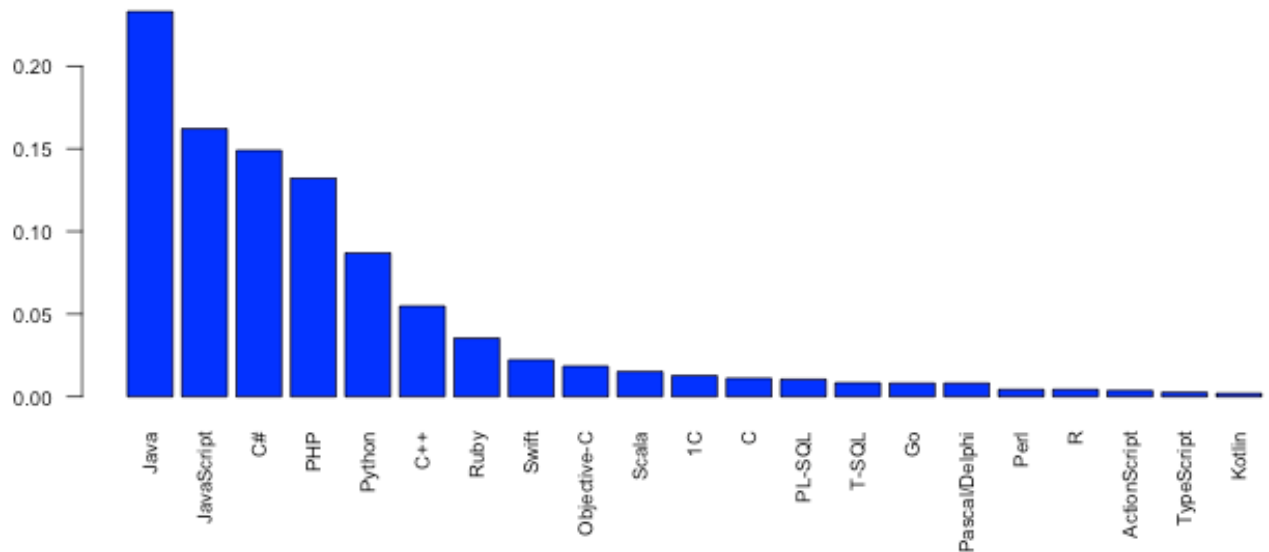


Рисунок 2.1 – Популярність мов програмування серед веб розробників  
(за матеріалами порталу DOU за 2023 рік)

Всі мови веб-програмування можна розділити на клієнтські та серверні мови. Як визначається з назви, мова клієнта використовується для написання клієнтських програм (веб-браузерів), а серверна мова для програм на стороні сервера.

До клієнтської мови програмування відноситься найпопулярніша мова програмування – JavaScript. JavaScript – це мова програмування для управління сценаріями перегляду веб-сторінок. При використанні цієї мови елементи середовища відображення мають бути змінені під час перегляду веб-сайту, тобто веб-сторінку не буде перезавантажено. Наприклад, зміна кольору фону веб-сторінки, підміна зображення, вкладеного у веб-сторінку, створення нового вікна відображення чи виведення динамічної інформації. Ця мова програмування активно розвивається та сьогодні є мультизадачною завдяки своїй популярності та простоті використання. JavaScript також широко використовується для розробки мобільних додатків за допомогою фреймворків, таких як React Native та Ionic. Це дозволяє розробникам використовувати ті ж знання і навички, які вони мають у веб-розробці, для створення мобільних додатків для платформ Android та iOS [7].

Крім того, JavaScript також використовується для розробки ігор, десктопних додатків, серверних додатків та багато іншого. Його універсальність і гнучкість робить його однією з найбільш важливих мов програмування сучасності. Саме тому для вибору основної мови для реалізації була JS [4].

Для забезпечення комфортного контролю пакетів та бібліотек набули популярності менеджери пакетів, які призначені для виконання цієї задачі. Сьогодні існують три популярні менеджери пакетів:

- Npm;
- Yarn;
- Pnpm;

NPM є централізованим сховищем для пакетів JavaScript, що використовується для управління залежностями в Node.js-проектах. Він надає можливість легко встановлювати, оновлювати та видаляти пакети за допомогою команд «npm install», «npm update» та «npm uninstall». За допомогою файлу package.json, NPM дозволяє керувати версіями пакетів та забезпечує консистентність середовища розробки [2].

Yarn, альтернатива NPM, пропонує широкий набір можливостей для управління залежностями. Він забезпечує легкість встановлення, оновлення та видалення пакетів, а також можливість керувати версіями пакетів та автоматизувати завдання розробки. Однією з переваг Yarn є його швидкодія, яка дозволяє швидше завантажувати пакети через ефективне кешування та паралельне завантаження.

pnpm відрізняється від NPM та Yarn у використанні символічних посилань для спільного використання однакових версій пакетів між різними проектами. Це дозволяє зменшити обсяг даних, необхідних для завантаження та збереження на диску, та прискорює час розгортання проектів.

При розробці дипломного проекту було обрано саме NPM, через його простоту, зручність та підтримку багатьох пакетів.

## 2.2 Вибір засобів для серверної розробки

Фреймворк – це динамічна бібліотека мови програмування, яка містить базові модулі цієї мови. Фреймворки створюються для спрощення процесу розробки, дозволяючи використовувати готові шаблони замість написання модулів з нуля..

Фреймворки для Node.js також є важливою складовою розробки програмного забезпечення, прискорюючи процес розробки та забезпечуючи безпеку та масштабованість додатків.

Переваги використання Node.js фреймворків включають в себе:

- **Продуктивність:** Фреймворки допомагають покращити продуктивність розробників, пропонуючи готові рішення для рутинних завдань.

- **Масштабованість:** Вони дозволяють легко масштабувати додатки для обробки більшого обсягу даних або запитів.

- **Зручність:** Фреймворки надають різноманітні функції та інструменти, що полегшують розробку і підтримку додатків.

- **Простота:** Вони дозволяють швидко створювати додатки за допомогою готових модулів та шаблонів.

- **Безпека:** Багато фреймворків мають вбудовані засоби для захисту додатків від різних видів атак.

Найпопулярніші фреймворки для Node.js включають в себе:

1. **Express.js:** Це легкий та гнучкий фреймворк, який швидко стає стандартом у світі Node.js розробки. Він дозволяє швидко створювати серверні додатки та API.

2. Koa.js: Цей фреймворк базується на промісах і спрощує асинхронне програмування. Він надає більш сучасний підхід до розробки, порівняно з Express.js.

3. NestJS: Це модульний фреймворк, який використовує TypeScript для розробки масштабованих та ефективних серверних додатків.

Використання фреймворків для Node.js дозволяє прискорити розробку додатків, покращити їх безпеку та масштабованість, тому вони є популярним вибором для розробників у сучасному програмуванні.

### **2.3 Вибір технологій для клієнтської частини**

Сьогодні фреймворки для візуальної розробки сайту мають не лише великий попит на ринку праці а й комфортні, для розробника, ресурси на сценарії, які полегшують процес розробки. Можна виділити три основні:

- React.js;
- Vue.js;
- Angular.js;

React – це бібліотека JavaScript для розробки інтерфейсів користувача. Він створений компанією Facebook і використовується для створення односторінкових додатків інтерфейсів (SPA) (рис. 2.2). Основна концепція React – це компоненти, які дозволяють розділити інтерфейс на невеликі, незалежні частини. React використовує віртуальний DOM для оптимізації роботи з реальним DOM і забезпечення високої продуктивності додатка [1].

Vue.js – це прогресивний фреймворк JavaScript для створення інтерфейсів користувача та односторінкових додатків. Він заснований на ідеї реактивного програмування та легко вивчається і використовується. Одним з ключових переваг Vue є його простота, гнучкість і швидкість. Vue має велику активну спільноту розробників і багато готових компонентів для швидкої розробки.

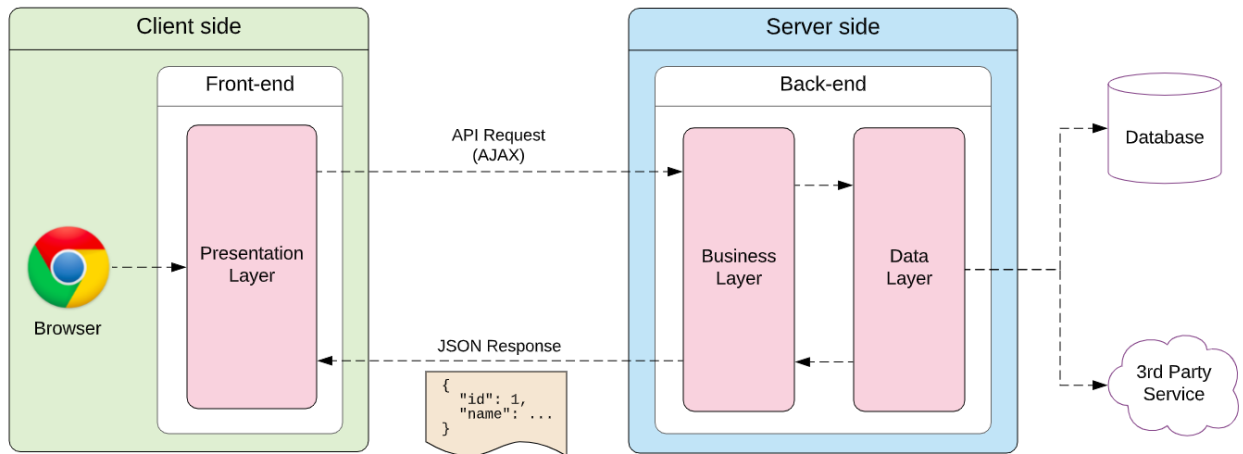


Рисунок 2.2 – Архітектура SPA додатку

Angular – це фреймворк JavaScript, розроблений і підтримуваний Google, для створення веб-додатків з багаторівневою архітектурою. Він пропонує повний стек інструментів для розробки, включаючи роутинг, HTTP-запити, валідацію форм і багато іншого. Angular використовує TypeScript, що дозволяє відразу отримати переваги сучасних функцій JavaScript, таких як статична типізація і покращений інтелектуальний код.

Для реалізації був обраний саме React.js, обумовлено це тим, що він має велике та активне суспільство розробників та підтримку багатьох пакетів та бібліотек, які додають багато функціоналу [1].

Серед додаткових пакетів та бібліотек було використано:

- React Router DOM: Використовується для навігації між різними сторінками та компонентами у веб-додатку.
- Bootstrap та React-Bootstrap: Ці бібліотеки використовуються для швидкого та зручного створення стильних та адаптивних веб-інтерфейсів.
- Material-UI: Ця бібліотека має набір готових компонентів та стилів, що дозволяє швидко розробляти сучасні та естетично приємні інтерфейси.
- Axios: Використовується для здійснення HTTP-запитів до сервера з веб-додатка.

- Formik та Yup: Ці бібліотеки допомагають валідувати та обробляти форми в веб-додатку, що підвищує його надійність та користувацький досвід.
- MobX та MobX-React-Lite: Вони використовуються для керування станом додатка та забезпечення реактивності компонентів.
- JWT-decode: Використовується для декодування токенів JSON Web Tokens, що забезпечує безпеку та автентифікацію користувачів.

## **2.4 Вибір архітектури мережевого WEB-ресурсу**

На основі виконаного огляду та аналізу систем-аналогів, було виявлено, що для успішного діяння мережевого WEB-ресурсу інтернет магазину, потрібна здійснити програмну розробку розвиненої інформаційної WEB-системи, яка реалізує зберігання та обробку даних, залучаючи сучасні способи та WEB-технології.

Мережевий WEB-ресурс повинен мати базу даних, для забезпечення зберігання, швидкого доступу і пошуку інформації. База даних об'єднує дані різного характеру та може організувати по певним правилам. На даний час WEB-технології вважаються найбільш перспективними для створення розподілених інформаційних систем з подальшою можливістю публікації у мережі Інтернет.

Такі інформаційні системи характеризуються можливістю доступу до баз даних на великих відстанях. Інтеграція WEB-технологій з технологіями СУБД має численні переваги, але також вимагає глибоких знань у цих галузях. Поєднання WEB-технологій і СУБД для організації доступу до даних має багато переваг, але вимагає не лише знання цих технологій, а й уміння аналізувати та обирати оптимальну архітектуру для таких інформаційних систем. Перехід до архітектури клієнт-сервер став вирішенням проблеми мережевого трафіку та інших труднощів, що виникають із збільшенням об'єму даних і кількості користувачів настільних СУБД.

При розробці програмної системи необхідно враховувати її архітектуру, яка займається не тільки структурою і поведінкою системи, але також правилами. Для того, щоб мати правильну і надійну архітектуру і забезпечити грамотне проектування та інтеграцію програмних систем, вичерпно потрібно дотримуватись. Якщо не дотримуватись цих стандартів, то архітектура може бути такою, що не зможе розвиватися й задовольняти зростаючі потреби користувачів.

Клієнт-серверна архітектура інформаційної системи характеризується наявністю двох взаємодіючих самостійних процесів — клієнта і сервера, які можуть виконуватися на різних комп'ютерах, обмінюючись даними через мережу. Сервер баз даних виконує комплекс дій з управління даними, забезпечуючи такі функції:

- виконання запитів користувачів на вибірку та модифікацію даних і метаданих;
- підтримка цілісності даних згідно з визначеними в базі даних правилами;
- забезпечення авторизованого доступу до даних на основі перевірки і привілеїв користувачів;
- протоколювання операцій і ведення журналу транзакцій.

Як робоче місце користувача може бути використаний звичайний персональний комп'ютер так і інший доступний пристрій, що дозволяє не відмовлятися від звичного робочого середовища. Іншими словами, в простому випадку клієнт-серверна інформаційна система складається з двох основних компонентів:

- сервера баз даних, який керує даними і виконує запити до клієнтських застосувань;
- клієнтських застосувань, що надають інтерфейс користувача і посилають запити до сервера.



Існують і складніші реалізації архітектури клієнт-сервер, наприклад багатоланкові інформаційні системи з використанням серверів додатків, що реалізують бізнес-логіку і що здійснюють обробку даних.

Дворівнева архітектура клієнт-сервер, використовується в клієнт-серверних системах, де сервер відповідає на клієнтські запити безпосередньо і в повному обсязі, при цьому використовуючи лише власні ресурси. Тобто сервер не викликає сторонні мережеві додатки і не звертається до сторонніх ресурсів для виконання якої-небудь частини запиту (рис. 2.3).

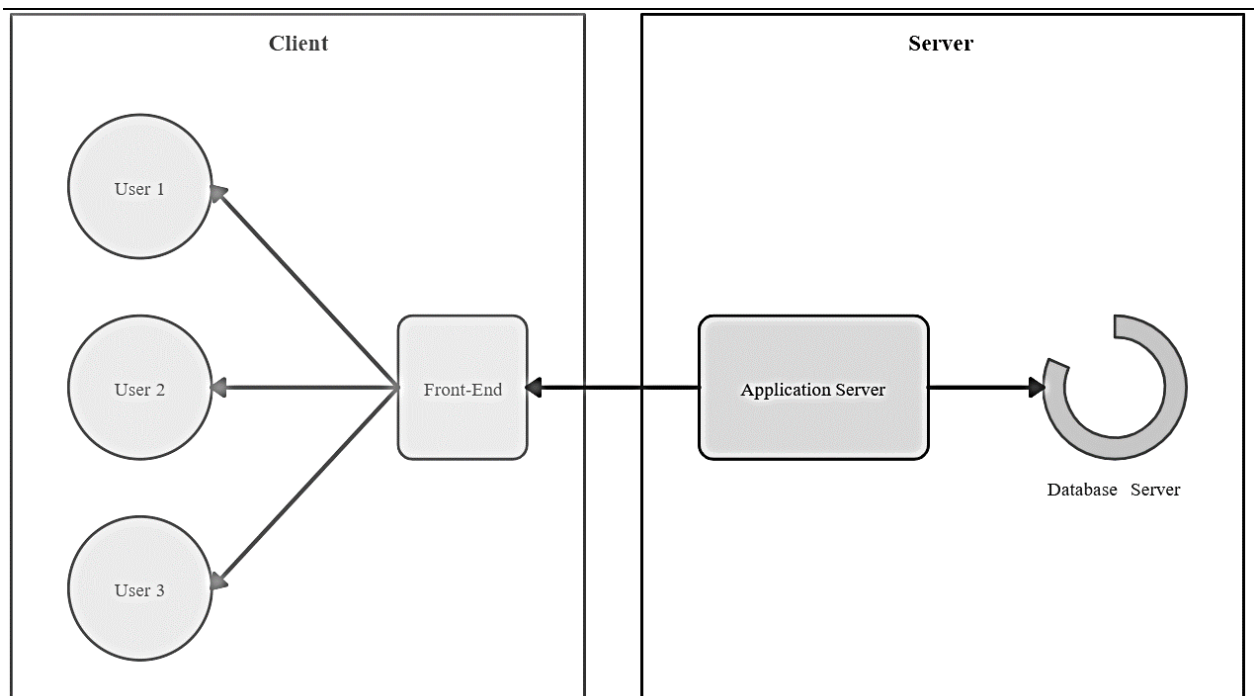


Рисунок 2.3 – Дворівнева архітектура «клієнт-сервер»

Необхідність масштабованості системи стала перешкодою для традиційної дворівневої архітектури клієнт-сервер. У результаті цього на стороні клієнта виникли проблеми, що заважають досягненню справжньої масштабованості додатків: для ефективної роботи "товстого" клієнта потрібні значні обчислювальні ресурси, включаючи дисковий простір, оперативну пам'ять і потужність центрального процесора. З метою вирішення цих проблем корпоративної масштабованості, було запропоновано новий варіант

традиційної дворівневої архітектури клієнт-сервер - трирівнева архітектура "клієнт-сервер", яка включає три рівні програмного забезпечення, кожен з яких може функціонувати на різних платформах:

- рівень інтерфейсу користувача, розташований на комп'ютері кінцевого користувача (Client);
- рівень бізнес-логіки й обробки даних, що розташовується на сервері і часто називається сервером додатків (Application server);
- СУБД, яка зберігає дані, необхідні для функціонування проміжного рівня, виконуючись на окремому сервері бази даних (Data server).

Клієнт відповідає тільки за інтерфейс користувача і, можливо, виконує деяку дуже просту логічну обробку даних, наприклад перевірку коректності введення даних (рис. 2.4).

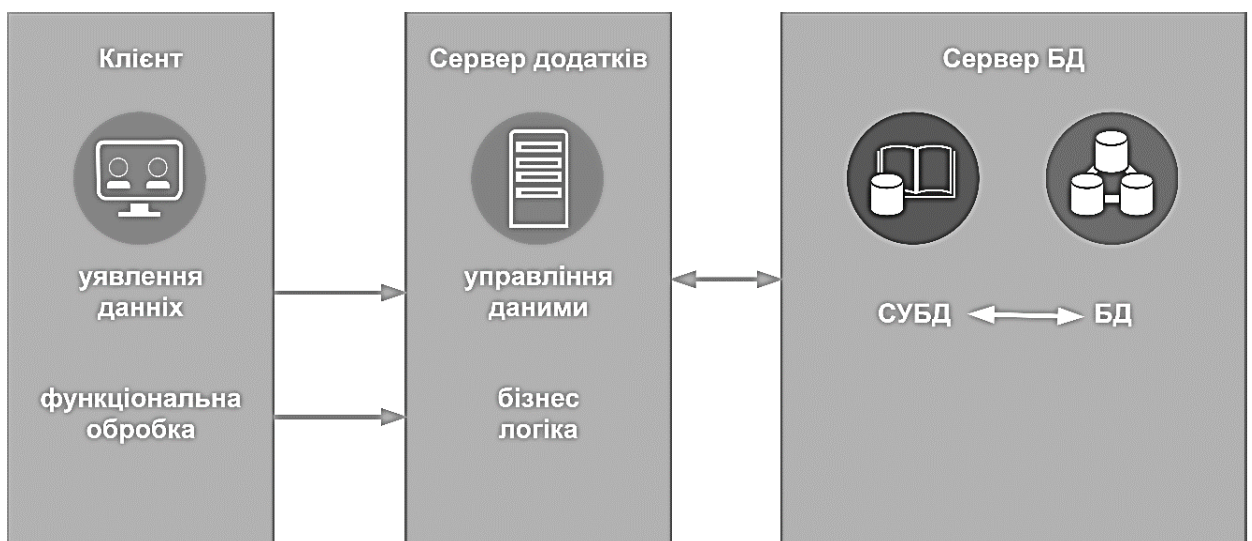


Рисунок 2.4 – Триврівнева архітектура «клієнт-сервер»

Клієнт, обмежений подібним функціональним набором, отримав назву "тонкого" клієнта. Основна бізнес-логіка додатка тепер розміщується на окремому виділеному рівні, який фізично пов'язаний з клієнтом і сервером бази даних через LAN або WAN мережу. Передбачається, що один сервер додатків може обслуговувати велику кількість клієнтів.

Трирівнева архітектура клієнт-сервер має багато переваг, деякі з них:

- для «тонкого» клієнта потрібно менше дорогого апаратного забезпечення;
- централізація бізнес-логіки для багатьох кінцевих користувачів на одному сервері програми.

Переваги використання трирівневої архітектури включають наступне: завдяки перерахованим механізмам відпадає потреба в розгортанні програмного забезпечення на великій кількості комп'ютерів. Додаткова модульність спрощує модифікацію або заміну програмного забезпечення на кожному рівні без впливу на інші рівні. Відокремлення основної бізнес-логіки програми від функцій бази даних полегшує завдання рівномірного розподілу навантаження. Ще однією перевагою є те, що трирівнева архітектура природно відображається в середовищі WWW, де браузер виконує роль "тонкого" клієнта, а WEB-сервер — роль сервера додатків (рис. 2.5).

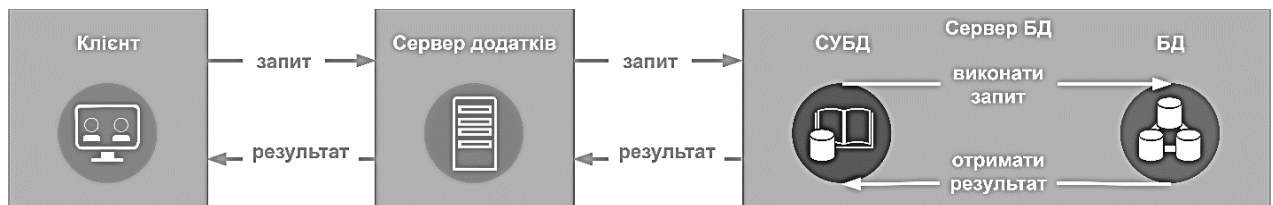


Рисунок 2.5 – Механізм взаємодії у трирівневій архітектурі «клієнт-сервер»

Трирівнева архітектура може бути розширена, щоб створити N-рівневу архітектуру. Наприклад, в проміжному рівні трирівневої архітектури можуть бути додані ще два рівні, один із яких відповідає за функції звичайного WEB-сервера, а інший - за функції типового сервера додатків.

Для реалізації інтернет-магазину була обрана трирівнева архітектура. Ця структура дозволяє реалізувати бізнес-логіку клієнтського застосування для взаємодії з базою даних системи.

## 2.5 Вибір системи управління базами даних

У сучасному суспільстві, яке переходить від індустріальної до постіндустріальної ери, автоматизовані бази даних стають надзвичайно важливими. Ці бази можуть бути величезними та містити різноманітну інформацію, необхідну для діяльності організацій. Банк даних - це інформаційна система, яка забезпечує централізоване зберігання, оновлення та надання даних користувачам за їх запитамі. Це комплекс апаратного та програмного забезпечення, а також персонал, який забезпечує його роботу. Банк даних може містити одну або декілька баз даних, систему управління базами даних і персонал, який відповідає за його функціонування. База даних - це організовані дані, збережені в пам'яті комп'ютерів. На наступному графіку показано популярність СУБД для розробки веб сайтів (рис. 2.6).

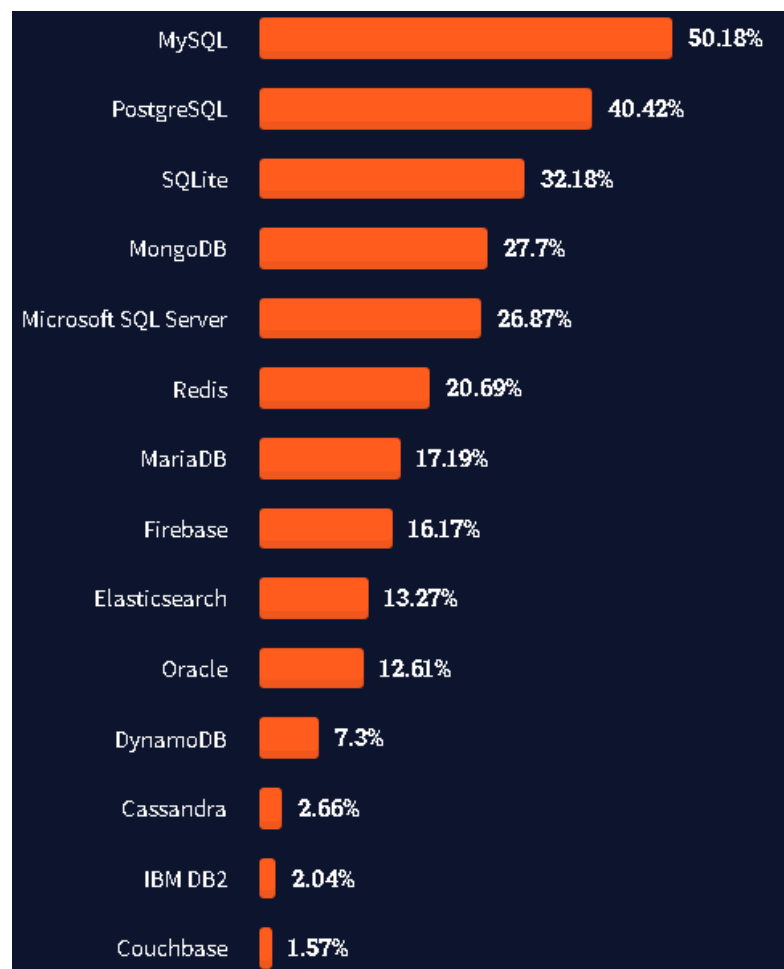


Рисунок 2.6 – Рейтинг популярності СУБД

SQL – це стандартизована мова виконання запитів, яка використовується і MySQL та PostgreSQL. Цей стандарт має кілька версій та був розроблений ще у 1986 році.

MySQL має підтримку не всіх функцій та можливостей SQL. Це зроблено для того, щоб працювати з MySQL було просто та зручно. Але якщо для проекту потрібне якесь розширення, розробники його можуть додати не на шкоду стандарту.

MySQL під час виконання запиту робить завантаження всієї відповіді сервера на згадку клієнта. Що стосується великих об'ємів це завжди зручно. За функціями PostgreSQL ширший, ніж MySQL. Наприклад, у PostgreSQL за допомогою курсору можна переміщувати отримані дані. Вам надається лише вказівник, а вся відповідь зберігається у пам'яті сервера баз даних. Цей покажчик можна зберігати між сеансами. PostgreSQL має підтримку регулярних виразів у запитах, рекурсивних запитів та успадкування таблиць.

PostgreSQL підтримує все нові стандарти SQL, тому цей проект досить складний і не настільки популярний як MySQL.

MySQL завжди був орієнтований на велику продуктивність, у той час як PostgreSQL був націлений на велику кількість налаштувань та стандартів. Але з часом ця ситуація змінилася і Postgre став більш продуктивним.

Для роботи з базою даних MySQL використовується таблиця InnoDB. А це означає, що MySQL буде значно швидше за Postgre у разі використання первинного ключа.

З приводу PostgreSQL, вся заголовна інформація таблиць розміщується в оперативній пам'яті. Можна використовувати декілька індексів до однієї таблиці для більшої зручності. Загалом PostgreSQL працює швидше, крім ситуацій із використанням первинних ключів.

MySQL та PostgreSQL мають схожий набір, який, звичайно ж, має свої відмінності. У Postgre типи більш різноманітні і є свої типи полів для певних видів даних, яких, наприклад, немає MySQL.

Згідно з обраною предметною областю та визначеними функціональним можливостям мережевого WEB-ресурсу, для інтернет-магазину підходить PostgreSQL.

СУБД PostgreSQL також має клієнт-серверну архітектуру, що робить її ефективним і надійним інструментом для зберігання та керування даними. SQL-сервер PostgreSQL підтримує багатопоточність і роботу на різних платформах, що робить його відмінним вибором для різноманітних застосунків. Зручні інструменти адміністрування PostgreSQL дозволяють легко керувати базами даних і виконувати адміністративні завдання. PostgreSQL також має широкий діапазон програмних інтерфейсів застосувань, що дозволяє легко інтегрувати його з різними програмами і платформами розробки. Подібно до інших реляційних баз даних, PostgreSQL забезпечує можливість створення реляційних баз даних, де дані зберігаються у вигляді окремих таблиць. Це дозволяє досягти високої продуктивності та гнучкості у роботі з даними, а також спрощує розробку та підтримку додатків.

Багатофункціональний SQL-сервер PostgreSQL, з відкритим вихідним кодом, робить його привабливим вибором для багатьох розробників і організацій, що шукають надійний та ефективний засіб для роботи з даними.

Вибір СУБД PostgreSQL зумовлений багатьма факторами, одним з яких є її відкритий код. Це дає користувачам можливість вільно використовувати, модифікувати та розповсюджувати програмне забезпечення відповідно до їхніх потреб. Наявність ліцензії PostgreSQL регламентує, що дозволено, а що ні у відношенні до комерційного використання. PostgreSQL також відома своєю надійністю та швидкодією. Вона забезпечує ефективне зберігання, пошук, сортування та отримання даних. Сервер PostgreSQL керує доступом до даних і може обслуговувати декілька користувачів одночасно. База даних PostgreSQL, як і MySQL, підтримує мову структурованих запитів SQL, що дозволяє виконувати складні запити до бази даних. Також PostgreSQL підтримує безліч користувачів, що працюють з базою даних одночасно, і забезпечує швидке виконання запитів.

Гнучкість СУБД PostgreSQL забезпечується підтримкою різних типів таблиць, що відповідають різним вимогам користувачів. Наприклад, користувачі можуть вибирати таблиці, що підтримують повнотекстовий пошук або транзакції на рівні окремих записів, в залежності від їхніх потреб. У PostgreSQL використовуються різні типи таблиць, такі як таблиці, що підтримують транзакції та зовнішні ключі, які є важливими для забезпечення цілісності та консистентності даних. Хоча ці можливості можуть призвести до деякої переваги в швидкості, PostgreSQL прагне забезпечити рівновагу між гнучкістю та продуктивністю, щоб задовольнити потреби користувачів. Підтримка транзакцій та зовнішніх ключів в PostgreSQL реалізована для різних типів таблиць, що дозволяє користувачам використовувати їх у своїх проєктах з врахуванням їхніх вимог до консистентності та безпеки даних. Ці можливості створюють базу для розробки надійних та безпечних веб-додатків та мережевих ресурсів, що відповідають сучасним вимогам ринку.

Можливість сервера СУБД PostgreSQL дозволяє одночасно підключатися необмеженій кількості користувачів, що робить її популярним вибором для веб-додатків та мережевих ресурсів з великою кількістю користувачів. Доступ до сервера PostgreSQL можна здійснити в інтерактивному режимі через різні інтерфейси, які дозволяють вводити SQL-запити та переглядати результати. Крім того, PostgreSQL забезпечує розвинену систему безпеки, яка захищає дані від несанкціонованого доступу та забезпечує їхню цілісність. Вона працює під управлінням різних операційних систем, таких як UNIX, Windows і OS/2, що дозволяє використовувати PostgreSQL у різноманітних середовищах. Використання PostgreSQL, як і MySQL, забезпечує доступ до розширених можливостей масштабування і адміністрування. Ця СУБД підтримує API для різних мов програмування, таких як Perl, C, PHP, JS і інші, що дозволяє розробникам створювати різноманітні клієнтські програми для роботи з базою даних PostgreSQL.

## 2.6 Вибір системи управління версіями та середовище розробки

Для зручності зберігання та взаємодії з проектом активно використовувалося таке програмне забезпечення, як GitHub.

GitHub – один з найбільших вебсервісів для спільної розробки програмного забезпечення. Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub, Inc (раніше Logical Awesome) (рис. 2.6).

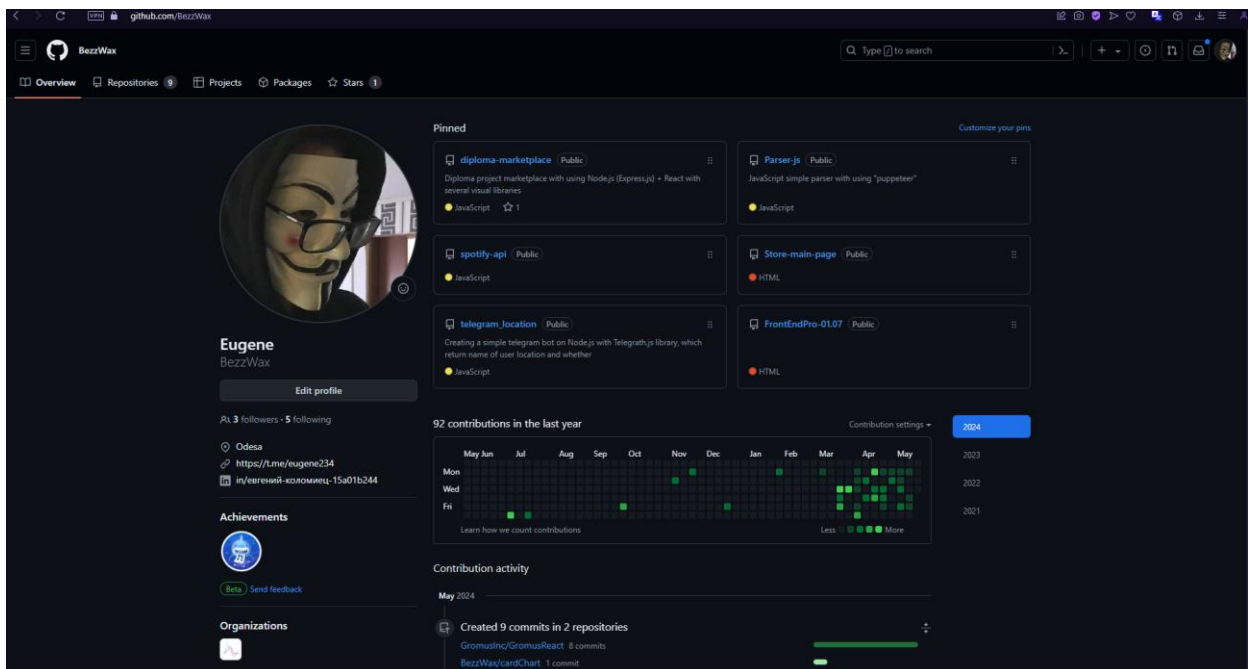


Рисунок 2.6 – Інтерфейс GitHub

Сервіс безкоштовний для проектів з відкритим вихідним кодом, з наданням користувачам усіх своїх можливостей (включаючи SSL), а для окремих індивідуальних проектів пропонуються різні платні тарифні плани.

У середовищі Windows система Git часто пакується у вигляді частини високорівневої програми з графічним інтерфейсом. Графічні інтерфейси для Git можуть абстрагувати та приховувати базові компоненти системи контролю



версій, що є основою. Це чудова підмога для новачків у Git, щоб вони могли швидко зробити свій внесок у проект. Але коли вимоги підвищуються та передбачається робота з рештою членів команди, необхідно розуміти принципи роботи вихідних методів Git. Інтерфейс терміналу Git пропонується у додатку Git Bash.

Git Bash – це програма для середовищ Microsoft Windows, що емулює роботу командного рядка Git (рис. 2.7). Bash – аббревіатура від Bourne Again Shell. Оболонка (Shell) є додатком терміналу для взаємодії з операційною системою за допомогою письмових команд. Bash – популярна оболонка, що використовується за замовчуванням у Linux та macOS. Git Bash є пакетом, який встановлює в операційну систему Windows оболонку Bash, деякі поширені утиліти Bash і систему Git.

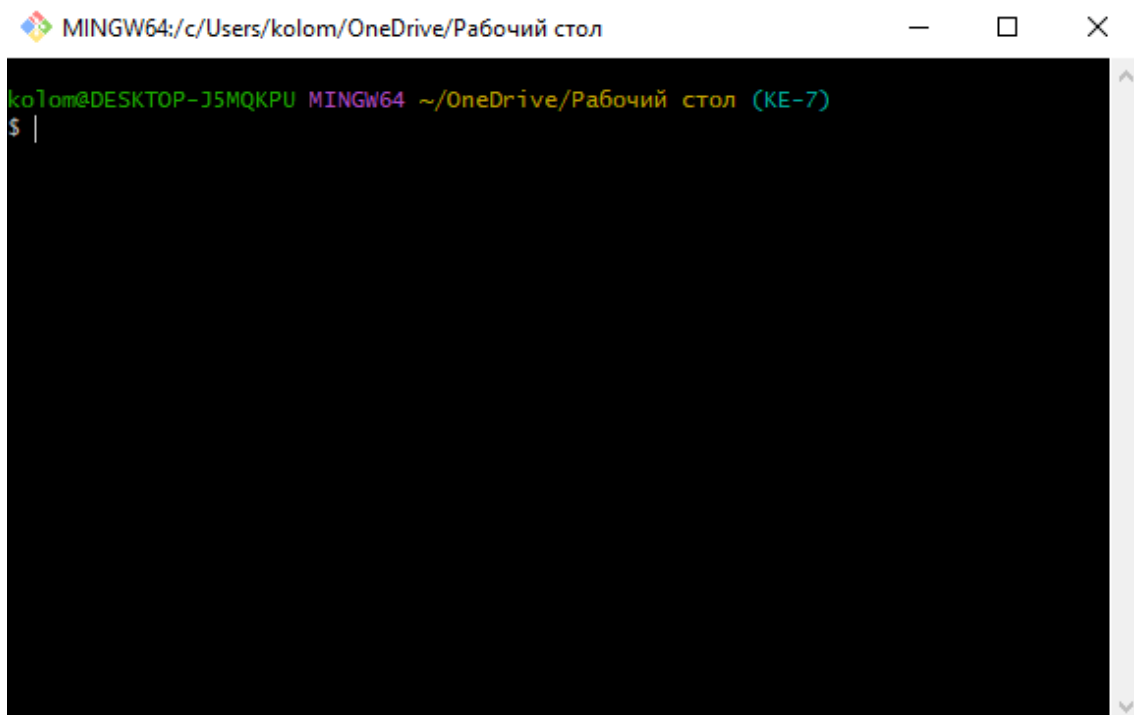


Рисунок 2.7 – Інтерфейс Git Bash

Bash-команда `pwd` використовується для виведення шляху до поточного робочого каталогу. Команда `pwd` еквівалентна виконанню команди `cd` у

терміналі DOS (або консолі Windows). Це папка або шлях до поточного сеансу Bash. Bash команда `ls` використовується для виведення списку вмісту поточного робочого каталогу. Команда `ls` еквівалентна команді `DIR` у консолі Windows. І оболонка Bash і консоль Windows підтримують команду `cd`. Команда `cd` є акронімом від «Change Directory» (змінити каталог). Команда `cd` викликається з додаванням як аргумент імені каталогу. При виконанні команди `cd` відбувається зміна поточного робочого каталогу сеансу терміналу каталог, переданий у вигляді аргументу.

Середовищем розробки було обрано Visual Studio Code. Visual Studio Code (VS Code) - це популярний безкоштовний редактор коду, розроблений компанією Microsoft, який використовується для написання та редагування коду в різних мовах програмування. Він забезпечує розробникам потужний набір інструментів для підвищення продуктивності.

Серед основних можливостей VS Code слід зазначити підтримку різних мов програмування. Він працює з великою кількістю мов "з коробки", таких як JavaScript, Python, C++, Java, PHP, та інші. Крім того, за допомогою розширень можна додати підтримку додаткових мов. Магазин розширень дозволяє користувачам розширювати функціональність редактора, додаючи інтеграції з системами контролю версій, такими як Git, або інструменти для розробки на конкретних мовах програмування.

VS Code пропонує інтелектуальні підказки (IntelliSense), що забезпечують автозаповнення коду, підказки та рекомендації, які значно прискорюють процес написання коду. Вбудований термінал дозволяє розробникам виконувати командний рядок безпосередньо в середовищі редактора, спрощуючи роботу з інструментами збірки, системами керування версіями тощо. Вбудовані інструменти для налагодження коду допомагають розробникам знаходити та виправляти помилки в коді, підтримуючи різні мови програмування. Інтеграція з Git дозволяє легко виконувати операції з керування версіями, такі як коміти, пул-реквести та відстеження змін. Функція "Живий перегляд" (Live Share) дозволяє розробникам працювати над одним

проектом в режимі реального часу, що особливо корисно для командної роботи та парного програмування.

Проте, VS Code має і свої недоліки. Хоча він відносно легкий у порівнянні з іншими середовищами розробки, він може споживати значну кількість пам'яті при роботі з великими проектами або при використанні численних розширень. Велика кількість параметрів налаштувань може бути складною для новачків. Хоча це дає можливість гнучко налаштовувати середовище під свої потреби, іноді це може заплутати користувачів. Багато можливостей VS Code реалізовані через розширення, тому для отримання повної функціональності потрібно знайти та встановити відповідні розширення, що може бути незручним. Іноді користувачі стикаються з проблемами продуктивності, особливо при роботі з дуже великими проектами або файлами.

В цілому, Visual Studio Code є потужним і гнучким інструментом для розробників, що поєднує в собі багатий функціонал і зручність використання, але вимагає деякого часу для налаштування та оптимізації під конкретні потреби.

### 3 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ

#### 3.1 Проектування інтернет-магазину за допомогою методології функціонального моделювання SADT

Під час проектування інтернет-магазину була використана методологія функціонального моделювання SADT (Structured Analysis and Design Technique) за стандартом IDEF0. Методологія SADT, яка є методологією аналізу та проектування систем, вона дозволяє відобразити важливі характеристики: Управління і ресурси потребують зворотного зв'язку. SADT відзначається тим, що ця методологія надає загальний опис функціонування системи. За допомогою графічної мови IDEF0, можливо зобразити інтернет-магазин, у графічному вигляді. На рисунку наведено контекстну діаграму інтернет-магазину рисунок. 3. 1.

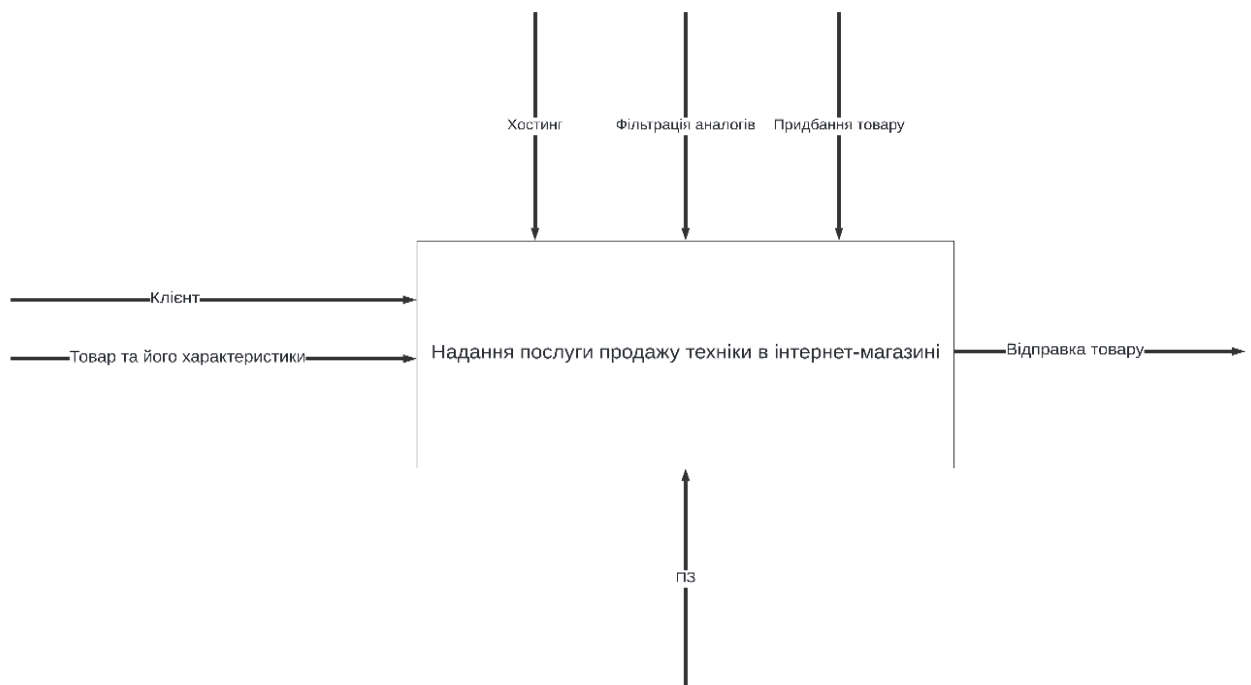


Рисунок 3.1 – Контекстна діаграма мережевого WEB-ресурсу

Методологія SADT є важливим інструментом для розробки інформаційних систем, оскільки вона надає розробникам набір методів, правил і процедур для аналізу та побудови функціональної моделі обраної предметної області. Одним з ключових компонентів SADT є графічна мова IDEF0, яка дозволяє описати бізнес-процеси у вигляді ієрархічної системи взаємопов'язаних функцій.

Контекстна діаграма є важливим етапом у розробці інформаційних систем, оскільки вона надає загальний опис інформаційної системи та її взаємодії з зовнішнім середовищем. Після створення головної роботи інтернет-магазину техніки, проводиться декомпозиція, яка полягає у розбитті системи на менші частини. Це дозволяє докладно описати кожен з них та їх функції, а також з'ясувати взаємозв'язок між ними.

Аналізуючи функції та взаємодію, можливо розбиття системи на більш дрібні фрагменти за допомогою функціональної декомпозиції. Після завершення кожного етапу декомпозиції проводиться перевірка створених діаграм фахівцями предметної області, щоб переконатись у відповіді. Для виправлення виявлених невідповідностей, необхідно забезпечити, щоб модель на кожному рівні декомпозиції точно відображала реальним процесам. Підхід, описаний вище, сприяє більшій систематизації та ефективності у процесу розробки інформаційних систем. Після кожного сеансу декомпозиції проводиться експертиза, під час якої експерти предметної області перевіряють відповідність реальних. Знайдені невідповідності виправляються. Це є досягнутим шляхом відповідності моделі реальним процесам на кожному рівню декомпозиції.

Синтаксис опису системи в цілому і кожного її фрагмента однаковий у всій моделі. Після декомпозиції контекстної діаграми інтернет-магазину, отримуємо три блоки – роботи (рис. 3.2).

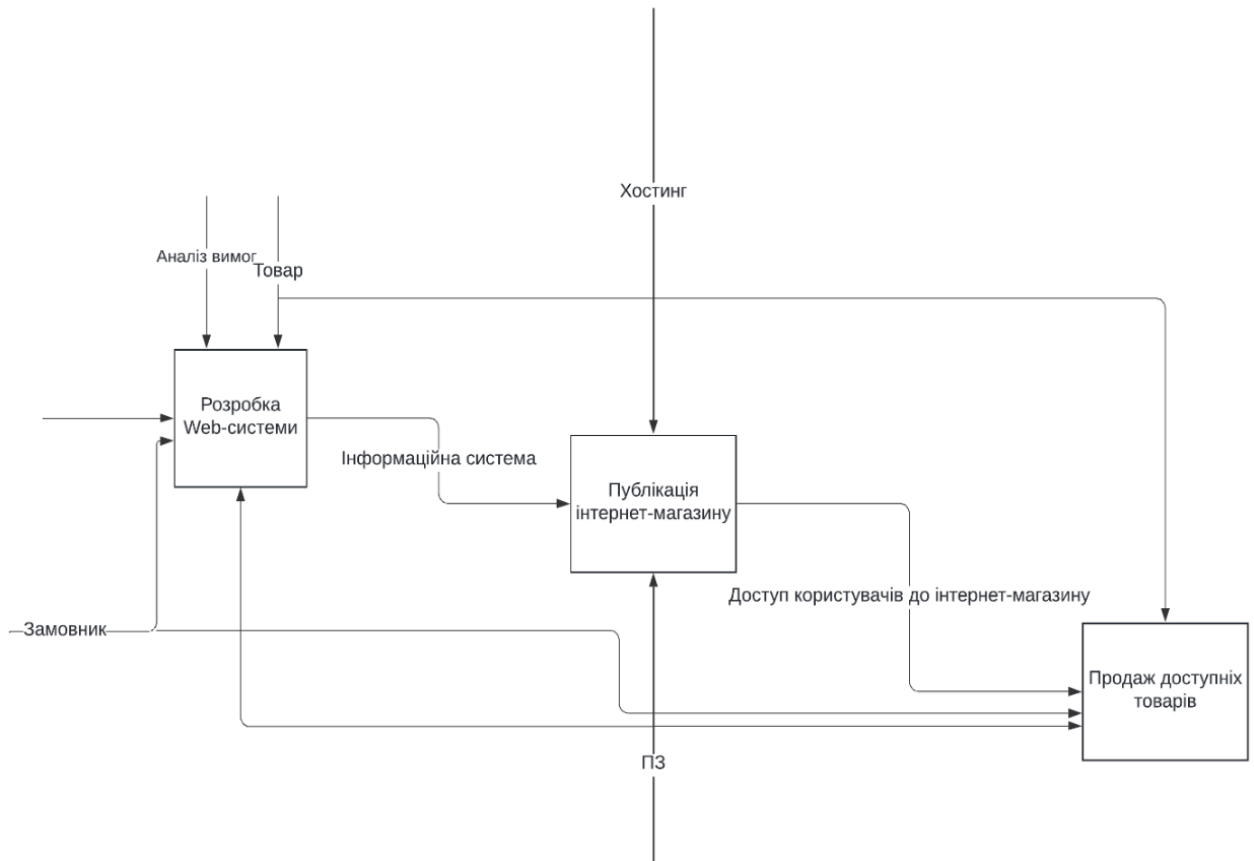


Рисунок 3.2 – Діаграма декомпозиції інтернет-магазину

Робота «Розробка WEB-системи» забезпечує виконання інформаційної системи на локальному комп'ютері: здійснюється розробка інтерфейсів, скриптів і баз даних, що забезпечують функціонування системи. Входом у неї внесення даних користувача, які потрібно занести у БД системи. Управляється за допомогою візуального інтерфейсу, запитом клієнта та правилами проектування системи. Механізмом є програмне забезпечення, яке потрібне для здійснення розробки, а результатом роботи є готова інформаційна система.

Наступна робота «публікування інтернет-магазину» забезпечує отримання доменного ім'я, а потім дозволяє здійснити публікацію WEB-системи на хостингу. Входом для роботи є готова інформаційна система для

розміщення. Управляється робота правилами оформлення оренди хостингу, а механізмом є – програмне забезпечення.

Робота «Продаж доступних товарів» призначена для того, щоб згідно з правилами додавання товару в кошик, клієнт вам можливість додавання необмеженої кількості товарів а потім редагування замовлення та покупки. У цієї роботи є два входи, це: дані про клієнта і доступна WEB-система, розміщена на хостингу. Управляється правилами оформлення замовлення і за умовами надання послуг Інтернет-провайдером. Механізмом є – програмне забезпечення, а виходом даної роботи є клієнт з товаром.

При здійсненні наступного етапу декомпозиції системи, отримаємо наступні блоки декомпозиції першого А1 блоку:

Робота «розробка інтерфейсу» має вхідні дані про характеристику товару, управління здійснюється за допомогою панелі адміністратора, запитом клієнта та правилами будівлі WEB-системи; механізм – програмне забезпечення, вихід розроблений прототип інтерфейсу інтернет-магазину. Робота «програмування системи» має вхідом прототип інтерфейсу макет – вихід попередньої роботи також замовник; управлінням є, як і в попередньому блоці, адміністративна панель; механізмом, як і у всіх роботах, є – програмне забезпечення; вихід – готова інформаційна система з необхідним функціоналом. Робота «створення БД» передбачає, що вхідом є функціонально готова система зберігати дані користувача та характеристики товару. Даний блок управляється правилами будівлі WEB-системи та правилами оформлення замовлення; робота виконується за допомогою програмного забезпечення; виходом є – функціонуюча, наповнена система.

Наступна робота, «тестування системи» –має три входи: Система заповнена замовником характеристиками товару, а програмне забезпечення виступає механізмом. Результатом даної роботи є готова інформаційна система, яка буде використана при наступному проекті. На рисунку представлена діаграма, яка показує процеси, що включені до блоку "розробка WEB-системи" (А1) рис. 3.3.

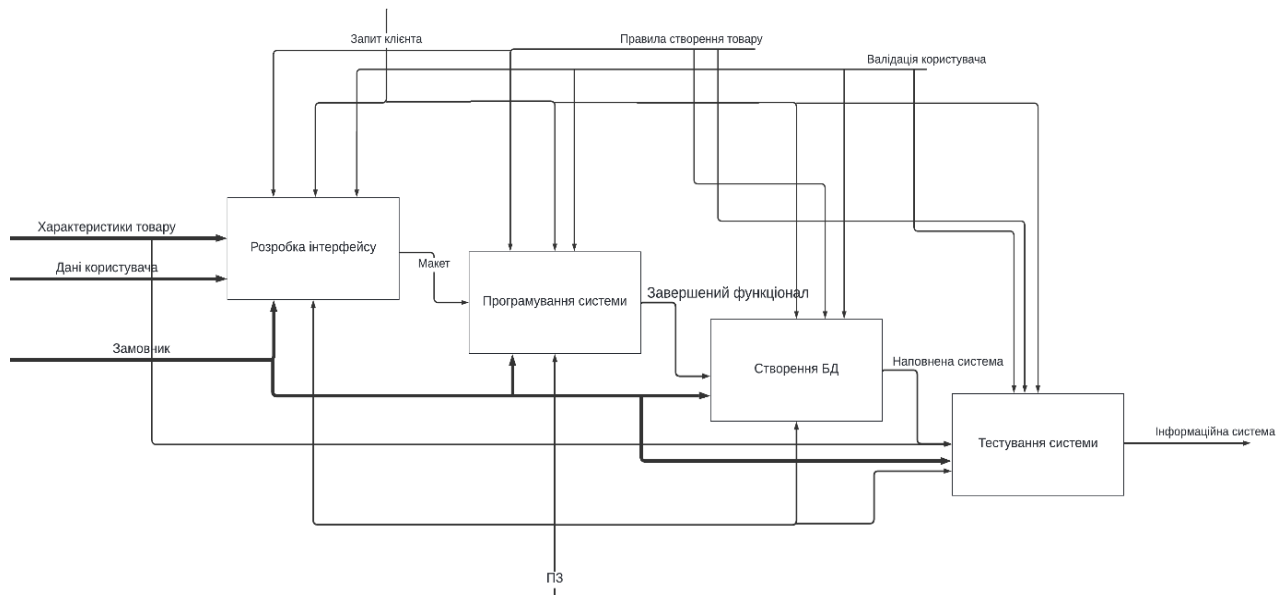


Рисунок 3.3 – Діаграми декомпозиції блоку «розробка WEB-системи»

При декомпозиції другого блоку А2 – «публікація на хостингу» – виділено наступні чотири завдання: «реєстрація домену», де вхідними даними є інформаційна система, а керування здійснюється хостингом; результатом є зареєстрований домен. Наступне завдання – «реєстрація хостингу», де вхідними даними є зареєстрований домен і інформаційна система під керуванням хостингу; результатом є зареєстрований хостинг. Останнє завдання – «прив'язка домену до сайту», де вхідними даними є зареєстрований хостинг і інформаційна система під керуванням хостингу; результатом є домен, закріплений за системою.

Робота «розміщення файлів на хостингу» передбачає на вході закріплений домен (домене ім'я,) та інформаційна система, управляється хостингом; виходом роботи є доступний у мережі Інтернет WEB-ресурс. Діаграма декомпозиції другого А2 блоку – «публікування на хостингу» наведена на рис. 3.4.



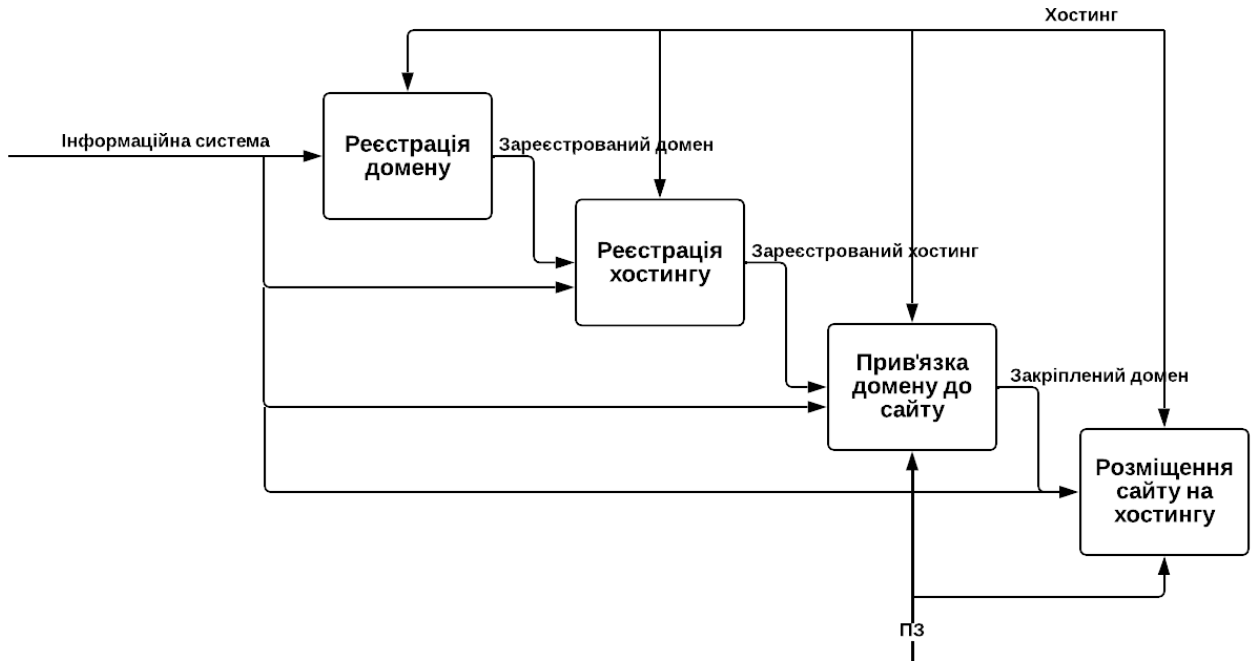


Рисунок 3.4 – Діаграми декомпозиції блоку «розміщення на хостингу»

В подальшому було проведено наступний етап здійснення декомпозиції інформаційної системи. При декомпозиції третього АЗ блоку – «продаж доступних товарів» виділені наступні три роботи.

Робота «реєстрація користувачів», щоб здійснити реєстрацію необхідні данні користувачів та доступний WEB-ресурс у мережі Інтернет; робота буде керуватися валідацією користувача; результатом роботи є зареєстрований у системі користувач. Робота «вибір товару» передбачає, що організовано каталог продукції та в системі є зареєстровані користувачі для вибору товару, управлінням є вибір товару, а результатом є додані до кошика товари. Робота «оформлення замовлення» передбачає, щоб організувати оформлення замовлення потрібно обрати товари; механізм даної роботи є програмне забезпечення, а результатом є замовник з оформленим замовленням. Діаграма декомпозиції третього АЗ блоку – «продаж доступних товарів» наведена на рис. 3.5.

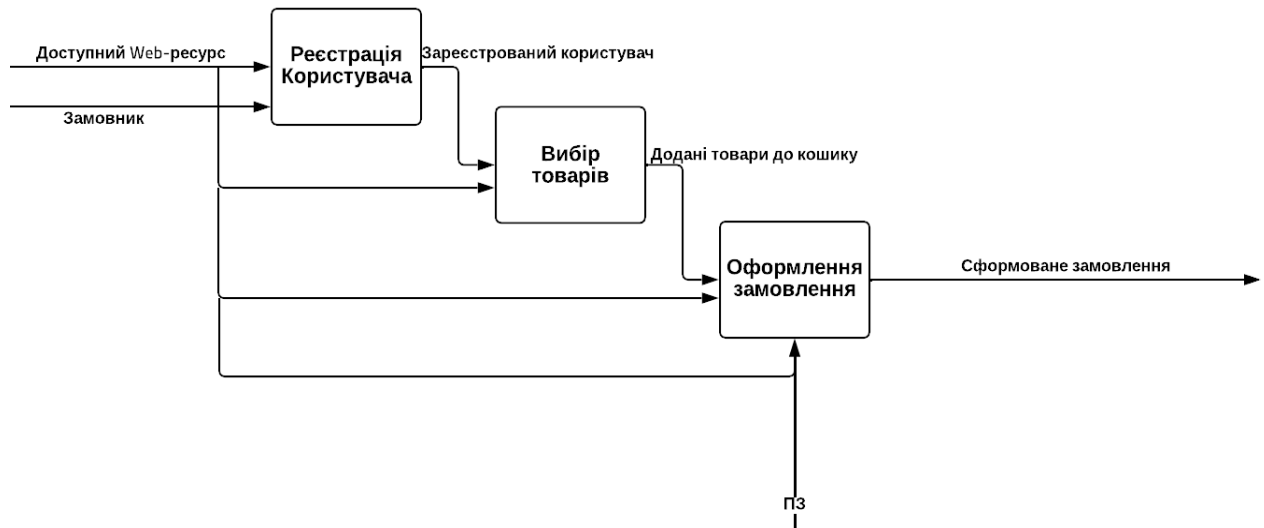


Рисунок 3.5 – Діаграми декомпозиції блоку «продаж доступних товарів»

### 3.2 Проектування мережевого WEB-ресурсу за допомогою методології Workflow Diagramming

Подальше проектування інтернет магазину було здійснено за допомогою методології послідовного виконання процесів Workflow Diagramming (стандарт IDEF3). Цей стандарт описує логіку виконання дій. IDEF3 може використовуватися самостійно і спільно з методологією IDEF0: будь-який функціональний блок IDEF0 може бути представлений у вигляді послідовності процесів або операцій засобами IDEF3. Якщо IDEF0 описує, що здійснюється в системі, то стандарт IDEF3 описує, як це робиться. Згідно з методологією IDEF3 інтернет-магазину має єдину роботу – «продаж доступних товарів», яку можливо навести у контекстній діаграмі системи. Провівши декомпозицію контекстної діаграми, спостерігається наступна послідовність виконання робіт (рис. 3.6).



Рисунок 3.6 – Діаграма декомпозиції інтернет-магазину

Першою роботою системи є «розробка WEB-системи», далі – робота «публікування на хостингу». Зв'язок між цими роботами означає, що робота-приймач може завершитись ще до закінчення роботи-джерела. Наступною є робота «продаж товарів», яка пов'язана з блоком «публікування на хостингу» старшим зв'язком, що передбачає завершення всіх попередніх робіт. При декомпозиції наступного рівня роботи «розробка WEB-системи» отримано чотири блоки – це роботи з двома перехрестями (рис. 3.7).

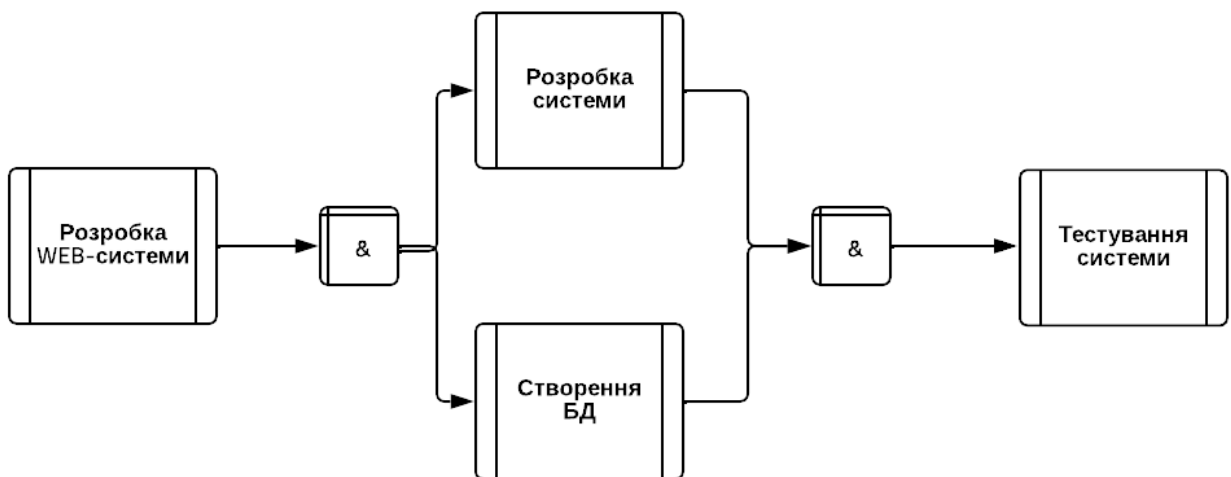


Рисунок 3.7 – Діаграма декомпозиції блоку «розробка WEB-системи»

Після завершення першої роботи «розробка інтерфейсу» йде перше перехрестя «асинхронне І», що передбачає, що подальші роботи можуть початися не одночасно, але обов'язково повинні бути запущені. Це такі роботи як: «розробка системи» та «створення бази даних». При злитті стрілок-виходів з цих робіт, використовується таке ж перехрестя «асинхронне І», що означає те, що роботи мають, завершитися, але це може бути не одночасно. Наступною є робота «тестування системи», яка і буде завершальною.

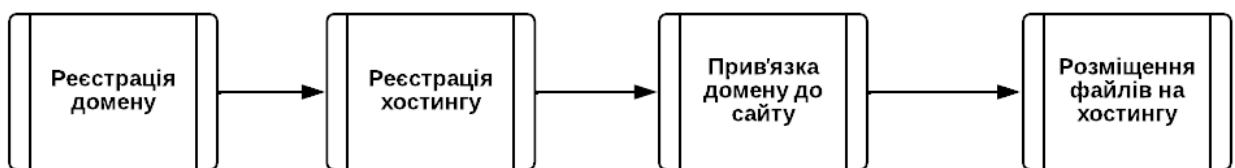


Рисунок 3.8 – Діаграма декомпозиції роботи «публікування на хостингу»

При декомпозиції роботи «публікування на хостингу» всі роботи пов'язані старшим зв'язком і розміщені послідовно: «реєстрація домену», «реєстрація хостингу», «прив'язка домену до сайту», «розміщення файлів на хостингу» (рис. 3.8).

При декомпозиції блоку «надання послуг» отримано три блоки робіт, які розміщені послідовно і пов'язані між собою старшим зв'язком (рис. 3.9).



Рисунок 3.9 – Діаграма декомпозиції роботи «продаж товарів»

### 3.3 Проектування мережевого WEB-ресурсу за допомогою методології потоків даних DFD

Модель аналізу інтернет-магазину визначається за допомогою ієрархії діаграм потоків даних (DFD), які описують процес перетворення інформації від її введення в систему до видачі користувачеві. DFD використовуються для зображення руху даних і обробки інформації як доповнення до методології функціонального моделювання IDEF0. На відміну від IDEF0, стрілки на DFD показують лише напрямок руху об'єктів, включаючи дані, між різними процесами. DFD - це граф, який демонструє перехід значень даних від їх джерел через оброблюючі процеси до приймачів у різних об'єктах. Контекстні діаграми верхнього рівня ієрархії відображають взаємозв'язок основного процесу системи з зовнішніми сутностями через відповідні входи та виходи. Деталізація контекстних діаграм відбувається за допомогою діаграм нижнього рівня. Ця декомпозиція триває до тих пір, поки не досягається рівень, на якому процеси стають елементарними, і додаткова деталізація стає неможливою.

У контекстній діаграмі головним процесом системи є «надання послуги у фотозйомці та пост обробці зображень». Зовнішніми сутностями, які впливають на систему, є: «клієнт», «сервер» і «куплений товар». Існує блок-сховище даних. Зв'язок між замовником і головною роботою полягає в «запиті» і «бюджеті». Із системи виходять дані в зовнішні сутності «куплений товар» – «оформлене замовлення», і в «сервер» – «інформаційна система». Зі сховища даних в систему передаються «данні про товар». Контекстна діаграма інтернет-магазину представлена на рис. 3.10.

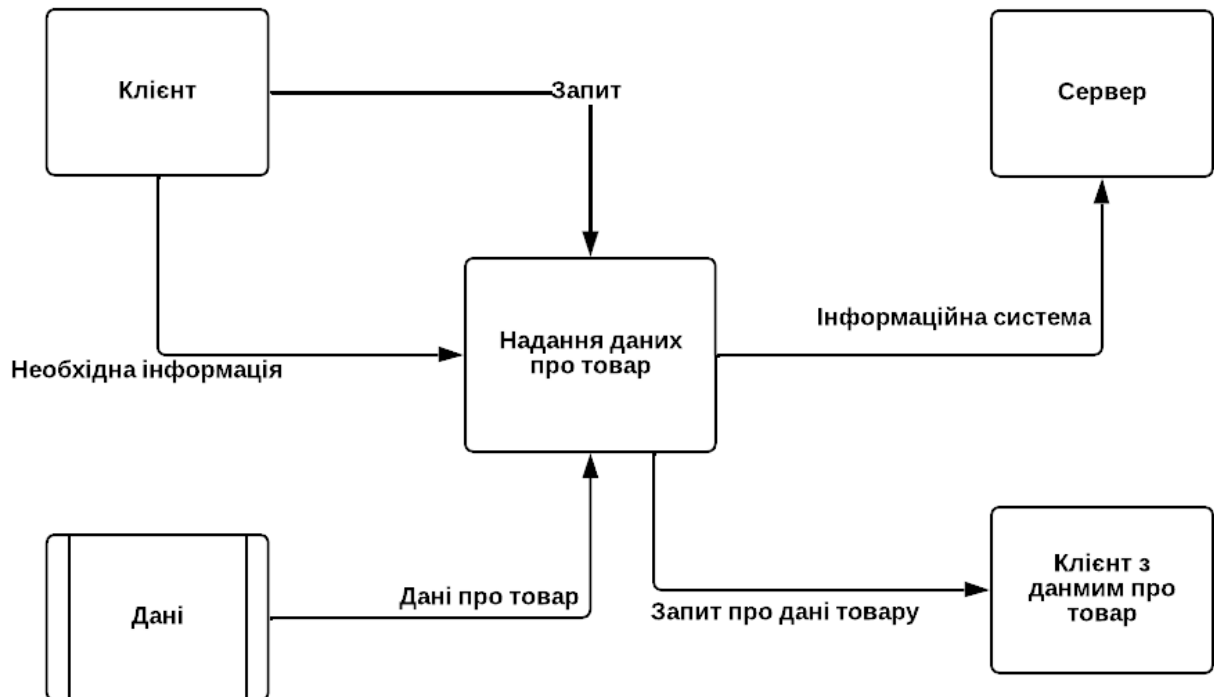


Рисунок 3.10 – Контекстна діаграма ІС для інтернет-магазину

Для головного процесу, представленого на контекстній діаграмі, проводиться декомпозиція. На першому рівні ієрархії показані основні внутрішні процеси системи та відповідні їм зовнішні сутності. Перший процес системи – «Розробка інтернет-магазину», приймає потік даних від зовнішньої сутності «користувач» у вигляді «бюджету» і «запиту», а також із сховища даних – «дані про товар». Вихідний потік даних з цього блоку – «інформаційна система».

Блок «хостинг» отримує вихідний потік даних з блоків «розробка WEB-системи» і «сервер» у вигляді «файлів». Його вихідний результат - «WEB-ресурс», який стає вхідним для зовнішньої сутності «купівля».

Наступний блок, «купівля», приймає вхідні дані від зовнішньої сутності «клієнт», а саме: «бюджет» та «запит». Він також отримує дані з блоку «хостинг», які представлені у вигляді «WEB-ресурсу», а також зовнішні файли від сервера. Крім того, він отримує інформацію про товари з блоку «дані».

Результатом роботи цього блоку є «доданий товар до кошика», що відображається у зовнішній сутності «куплений товар» на рис. 3.11.

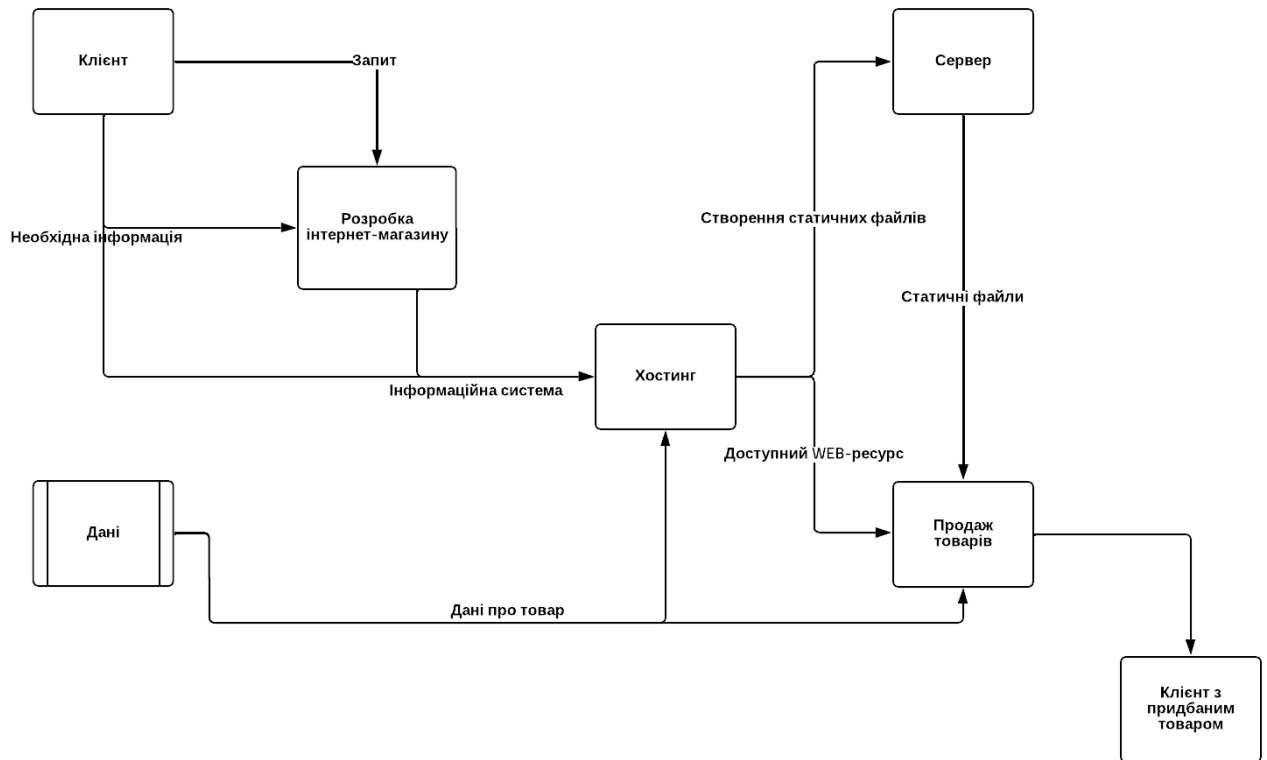


Рисунок 3.11 – Діаграма декомпозиції ІС для інтернет-магазину

Завдяки здійсненню проектування засобами методології потоків даних, в інформаційній системі інтернет-магазину проведено опис обробки інформації в системі, що забезпечує виконання продажу товару клієнту в системі, як додаток до методології функціонального моделювання IDEF0.

### 3.4 Проектування бази даних системи

Для створення та розробки інтернет-магазину обрана трирівнева архітектура "клієнт-сервер". У цій архітектурі база даних знаходиться на сервері локальної комп'ютерної мережі. На робочих станціях розміщена

клієнтська програма-браузер, за допомогою якої відбувається формування та відсилка запитів до сервера бази даних. Сервер бази даних, у свою чергу, обробляє запити та надсилає клієнту результати, які відображаються на html-формах, створених за допомогою запитів до бази даних.

Перший етап процесу проектування бази даних називається концептуальним проектуванням бази даних. Він включає створення концептуальної моделі даних для аналізованої частини системи, що базується на функціональних вимогах користувачів. Концептуальне проектування бази даних повністю незалежне від деталей її реалізації, таких як тип обраної СУБД, набір створюваних прикладних програм, використовувані мови програмування, тип обчислювальної платформи та інші особливості фізичної реалізації. На цьому етапі створюється концептуальне уявлення бази даних, яке включає визначення основних типів сутностей, їх зв'язків і атрибутів. Послідовність етапів проектування концептуальної моделі даних наступна: визначення сутностей, встановлення зв'язків між ними, опис атрибутів сутностей, призначення первинних та альтернативних ключів.

Сутність КЛІЄНТ містить наступні атрибути (табл. 1): id, email, password, role.

Таблиця 1 – Сутність «КЛІЄНТ»

№	Атрибут	Тип
1	id	int
2	email	varchar(255)
3	password	varchar(255)
4	role	string

Сутність ДЕВАЙС містить атрибути (табл. 2): id, name, price, rating, img, typeId, brandId,.



Таблиця 2 – Сутність «ДЕВАЙС»

№	Атрибут	Тип
1	id	int
2	name	varchar(32)
3	price	int
4	rating	Int(10)
5	img	varchar(255)
6	typeId	int
7	brandId	int

Сутність ДАНІ ПРО ДЕВАЙС містить атрибути (табл. 3): id, device\_id, title, description.

Таблиця 3 – Сутність «ДАНІ ПРО ДЕВАЙС»

№	Атрибут	Тип
1	id	int
2	device_id	int
3	title	varchar(255)
4	description	varchar(255)

Сутність ТИП ДЕВАЙСУ містить атрибути (табл. 4): id, name.

Таблиця 4 – Сутність «ТИП ДЕВАЙСУ»

№	Атрибут	Тип
1	id	int
2	name	varchar(255)

Сутність БРЕНД ДЕВАЙСУ містить атрибути (табл. 5): id, name.

Таблиця 5 – Сутність «БРЕНД ДЕВАЙСУ»

№	Атрибут	Тип
1	id	int
2	name	varchar(255)

Сутність КОШИК містить атрибути (табл. 6): id, name.

Таблиця 6 – Сутність «КОШИК»

№	Атрибут	Тип
1	id	int
2	name	varchar(255)

Сутність ДЕВАЙС В КОШИКУ містить наступні атрибути (табл. 7): id, device\_id, basket\_id.

Таблиця 7 – Сутність «ДЕВАЙС В КОШИКУ»

№	Атрибут	Тип
1	id	int
2	device_id	int
3	basket_id	int

Сутність РЕЙТИНГ містить наступні атрибути (табл. 8): id, user\_id, device\_id, rate.

Таблиця 8 – Сутність «РЕЙТИНГ»

№	Атрибут	Тип
1	Id	int
2	user_id	int
3	device_id	int
4	rate	int

Описавши, всі головні сутності й атрибути розроблюваної бази даних для інтернет-магазину, можна визначити зв'язки між сутностями. Уявімо базу даних у вигляді моделі «сутність-зв'язок». Тип використовуваного зв'язку «один-до-багатьох». Діаграма «сутність-зв'язок» представлена на рис. 3.12.

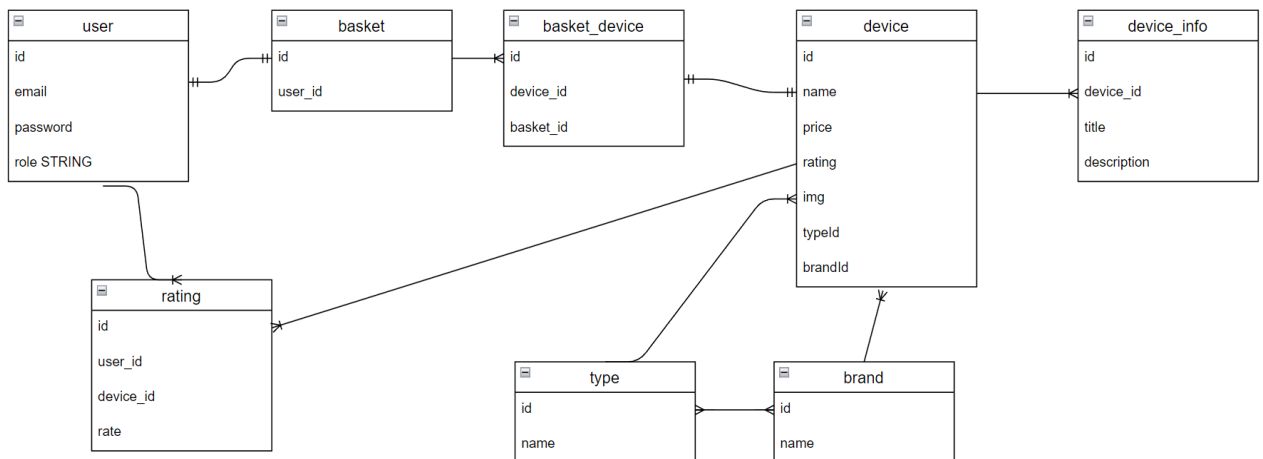


Рисунок 3.12 – Модель «сутність-зв'язок»

Після визначення структури бази даних, таких як таблиці, поля, індекси та зв'язки між таблицями, важливим кроком є нормалізація. Цей процес дозволяє розбити великі таблиці, які зазвичай містять надмірну інформацію, на менші логічні одиниці. Нормалізація включає усунення повторюваних груп даних (досягнення першої нормальної форми), видалення частково залежних атрибутів (досягнення другої нормальної форми) та видалення транзитивно залежних атрибутів (досягнення третьої нормальної форми).

Між сутностями «Клієнт» та «Корзина» існує зв'язок один-до-одного: один клієнт належить до одного кошика. Між сутностями «Кошик» та «Девайс в кошику» існує зв'язок один-до-багатьох: багато товарів належать до одного кошику. Між сутностями «Клієнт» та «Рейтинг» існує зв'язок один-до-багатьох: багато оцінок товару належать до одного клієнта. Між сутностями «Девайс в кошику» та «Девайс» існує зв'язок один-до-одного: один девайс

належить до одного девайсу в кошику. Між сутностями «Девайс» та «Дані про девайс» існує зв'язок один-до-одного: один девайс до одних даних про девайс. Між сутностями «Девайс» та «Бренд» існує зв'язок багато-до-одного: багато девайсів належать до одного бренду. Між сутностями «Девайс» та «Тип» існує зв'язок багато-до-одного: багато девайсів належать до одного типу. Між сутностями «Девайс» та «Рейтинг» існує зв'язок один-до-багатьох: один девайс належить до багатьох оцінок клієнтів.

У результаті проведеного, в ході дипломної роботи, проектування інтернет-магазину, що забезпечує засобами мережі Інтернет надання послуг у продажі техніки, визначена архітектура системи, проведено моделювання процесів в системі, розроблена база даних системи. Основним результатом проектування стало реалізація вимог, функціональних можливостей і бізнес-логіки інтернет-магазину.

## 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ

### 4.1 Керівництво додатком користувача-клієнта системи

Розроблено інтернет-магазин, який не лише забезпечує інформаційну присутність компанії в Інтернеті, але й надає інтерактивну можливість здійснення замовлень на техніку. Використання цієї системи спрямовано на підвищення прибутковості компанії та залучення нових клієнтів. Головна сторінка системи зображена не для авторизованого користувача на рис. 4.1.

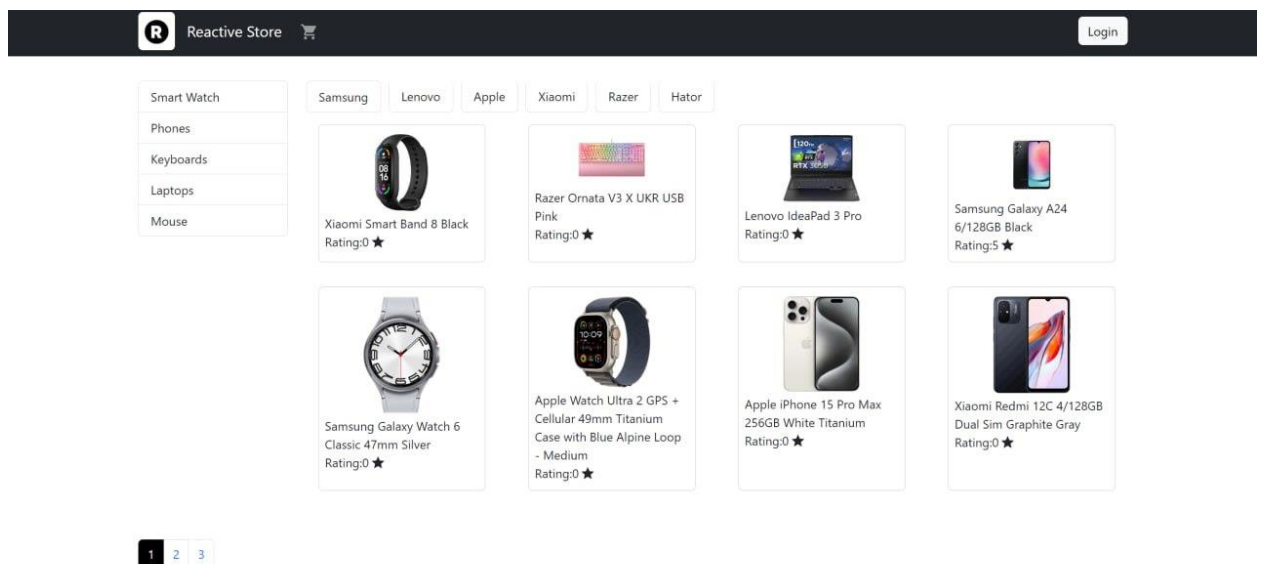


Рисунок 4.1 – Головна сторінка ІС інтернет-магазину

У верхній частині системи розміщене меню, на якому знаходиться посилання на пункти головного меню: кошик та авторизація. Основний контент представлений інформаційними картками про товар. Якщо натиснути на одну із них відкриється модальне вікно з більш детальною інформацією про товар.

Система фільтрації товарів у інтернет-магазині забезпечує користувачам можливість легко знаходити необхідні продукти, зручно організовуючи їх за типами та брендами. Цей функціонал створений для підвищення зручності та ефективності процесу пошуку товарів, дозволяючи користувачам швидко знайти те, що їм потрібно (рис. 4.2).

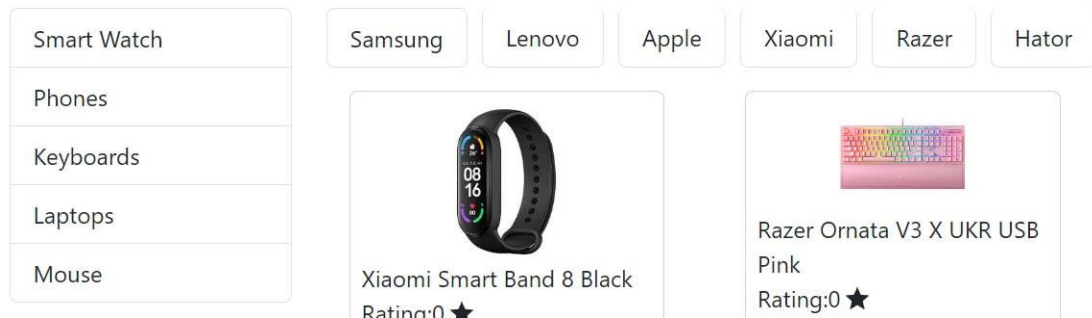


Рисунок 4.2 – Фільтри за «Типом» та «Брендом»

На лівій стороні сторінки розташований вертикальний список категорій товарів, таких як «Smart Watch», «Phones», «Keyboards», «Laptops», та «Mouse». Ці категорії дозволяють користувачам швидко звузити вибір до певного типу продуктів. Наприклад, натиснувши на типом "Smart Watch", користувач побачить лише смарт-годинники, що є у наявності, виключаючи всі інші типи товарів (рис. 4.3).

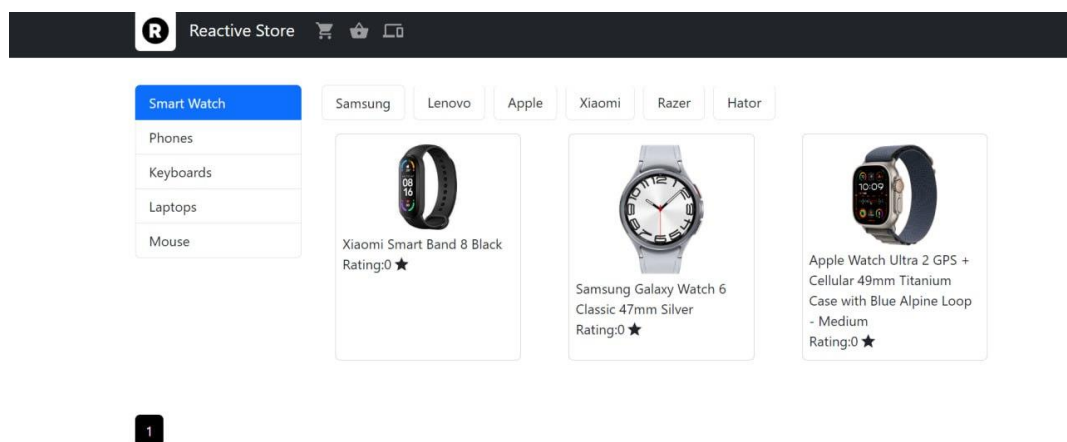


Рисунок 4.3 – Фільтри за типом «Smart Watch»

Над списком товарів розміщений горизонтальний рядок фільтрів за брендами, включаючи такі відомі бренди, як «Samsung», «Lenovo», «Apple», «Xiaomi», «Razer», та «Hator». Ці фільтри дозволяють користувачам обирати товари певного бренду. Натиснувши на будь-який бренд, наприклад, «Xiaomi», користувач побачить усі товари цього бренду, представлені в інтернет-магазині (рис. 4.4).

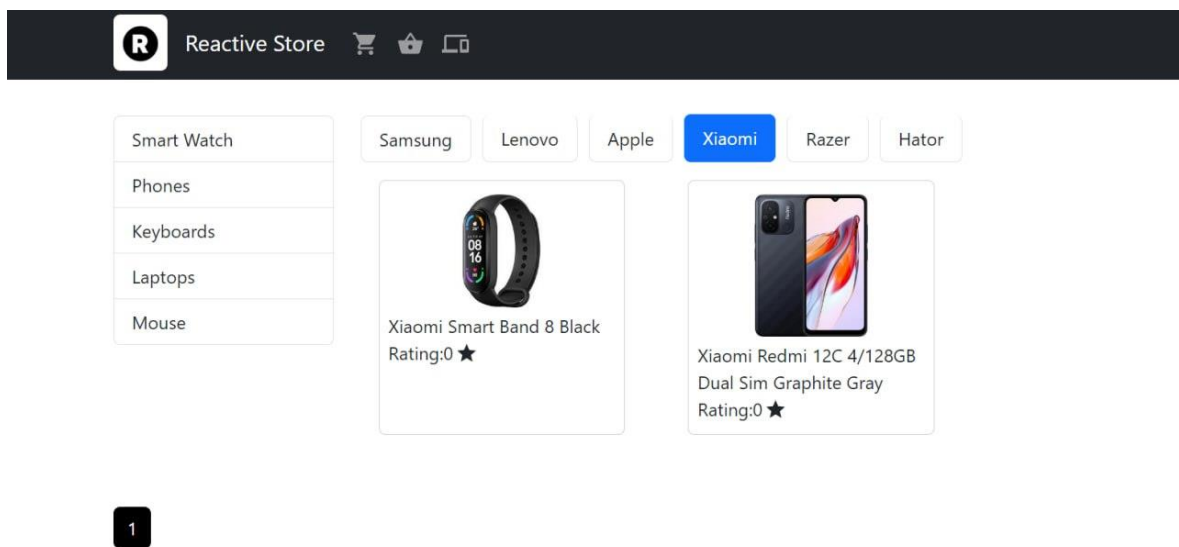


Рисунок 4.4 – Фільтри за брендом «Xiaomi»

Коли користувач застосовує фільтр за типом товару або брендом, список товарів автоматично оновлюється, відображаючи тільки ті продукти, що відповідають вибраним критеріям. Це дозволяє значно скоротити час пошуку потрібного товару та підвищити зручність користування сайтом.

На зображенні представлено приклад, де користувач обрав тип «Smart Watch» та бренд «Xiaomi» (рис. 4.5). У результаті на екрані відображено відповідні товари, такі як «Xiaomi Smart Band 8 Black». Це дає можливість користувачу легко порівнювати різні моделі та бренди, зосереджуючись лише на тих товарах, які його цікавлять.

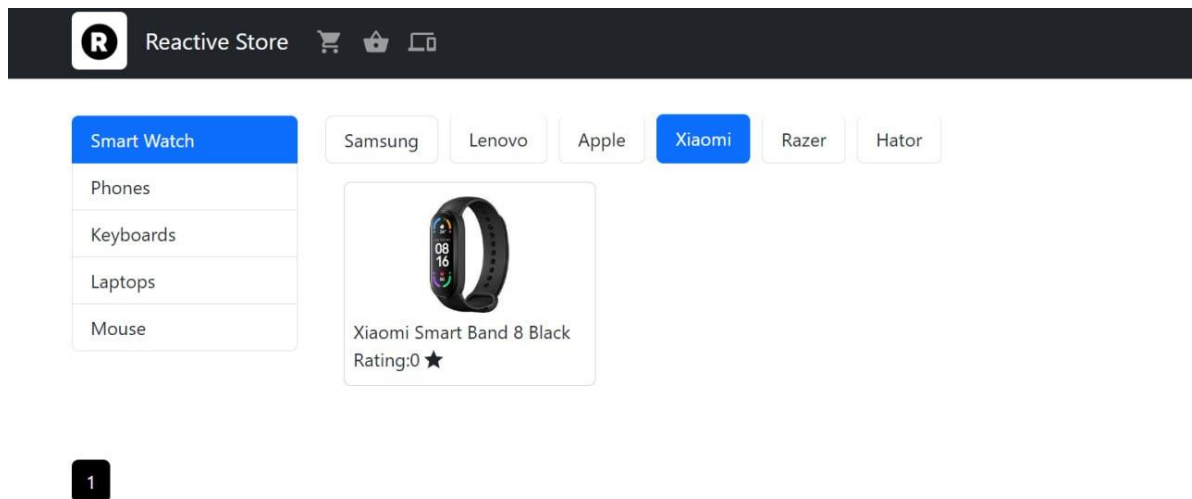


Рисунок 4.5 – Фільтри за брендом «Xiaomi» та типом «Smart Watch»

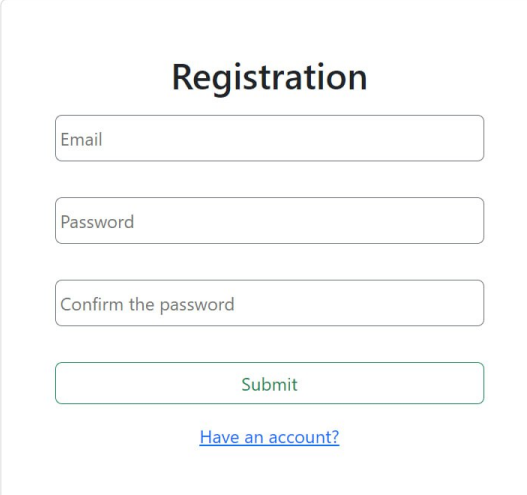
Фільтрація за типами та брендами також допомагає користувачам відкривати нові продукти та бренди, з якими вони раніше не стикалися. Цей інтуїтивний та ефективний інструмент робить процес пошуку та покупки товарів швидким та зручним, що позитивно впливає на загальне враження від користування інтернет-магазином.

В низу сторінки знаходиться компонент пагінації. Пагінація – це розбивка величезного масиву даних (зазвичай однотипного) на невеликі за обсягом сторінки. Причому з відображенням нумерованої навігації по ним. Найчастіше пагінація сторінок застосовується в інтернет-магазинах – в категоріях та розділах. Також вона зустрічається в блогах, де публікується дуже багато статей.

Правильно налаштована пагінація істотно збільшує час перебування користувача на сайті, покращує поведінкові характеристики. Це важливо для SEO просування.

Сторінка реєстрації призначена для створення нового облікового запису користувача. Вона містить просту та зрозумілу форму, яка дозволяє користувачам ввести свої дані для реєстрації. Інтерфейс сторінки має сучасний та мінімалістичний дизайн для полегшення процесу реєстрації (рис. 4.6).

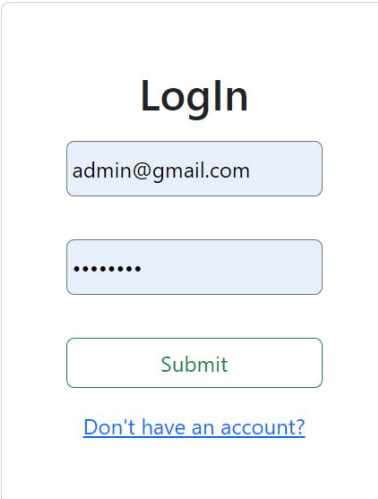




The registration form is titled "Registration" and is contained within a light gray rounded rectangle. It features four input fields: "Email", "Password", and "Confirm the password", each with a light gray border and rounded corners. Below these fields is a "Submit" button with a light gray border and rounded corners. At the bottom of the form is a blue hyperlink labeled "Have an account?".

Рисунок 4.6 – Сторінка реєстрації нового облікового запису клієнта

Сторінка логінації призначена для входу користувачів до їхніх облікових записів. Вона містить просту та зрозумілу форму, яка дозволяє користувачам ввести свої облікові дані для авторизації. Інтерфейс сторінки має сучасний та мінімалістичний дизайн для полегшення процесу входу (рис.4.7).



The login form is titled "Login" and is contained within a light gray rounded rectangle. It features two input fields: the first contains the email address "admin@gmail.com" and the second contains a masked password ".....", both with light gray borders and rounded corners. Below these fields is a "Submit" button with a light gray border and rounded corners. At the bottom of the form is a blue hyperlink labeled "Don't have an account?".

Рисунок 4.7 – Сторінка логінації існуючого облікового запису клієнта

Після успішної логінації або реєстрації користувач отримує доступ до ряду важливих функцій, що дозволяють йому повноцінно використовувати можливості інтернет-магазину. Основні функції включають додавання товарів до кошика, перегляд товарів у кошику, здійснення покупок та редагування замовлень.

Після входу в обліковий запис користувач може легко додавати товари до свого кошика. На сторінці товару присутня кнопка "To basket", яку користувач може натиснути, щоб додати обраний товар до свого кошика. Після натискання кнопки товар автоматично додається до кошика, і користувач може продовжити перегляд інших товарів або перейти до кошика.

Користувач має можливість переглянути вміст свого кошика у будь-який момент. Для цього він може натиснути на іконку кошика в верхній частині сторінки (рис. 4.8), що перенаправить його на сторінку кошика. На цій сторінці відображаються всі додані товари, їх кількість, ціна та загальна сума. Користувач може змінювати кількість товарів, видаляти непотрібні товари або переходити до оформлення замовлення.

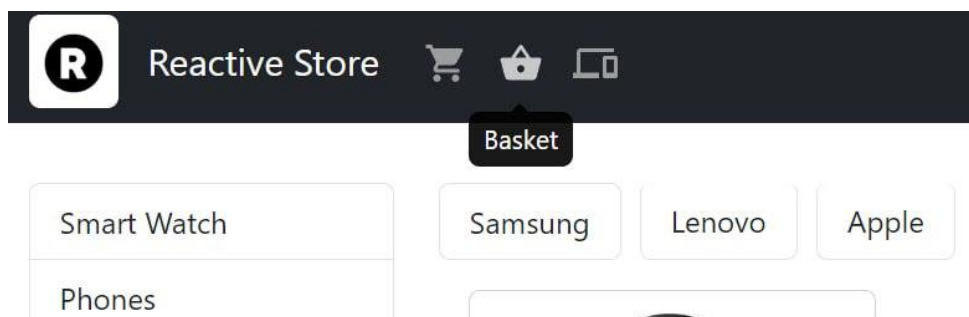


Рисунок 4.8 – Посилання на сторінку кошика з підписом

Після того як користувач задоволений вмістом свого кошика, він може перейти до оформлення замовлення. Після натискання на кнопку "Buy all" з вказаною сумарною ціною. Також він має можливість придбати лише один товар та можливість видалення товару (рис. 4.9).

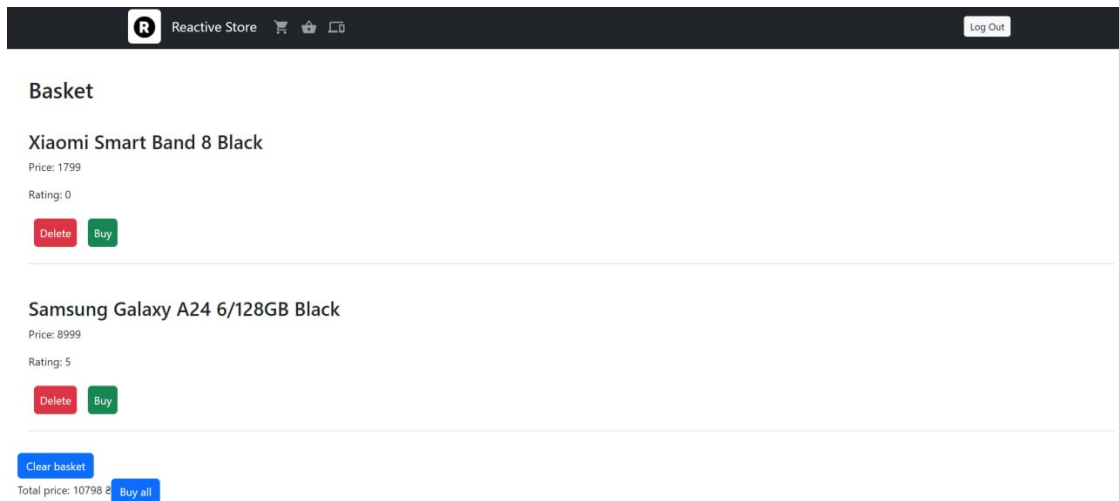


Рисунок 4.9 – Сторінка «Basket» з доданими товарами

Цей набір функцій забезпечує зручний та ефективний процес купівлі товарів, що дозволяє користувачам легко керувати своїми замовленнями та кошиком, забезпечуючи приємний користувацький досвід.

Після придбання одного або декількох товарів користувача повідомляють про покупку модальним вікном (рис. 4.10)

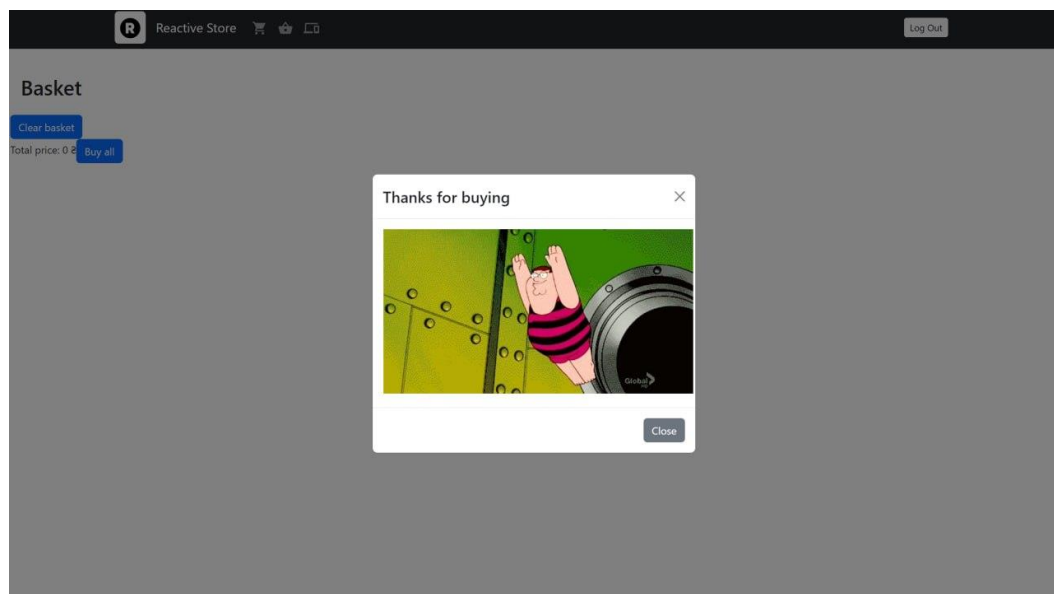
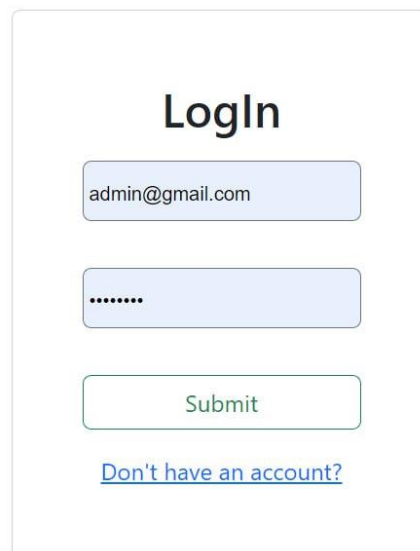


Рисунок 4.10 – Модальне вікно успішної покупки

## 4.2 Реалізація додатку користувача-адміністратора системи

Інтернет-магазин передбачає можливість адміністрування системою для забезпечення актуальності та достовірності інформації, яку використовуватимуть клієнти компанії. Користувач-адміністратор має можливість через інтерфейс адміністратора додавати типи товарів, бренди та самі товари з вказаними характеристиками та атрибутами. Для входу до адміністраторської панелі користувачеві-адміністратору необхідно буде ввести унікальний логін і пароль (рис. 4.11).



The image shows a login form with the following elements:

- Title: **Login**
- Input field 1: Contains the text "admin@gmail.com".
- Input field 2: Contains a series of dots "....." representing a password.
- Submit button: Labeled "Submit".
- Link: "[Don't have an account?](#)"

Рисунок 4.11 – Вхід в систему в якості адміністратора

Після введення правильного логіну і паролю адміністратор має можливість увійти в систему. У випадку, якщо логін або пароль вказані неправильно, система видає повідомлення "User not found". Управління системою та надання адміністратору всіх необхідних інструментів для адміністрування здійснюється через адміністраторську панель, де зібрані всі необхідні основні сторінки та категорії, які доступні у системі (рис. 4.12).

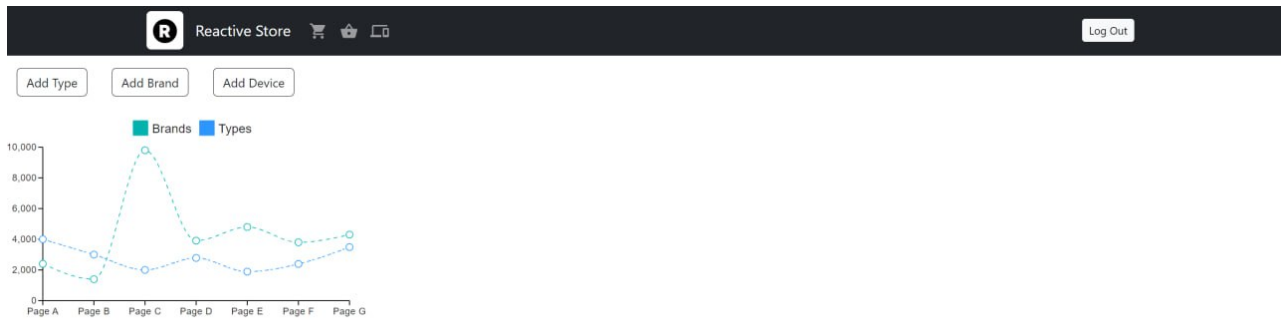


Рисунок 4.12 – Загальний вигляд сторінки адміністратора

Адміністратору доступна можливість створювати товари, які продає інтернет-магазин. Також для функціональності сайту адміністратор повинен мати можливість створювати нові типи та бренди, які потрібні для маркування товару. Приклад заповнення форми «Add type» адміністратором системи наведено на рис. 4.13.

The screenshot shows a modal window titled 'Create Type'. It has a close button (X) in the top right corner. Below the title is a text input field labeled 'Name of type'. At the bottom right of the modal, there are two buttons: 'Add' and 'Close'.

Рисунок 4.13 – Модальне вікно для створення типу товару

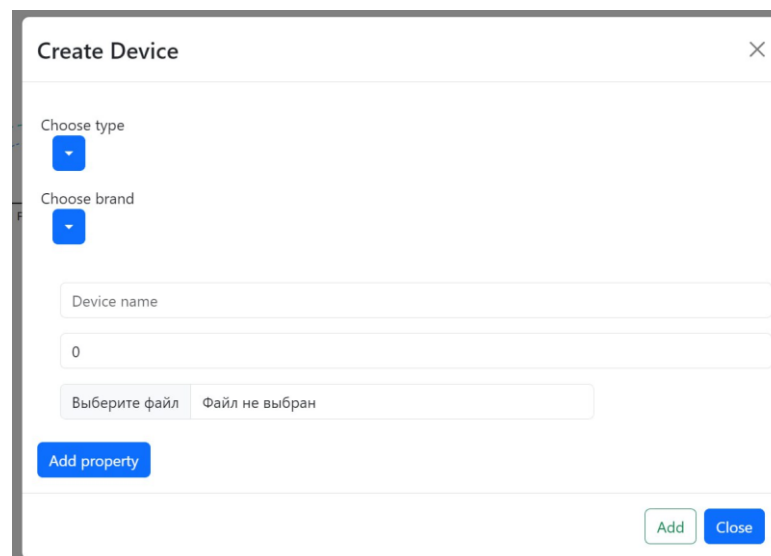
Системою передбачено для адміністратора функція функціонал для створення брендів, які потрібні для ідентифікації товару (рис. 4.14).

The screenshot shows a modal window titled 'Create Brand'. It has a close button (X) in the top right corner. Below the title is a text input field labeled 'Name of brand'. At the bottom right of the modal, there are two buttons: 'Add' and 'Close'.

Рисунок 4.14 – Модальне вікно для створення бренду товару

Для створення товару реалізовано модальне вікно (рис. 4.15) в якому потрібно заповнити поля даними про товар та натиснути кнопку «Add». Для створення товару треба заповнити такі поля як:

- обрати тип товару;
- обрати бренд товару;
- вказати назву товару;
- вказати ціну;
- завантажити фото товару;
- додати необхідну кількість полів параметрів;



The image shows a modal window titled "Create Device" with a close button in the top right corner. The window contains the following elements:

- A "Choose type" dropdown menu with a blue arrow icon.
- A "Choose brand" dropdown menu with a blue arrow icon.
- A text input field labeled "Device name".
- A text input field containing the value "0".
- A file selection field with the text "Выберите файл" and "Файл не выбран".
- A blue "Add property" button.
- At the bottom right, there are "Add" and "Close" buttons.

Рисунок 4.15 – Модальне вікно для створення товару

## ВИСНОВКИ

Дипломна робота спрямована на проектування та програмну реалізацію інтернет-магазину, який надає можливість користувачам на купівлю товарів через мережу Інтернет. Це, в свою чергу, збільшить прибуток компанії, оскільки здійснення електронних покупок сьогодні – це шлях до сучасного та ефективного розвитку бізнесу.

У ході виконання дипломного проекту було проведено детальний аналіз існуючих аналогічних ІС. Це дало змогу виявити не лише їх позитивні сторони але й недоліки, на підставі яких були визначені вимоги до розробки оптимального інтернет-магазину. Маючи результати аналізу, була розроблена оптимальна архітектура системи, вибрані підходящі засоби реалізації для її створення та розроблена база даних, яка забезпечує ефективне зберігання і оброблення інформації. Даний сервіс представлений у вигляді інтерфейсу інтернет-магазину, котрий включає в собі властивості цілісності, доступу, зручних діалогових вікон та якісний користувацький інтерфейс. Це дозволить створювати простий і зручний користувацький інтерфейс, який сприятиме привабленню та збереженню клієнтів. Кожний елемент інтерфейсу опрацьовано для надання клієнтам кращого досвіду користування, що підвищить їхню задоволеність та лояльність до бренду.

Для реалізації системи використовувалися сучасні WEB-технології та програмні засоби, зокрема JS-фреймворки React.js та Node.js, СУБД PostgreSQL, а також технології HTML, CSS та бібліотеки Bootstrap, Express.js та інші. Використання цих інструментів дозволило створити надійну, масштабовану та гнучку систему, яка відповідає сучасним вимогам і стандартам веб-розробки. Це забезпечує можливість легкої інтеграції з іншими системами, що дозволяє розширювати функціонал інтернет-магазину відповідно до зростаючих потреб бізнесу.

Впровадження таких передових технологій, як React.js та Node.js, забезпечує високу продуктивність і швидкодію системи, що є критично важливим для онлайн-комерції, де кожна секунда затримки може вплинути на рішення клієнта про покупку. Використання PostgreSQL як системи управління базами даних забезпечує надійне зберігання і швидкий доступ до великих обсягів даних, що є важливим для підтримки актуальності та точності інформації про товари.

Загалом, розробка цього інтернет-магазину не лише підвищує ефективність комерційної діяльності компанії, але й забезпечує їй конкурентні переваги на ринку за рахунок сучасних технологій та зручного інтерфейсу для користувачів.



## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. React.js. URL: <https://ru.legacy.reactjs.org> (дата звернення: 19.05.2024)
2. Node.js. URL: <https://nodejs.org/en> (дата звернення: 19.05.2024)
3. Express.js. URL: <https://expressjs.com> (дата звернення: 19.05.2024)
4. Компанія «Rozetka». URL: <https://rozetka.com.ua> (дата звернення: 20.05.2024)
5. Компанія «Comfy». URL: <https://comfy.ua/ua/odesa/> (дата звернення: 20.05.2024)
6. Компанія «Citrus». URL: <https://www.ctrs.com.ua/> (дата звернення: 20.05.2024)
7. Сімпсон Кайл. Ви не знаєте JS. 2019. 240 с.
8. Фреймворк – важливий інструмент програміста. URL: <https://fructcode.com/ru/blog/features-of-popular-frameworks-html-css-php-and-python-frameworks/> (дата звернення: 02.06.2024)
9. JavaScript Guide. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (дата звернення: 02.06.2024)
10. TypeScript is JavaScript with syntax for types. URL: <https://www.typescriptlang.org> (дата звернення: 02.06.2024)
11. Закон України «Про електронну комерцію». URL: <https://zakon.rada.gov.ua/laws/show/675-19#Text>
12. Електронна комерція в Україні: Правила бізнесу. URL: [https://biz.ligazakon.net/analytics/226702\\_elektronna-komertsya-v-ukran-pravila-dlya-bznesu](https://biz.ligazakon.net/analytics/226702_elektronna-komertsya-v-ukran-pravila-dlya-bznesu)