



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної
підготовки

Випускна кваліфікаційна роботи бакалавра
«СИСТЕМА УПРАВЛІННЯ КОНТЕНТОМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ
(за прикладом сервісу Comfy)»

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проєкту

№	П.І.Б	Група
1	Ісаєнко Д.О	КН-П-191
2	Бондарчук М.Є	КН-П-191
3	Єлізар`єв В.А	КН-П-191
4	Лавров А.В	
5	Сідху Т.	
6	Тітова О.Є	

Автор звіту:
Ісаєнко Дмитро Олександрович

АНОТАЦІЯ

Ісаєнко Д.О. Система управління контентом електронної комерції.
Автореферат випускної кваліфікаційної роботи на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». - Одеса: ОТУШ, 2023.

Робота присвячена дослідженню систем управління контентом електронної комерції. Вивчаються основні аспекти, функції та переваги систем управління контентом.

Також у роботі розглядаються основні функції системи управління контентом, включаючи управління товарами, каталогами, замовленнями, корзинами та іншими елементами комерційного контенту. Пояснюється, як система допомагає покупцям швидко знайти потрібні товари, робить процес замовлення зручним та забезпечує ефективне керування товарами.

Актуальність роботи визначається реалізацією задач, які ґрунтуються на зберіганні, сортуванні товарів за різними характеристиками та категоріями, щоб мати змогу надати користувачу доступ до цих товарів.

Об'єктом дослідження у рамках даної роботи виступає розробка бази даних та комплексної системи управління товарами. А Також розробка інтерфейсу для надання послуг з замовлень товарів клієнтам.

ЗМІСТ

ВСТУП _____	3
Технічне завдання до проєкту _____	4
РОЗДІЛ 1. Загальна характеристика системи _____	8
РОЗДІЛ 2. Реалізація серверної частини _____	9
РОЗДІЛ 3. Реалізація адмін панелі _____	13
РОЗДІЛ 4. Результати випробувань _____	17
ВИСНОВКИ _____	18
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ _____	19

ВСТУП

Інтернет-технології надають безліч можливостей для спрощення і поліпшення різних сфер життя, включаючи торгівлю. Зростання популярності онлайн-шопінгу та зростання попиту на зручні та ефективні способи покупок стимулюють розвиток глобального електронного ринку. У цьому контексті виникає потреба в рішеннях, що дозволяють споживачам здійснювати покупки онлайн зі зручністю та ефективністю, а продавцям - розширювати свій бізнес та досягати нових ринків та аудиторій.

Онлайн гіпермаркети виступають як відповідь на ці вимоги, поєднуючи широкий асортимент товарів, зручний інтерфейс та безліч послуг, що полегшують процес покупок.

У даному документі розглядається концепція і розробка онлайн гіпермаркету, який відповідає потребам споживачів і сприяє ефективному функціонуванню бізнесу. Основною метою дослідження є створення інноваційної цифрової платформи, яка забезпечує безперервний доступ до широкого спектра товарів, зручну навігацію та високу якість обслуговування.

Автореферат розпочинається аналізом технічних вимог до проєкту. Далі проводиться проєктування архітектури онлайн гіпермаркету, враховуючи вимоги зручності, безпеки та ефективності. Потім розглядаються етапи розробки, включаючи вибір технологій, розробку серверної та frontend частини. Висновки дослідження демонструють результати, отримані під час розробки, тестування та використання онлайн гіпермаркету. В цілому, автореферат дозволяє зрозуміти сутність та переваги цього рішення.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЄКТУ

Загальний опис

Програмне забезпечення для створення інформаційної системи інтернет крамниці для організації роздрібної інтернет торгівлі.

Призначення

Надання в Інтернет просторі інформації про доступні товари (оферти продавця) та організації оформлення замовлень (акцепту покупця) через мережу Інтернет. Оперативне управління веб-контентом Інтернет крамниці. Збереження даних та регулювання доступу до них. Організація взаємодії з іншими інформаційними системами.

Вимоги до функціоналу:

- Інтернет крамниця
 - Навігація по веб вітрині крамниці
 - Пошук товарів за критеріями
 - Реєстрація покупців
 - Формування кошика покупця
 - Оформлення замовлень
 - Зворотній зв'язок та відгуки
- Управління контентом інтернет крамниці
 - Навігація, перегляд, створення, редагування структури та контенту
 - Керування доступом та розподіл дозволів користувачів
 - Створення вітрини товарів

- Управління даними користувачів
- Оформлення замовлень
- Модерація коментарів
- Реєстрація та авторизація користувачів
 - Користувач ПЗ має змогу зареєструватись чи увійти у систему як за допомогою авторизаційних даних (email та пароль), так і за допомогою облікових записів популярних соціальних мереж (Google, тощо)
 - Користувач ПЗ має змогу увімкнути та вимкнути функцію двофакторної автентифікації
- Особистий кабінет покупця
 - Перегляд історії замовлень
 - Кошик покупця
 - Реквізити для платежів та доставки
- Взаємодія із інтернет додатками
 - Web-API моделі та запити для управління аккаунтом користувача
 - Web-API моделі та запити для одержання, створення та редагування інформації, що формує профіль покупця
 - Web-API моделі та запити для вибору товару в кошику та оформлення замовлення користувачем
 - Web-API моделі та запити для створення питань та відгуків, а також для їх оцінок (лайків/дізлайків)
 - Web-API моделі та запити для пошуку товарів з необхідними фільтрацією та сортуванням

- Web-API моделі та запити для отримання категорій товарів, банерів, тощо

Архітектура:

- Сервер бази даних - серверне програмне забезпечення, що забезпечує збереження та обробку поточних даних.
- Сервіс Web-API – серверне програмне забезпечення, що надає програмний інтерфейс для взаємодії (REST-API) з іншим програмним забезпеченням.
- Веб додаток керування контентом - доступний з веб-простору сайт, що забезпечує візуальний інтерфейс для менеджерів системи.
- Веб крамниця - доступний з веб-простору сайт, що дозволяє взаємодіяти з інтернет-магазином покупцям.

Технології:

- Для кожного з модулів рекомендується (з можливістю часткової зміни) наступні технології та методології створення:
 - Бази даних – PostgreSQL, Redis
 - Сервіс Web-API – REST-API, ASP.NET Core, C#
 - Веб додаток керування контентом – MVC, ASP.NET Core, Razor Views, C#
 - Веб додаток для покупця – Node JS, React, Redux
- При розгортанні програмного забезпечення рекомендується задіяти хмарні технології

- При розробці програмного забезпечення необхідно задіяти зберігання вихідних кодів ПЗ повинно здійснюватись за допомогою системи контролю версій (вибір конкретної системи узгоджується групою проекту).
- Вихідний код повинен оформлятися у відповідності до загальноприйнятих стандартів, на зразок стандартів GNU <http://www.gnu.org/prep/standards/>, у тому числі з дотриманням вимог до коментування та документування коду.

РОЗДІЛ 1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ

У відповідності до технічного завдання (ТЗ) система була спроектована за клієнт-серверною розподіленою архітектурою. У складі системи було умовно відокремлено наступні архітектурні частини:

- «UI/UX»,
- «Backend»,
- «Frontend».

Частина UI/UX відповідає за дизайн проекту. Дизайн представляє собою набір моделей котрі створюють інтерфейс для взаємодій з користувачем, забезпечуючи йому інтуїтивно зрозумілий вигляд. При розробці та проектуванні інтерфейсів були використані основні принципи UI/UX дизайну.

Частина дизайну розміщена за посиланням:

[https://www.figma.com/file/PETLgTkhYxAYbqXE2jHX30/Untitled-\(Copy\)?type=design&node-id=0-1&mode=design&t=hPSPT9oSrbfJzsJy-0](https://www.figma.com/file/PETLgTkhYxAYbqXE2jHX30/Untitled-(Copy)?type=design&node-id=0-1&mode=design&t=hPSPT9oSrbfJzsJy-0)

Backend - це серверне програмне забезпечення, яке відповідає за збереження, обробку, модерування та надання актуальних даних проекту протягом усього життєвого циклу інформаційної системи. Частина Backend реалізована з використанням СУБД PostgreSQL та ASP.NET Core. Вона включає базу даних і надає програмний інтерфейс взаємодії (API) для доступу до різних функціональностей, такі як реєстрація, авторизація, надання відгуків та оцінок товару тощо.

Backend частина розміщена за посиланням:

<https://github.com/storedev2023>

Частина Frontend - це клієнтська частина системи, котра призначена для надання користувачу доступу до різних функцій системи шляхом реалізації візуального інтерфейсу. Функції цієї частини включають: порівняння товарів, створення відгуків, розміщення замовлень. Вона також забезпечує можливість надання користувачу таких функціональностей, як пошук, сортування і фільтрацію товарів.

Частина Frontend - розміщена за посиланням:

<https://github.com/storedev2023/Comfy.Frontend>

РОЗДІЛ 2

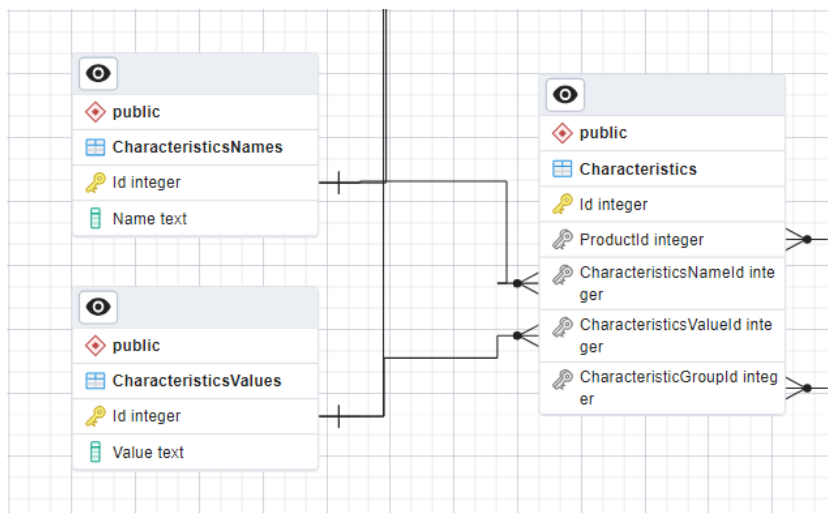
РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ

Сервера частина написана з використанням наступних технологій:

- Мова програмування C#
- Фреймворк ASP.NET Core 6
- База даних PostgreSQL, MySQL
- Інструмент адміністрування бази даних PgAdmin, phpMyAdmin
- Середовище розробки Visual Studio
- Платформа для тестування та розгортання додатків Docker

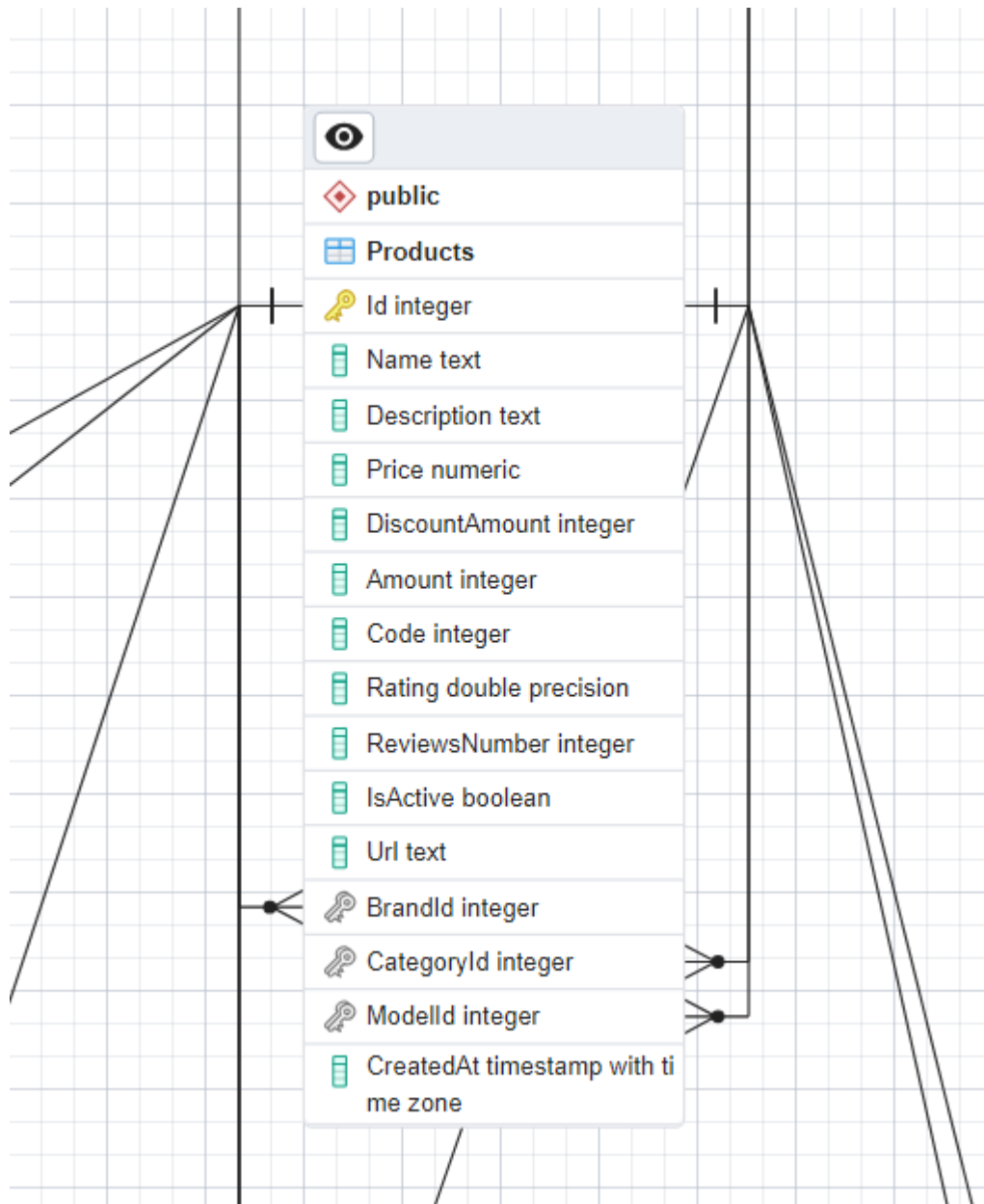
У даному розділі розглядається реалізація серверної частини системи, зокрема розробка та впровадження бази даних. База даних є одним з ключових компонентів ефективної та надійної роботи системи управління. Вона забезпечує зберігання, організацію та доступ до потрібної інформації, необхідної для функціонування системи управління контентом електронної комерції. Реалізація бази даних включає в себе проектування її структури, створення необхідних таблиць, встановлення зв'язків між ними, а також розробку механізмів для збереження, оновлення та вибірки даних. Оптимальна організація бази даних дозволяє ефективно виконувати запити та операції, забезпечувати швидкий доступ до інформації та забезпечувати цілісність та безпеку даних.

Основним було створити структуру для характеристик товару, бо товари усі різні і їх багато і кожен з них має свої унікальні характеристики.



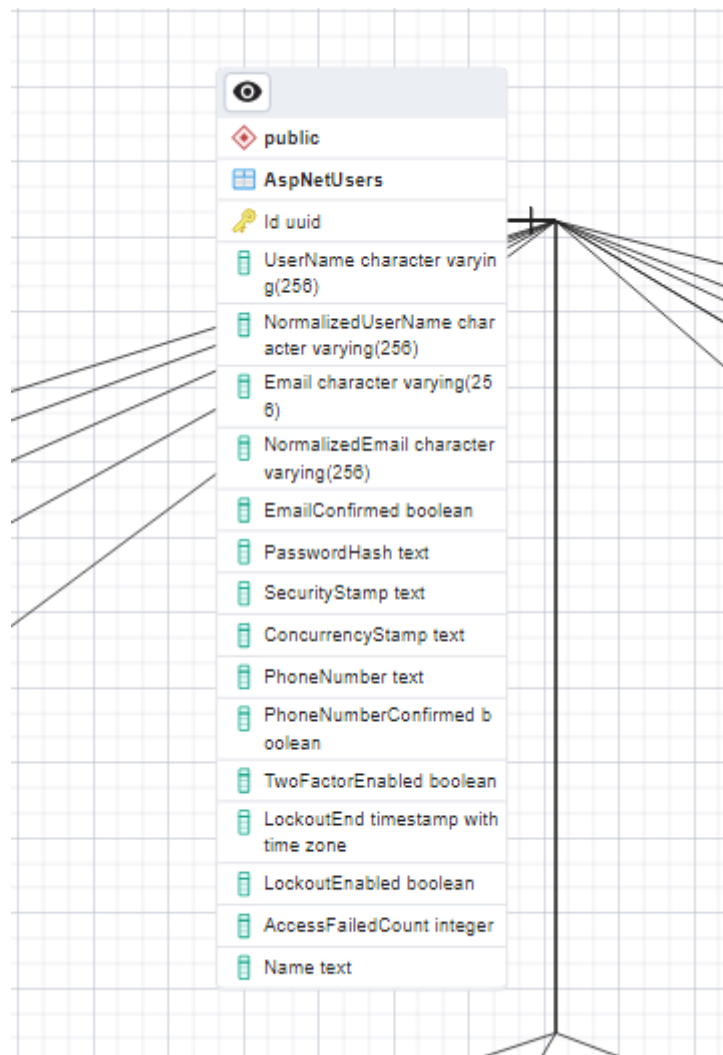
В базі присутні 3 таблиці пов'язаних з характеристиками. Перша таблиця містить в собі назву характеристики, друга містить її значення. Третя таблиця поєднує в собі ці значення та зберігає id товару до якого ці характеристики належать.

В основному користувач на сайті взаємодіє з товарами. Кожен товар має декілька спільних характеристик таких як: назва, опис, рейтинг, кількість, ціну та код товару.



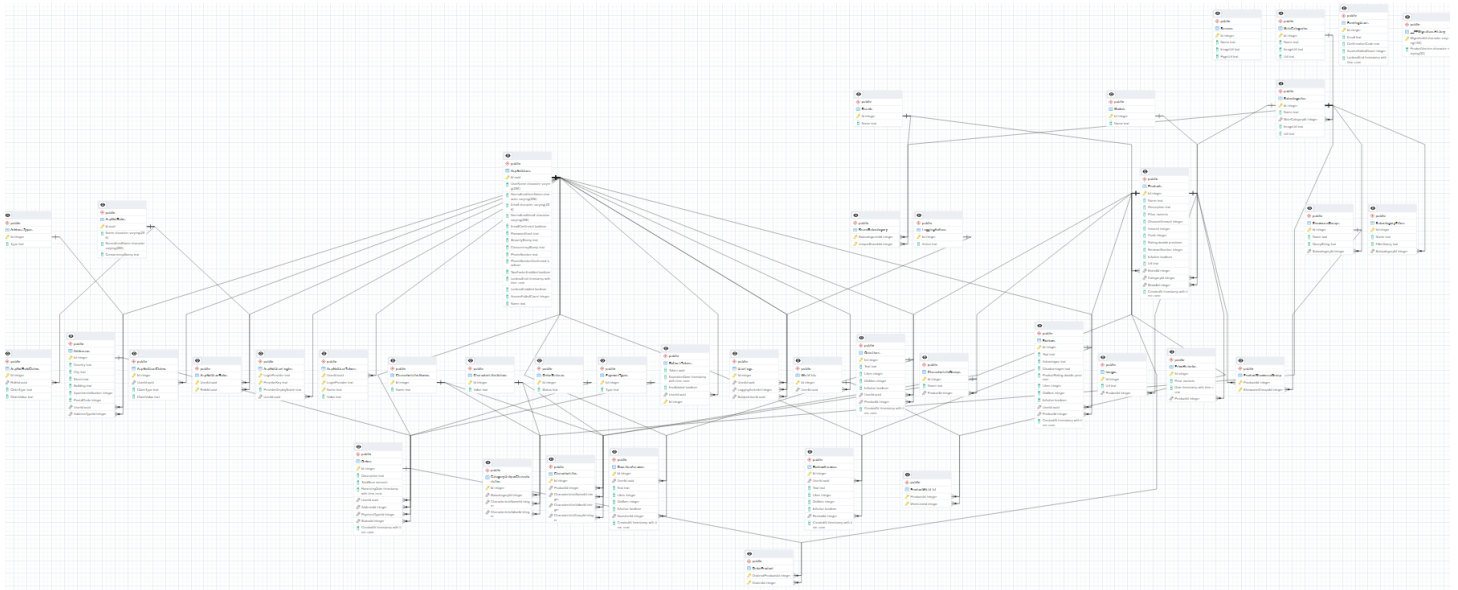
Також в таблиці товару присутній id його таблиці з характеристиками.

В нашій системі придбати товар може тільки зареєстрований користувач, тому в базі повинна бути таблиця котра зберігає усіх користувачів в тому числі і адміністрацію сайту.



Таблиця користувачів виконана за допомогою ASP.NET Core Identity котрий є фреймворком для управління аутентифікацією, авторизацією та управлінням користувачами в додатках на платформі ASP.NET Core. Він надає готові рішення для реалізації функцій, таких як реєстрація користувачів, вход в систему, управління ролями та правами доступу. ASP.NET Core Identity забезпечує основні можливості, необхідні для безпечного та ефективного управління користувачами.

База створена за всіма нормальними формами, та з підтримкою додавання нових функцій у систему. В загалом в базі 41 таблиця, кожна з котрих взаємодіє з іншими таблицями за допомогою різних типів зв'язків, таких як: один до одного (One-to-One): У цьому типі зв'язку один запис в одній таблиці відповідає одному запису в іншій таблиці. Цей тип зв'язку використовується тоді, коли існує пряма взаємодія між двома сутностями, і кожна з них має унікальну специфікацію, яка не може бути пов'язана з багатьма записами іншої таблиці, Один до багатьох (One-to-Many): Це найпоширеніший тип зв'язку, де один запис в одній таблиці відповідає багатьом записам в іншій таблиці. Наприклад, у базі даних інтернет-магазину один користувач може мати кілька замовлень, багато до багатьох (Many-to-Many): В цьому типі зв'язку багато записів в одній таблиці відповідає багатьом записам в іншій таблиці, і навпаки. Цей тип зв'язку вимагає використання проміжної таблиці або "таблиці зв'язку", яка містить зв'язки між об'єктами.



РОЗДІЛ 3

РЕАЛІЗАЦІЯ АДМІН ПАНЕЛІ

Frontend частина написана з використанням наступних технологій:

- HTML5
- Js, Ajax, Razor pages
- Середовище розробки Visual Studio та Visual Studio

У складі frontend частини адмін панелі передбачено та реалізовано такі функції:

- Авторизація користувача
- Створення та редагування користувача
- Створення товару
- Редагування товару
- Відображення усіх товарів та їх пошук
- Відображення та редагування категорій
- Відображення та керування відгуками та питаннями
- Редагування банерів
- Журнал користувачів

Початок роботи з адмін панеллю починається з головної сторінки з головним меню адмін панелі, в якому є доступ до всіх сторінок панелі. Нижче наведений приклад відображення меню:

```
var URL = document.location.pathname;
var userMainPosition = URL.split("/")

var div = document.createElement("div")
div.classList.add("subLink")
var ul = document.createElement("ul")
var li, a, span
var links, href

function blockMainLink(elem) {
    elem.querySelector(".menu-link").removeAttribute("href")
}

function createLI(links, href) {
    for (var i = 0; i < links.length ; i++) {
        li = document.createElement("li")
        a = document.createElement("a")

        a.setAttribute("href", href[i])
        a.textContent = links[i]

        li.appendChild(a)
        ul.appendChild(li)
    }
}
```

```

switch (userMainPosition[1]) {

    case "Orders":
        let divMain = document.getElementById("Main");
        blockMainLink(divMain)

        links = ["Усі замовлення"];
        href = ["/Orders/Orders"]
        createLI(links, href)
        div.appendChild(ul)
        divMain.appendChild(div)
        break;

    case "Products":
        let divGoods = document.getElementById("Goods");
        blockMainLink(divGoods)

        links = ["Товари", "Бренди", "Моделі", "Вітрина товарів", "Категорії", "Фільтри підкатегорій"];
        href = ["/Products/Products", "/AdminPanel/Brands", "/AdminPanel/Models", "/Showcase/Index", "/Categories/Index", "/Categories/SubcategoryFilters"]
        createLI(links, href)
        div.appendChild(ul)
        divGoods.appendChild(div)
        break;

    case "Comments":
        let divComments = document.getElementById("Comments");
        blockMainLink(divComments)

        links = ["Активні питання", "Активні відгуки", "Активні відповіді на питання", "Активні відповіді на відгуки", "Неактивні питання",
            "Неактивні відгуки", "Неактивні відповіді на питання", "Неактивні відповіді на відгуки"];
        href = ["/Comments/ActiveQuestions", "/Comments/ActiveReviews", "/Comments/ActiveQuestionAnswers", "/Comments/ActiveReviewAnswers",
            "/Comments/InactiveQuestions", "/Comments/InactiveReviews", "/Comments/InactiveQuestionAnswers", "/Comments/InactiveReviewAnswers"]
        createLI(links, href)
        div.appendChild(ul)
        divComments.appendChild(div)
        break;

```

```

case "AdminPanel":
    let divApanel = document.getElementById("Goods");
    blockMainLink(divApanel)

    links = ["Товари", "Бренди", "Моделі", "Вітрина товарів", "Категорії", "Фільтри підкатегорій"];
    href = ["/Products/Products", "/AdminPanel/Brands", "/AdminPanel/Models", "/Showcase/Index", "/Categories/Index", "/Categories/SubcategoryFilters"]
    createLI(links, href)
    div.appendChild(ul)
    divApanel.appendChild(div)
    break;

case "Categories":
    let divCategories = document.getElementById("Goods");
    blockMainLink(divCategories)

    links = ["Товари", "Бренди", "Моделі", "Вітрина товарів", "Категорії", "Фільтри підкатегорій"];
    href = ["/Products/Products", "/AdminPanel/Brands", "/AdminPanel/Models", "/Showcase/Index", "/Categories/Index", "/Categories/SubcategoryFilters"]

    createLI(links, href)

    div.appendChild(ul)
    divCategories.appendChild(div)

    break;

case "Showcase":
    let divShowcase = document.getElementById("Goods");
    blockMainLink(divShowcase)

    links = ["Товари", "Бренди", "Моделі", "Вітрина товарів", "Категорії", "Фільтри підкатегорій"];
    href = ["/Products/Products", "/AdminPanel/Brands", "/AdminPanel/Models", "/Showcase/Index", "/Categories/Index", "/Categories/SubcategoryFilters"]

    createLI(links, href)

    div.appendChild(ul)
    divShowcase.appendChild(div)

    break;

```

Сторінка додавання товару є важливою частиною інтернет-маркетплейсу, оскільки вона надає можливість додавати нові товари до асортименту.

Описуючи цю сторінку для, можна включити такі елементи:

Форма додавання товару: Створення товару зазвичай виконується за допомогою форми, яка містить поля для введення інформації про товар. Ці поля включають назву товару, опис, ціну, бренд та категорію.

Валідація даних: Для забезпечення правильності та консистентності даних, на сторінці додавання товару застосовані правила валідації. Поля для введення перевіряються на обов'язковість заповнення, коректність формату та інші обмеження.

Створення товару

Основна інформація

Бренд
LG

Модель
Телевізор LG

Назва позиції
Введіть назву позиції...

Опис

Ціна, грн
Ціна

Головна категорія
Оберіть категорію

Створити товар

Сторінка редагування товарів також є необхідною складовою інтернет-маркетплейсу, оскільки вона надає можливість вносити зміни до існуючих товарів у асортименті. При описі цієї сторінки можна використати такі елементи:

Форма редагування товару: Редагування товару зазвичай здійснюється за допомогою форми, що містить поля для введення інформації про товар. Ці поля можуть включати назву товару, опис, ціну, бренд та категорію.

Зображення товару: На сторінці редагування товару може бути можливість змінювати або оновлювати зображення товару. Це дає можливість оновити візуальне представлення товару для покупців.

Редагування товару

Основна інформація

Бренд

Apple

Модель

iPhone 14

Назва позиції

Смартфон Apple iPhone 14 128Gb Midnight

Опис

Більше ніж вражає.

Ціна, грн

38999

Розмір знижки

10

Рейтинг

4.33

Головна категорія

Смартфони та телефони

Підкатегорія

Смартфони

Видимість

Активовано

Деактивовано

Зображення





Загрузка зображень
Оберть зображення

Вибрати файли

Файл не вибрано

Зберегти зміни

Загрузити

Також на сторінці редагування товару йому можна надати характеристики та групи характеристик.

Редагування товару

Смартфони

Вибрати файли | Файл не вибрано

Зберегти зміни

Загрузити

Додати групу характеристик

Ім'я

Додати

Список наявних груп і характеристик

Загальні

Видалити групу

Редагувати групу

Назва характеристики

Значення

Додати характеристику

Назва	Значення		
Діагональ дисплея	6,7"	Редагувати	Видалити
Роздільна здатність екрану	2796x1290	Редагувати	Видалити
Тип екрану	Super Retina XDR	Редагувати	Видалити
Частота оновлення екрану	120 Гц	Редагувати	Видалити

Зв'язок

Видалити групу

Редагувати групу

Назва характеристики

Значення

Додати характеристику

Назва	Значення		
Кількість SIM-карт	2 SIM	Редагувати	Видалити
Розмір SIM-карти	Nano-SIM	Редагувати	Видалити
Підтримка e-SIM	Є	Редагувати	Видалити

ОС

Видалити групу

Редагувати групу

Назва характеристики

Значення

Додати характеристику

Назва	Значення		
Операційна система	Apple iOS 16	Редагувати	Видалити

РОЗДІЛ 4

РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

Результатом компіляції програмного коду є з'єднання серверної та клієнтської частини. Далі ми можемо розмістити наш проект на будь-якому хостингу та користуватися цим продуктом.

Початок роботи з інтернет-магазином розпочинається з головної сторінки, де він може подивитися на нові товари. Також користувач може переглянути товари за категоріями або скористатися пошуком та фільтрами характеристик. Для замовлення товару користувачеві необхідно додати товар до кошика і бути зареєстрованим і авторизованим на сайті.

Після того, як користувач успішно завершує процес оформлення замовлення, система реєструє ці дані і передає їх до бази даних для збереження. Звідки вони потрапляють на обробку до адмін панелі, звідки можуть оброблятися адміністраторами.

Всі функції клієнтської та серверної частин, які були передбачені в технічному завданні, були успішно реалізовані та протестовані. Поточна версія додатку, що включає клієнтську та серверну частину, доступна для завантаження з репозиторію за адресою <https://github.com/storedev2023>.

ВИСНОВКИ

Проведена робота над розробкою інтернет магазину за прикладом сервісу Comfy. Були реалізовані всі основні можливості клієнтської та серверної частини продукту.

Була розроблена, втілена та протестована система взаємодії між надавачем послуг і клієнтом, що сприяє швидкому пошуку, фільтрації та вибору пропозицій для клієнтів.

Система використовує розподілену клієнт-серверну архітектуру і складається з трьох компонентів, що знаходяться під наступними адресами:

- Частина дизайну - [https://www.figma.com/file/PETLgTkhYxAYbqXE2jHX30/Untitled-\(Copy\)?type=design&node-id=0-1&t=myj3NjblQoztUXro-0](https://www.figma.com/file/PETLgTkhYxAYbqXE2jHX30/Untitled-(Copy)?type=design&node-id=0-1&t=myj3NjblQoztUXro-0)
- Backend частина - <https://github.com/storedev2023>
- Frontend частина - <https://github.com/storedev2023/Comfy.Frontend>

Для перевірки працездатності та зручності для користувача система була піддана випробуванню шляхом запуску на різних пристроях і проведення відповідних тестів. Результати випробування підтвердили, що система працює надійно і задовольняє вимоги щодо продуктивності та зручності.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Microsoft sql documentation - <https://learn.microsoft.com/en-us/sql/?view=sql-server-ver16>
2. Description of the database normalization basics - <https://learn.microsoft.com/uk-ua/office/troubleshoot/access/database-normalization-description>
3. SQL Server Express LocalDB - <https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-express-localdb?view=sql-server-ver16>
4. Creating Foreign Key Relationships - <https://learn.microsoft.com/en-gb/sql/relational-databases/tables/create-foreign-key-relationships?view=sql-server-ver16>
5. Metanit - <https://metanit.com/sql/>
6. Посібник з MS SQL Server 2022 - <https://metanit.com/sql/sqlserver/>
7. Посібник з PostgreSQL - <https://metanit.com/sql/postgresql/>
8. phpMyAdmin documentation - <https://www.phpmyadmin.net/docs/>
9. GitHub - <https://github.com/>
10. GitHub Desktop - <https://desktop.github.com/>
11. YouTube - <https://youtu.be/q5wFWfsS-4I>
12. StackOverflow - <https://stackoverflow.com/questions/tagged/sql>
13. React - <https://uk.legacy.reactjs.org/>
14. Ajax - <https://habr.com/ru/articles/14246/>