

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему: «**РОЗРОБКА ІНТЕРНЕТ МАГАЗИНУ  
ЕЛЕКТРОНІКИ**»

Виконав: студент 4 курсу, групи 6.1219-1 пі  
спеціальності 121 інженерія програмного забезпечення  
(шифр і назва спеціальності)  
освітньої програми програмна інженерія  
(назва освітньої програми)

М.С. Цис

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,  
доцент, к.т.н. Мухін В.В.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри загальної математики,  
доцент, к.ф.-м.н. Зіновєєв І.В.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний  
Кафедра програмної інженерії  
Рівень вищої освіти бакалавр  
Спеціальність 121 інженерія програмного забезпечення  
(шифр і назва)  
Освітня програма програмна інженерія

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри програмної  
інженерії, к.ф.-м.н., доцент

\_\_\_\_\_ Лісняк А.О.  
(підпис)

“ 07 ” 02 2023 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Цису Максиму Сергійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка інтернет магазину електроніки

керівник роботи Мухін Віталій Вікторович, к.т.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 26 » січня 2023 року № 102-с

2. Строк подання студентом роботи 07.06.2023 р.

3. Вихідні дані до роботи 1. Постановка задачі.  
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Моделювання та проектування інформаційної системи.

4. Реалізація та тестування інформаційної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

презентація за темою докладу

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------|---|----------------|------------------|
|        |   | Завдання видав | Завдання прийняв |
|        |   |                |                  |
|        |   |                |                  |
|        |   |                |                  |

7. Дата видачі завдання 07.02.2023 р.

### КАЛЕНДАРНИЙ ПЛАН

| №  | Назва етапів кваліфікаційної роботи                           | Строк виконання етапів роботи | Примітка |
|----|---|-------------------------------|----------|
| 1. | Розробка плану роботи.  | 08.02.2023                    |          |
| 2. | Збір вихідних даних.  | 15.02.2023                    |          |
| 3. | Обробка методичних та теоретичних джерел.                     | 09.03.2023                    |          |
| 4. | Розробка першого та другого розділу.                          | 14.04.2023                    |          |
| 5. | Розробка третього розділу.                                    | 15.05.2023                    |          |
| 6. | Оформлення та нормоконтроль кваліфікаційної роботи бакалавра. | 01.06.2023                    |          |
| 7. | Захист кваліфікаційної роботи.                                | 22.06.2023                    |          |

Студент \_\_\_\_\_  
(підпис)

М.С. Цис  
\_\_\_\_\_  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

В.В. Мухін  
\_\_\_\_\_  
(ініціали та прізвище)

### Нормоконтроль пройдено

Нормоконтролер \_\_\_\_\_  
(підпис)

А.В. Столярова  
\_\_\_\_\_  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка інтернет магазину електроніки»: 47 с., 44 рис., 5 табл., 8 джерел, 1 додаток.

БД, ER, LARAVEL, MYPHPADMIN, NODE.JS, PHP, SQL, CSS.

Об'єкт дослідження – інтернет магазин електроніки.

Мета роботи: розробка інтернет – магазину дієт з використанням Laravel та Node.js.

Метод дослідження – при розробці інформаційної системи було проведено аналіз предметної області, обрана архітектура і платформа реалізації, розроблена функціональна модель системи у нотаціях IDEF0, IDEF1, IDEF4, UML, Класова та Компонентна діаграми. Були використані phpmyadmin та SQL менеджер. Середовищем розробки був phpStorm. Здійснене тестування системи та оформлена супровідна документація. Були використані HTML для структурування контенту та CSS для стилізації.

## SUMMARY

Bachelor's qualifying paper «Development of The Electronics Online Store»:  
47 pages, 44 figures, 5 tables, 8 references, 1 supplement.

BD, ER, LARAVEL, MYPHPADMIN, NODE.JS, PHP, SQL, CSS.

The object of research is an online electronics store.

The purpose of the work: development of an online diet store using Laravel and Node.js.

During the development of the information system, an analysis of the subject area was carried out, the architecture and implementation platform were selected, a functional model of the system was developed in the IDEF0, IDEF1, IDEF4, UML, Class and Component diagram notations. phpmyadmin and SQL manager were used. The development environment was phpStorm. The system was tested and the accompanying documentation was prepared. HTML was used for structuring the content and CSS for styling.

## ЗМІСТ

|   |    |
|---|----|
| Завдання на кваліфікаційну роботу.....                              | 2  |
| Реферат .....   | 4  |
| Summary .....   | 5  |
| Вступ.....  | 7  |
| 1 Аналіз .....  | 9  |
| 1.1 Характеристика і функціонал інтернет магазину електроніки ..... | 9  |
| 1.2 Вимоги.....   | 9  |
| 2 Моделювання.....  | 11 |
| 2.1 Дослідження предметної області.....                             | 11 |
| 2.1.1 Початок проєктування.....                                     | 13 |
| 2.1.2 Проєктування інформаційної моделі .....                       | 15 |
| 2.1.3 Проєктування інформаційної моделі .....                       | 18 |
| 2.1.4 Діаграма прецедентів.....                                     | 22 |
| 2.1.5 Діаграма взаємодії та послідовності .....                     | 24 |
| 2.1.6 Діаграма розгортання та компонентів.....                      | 25 |
| 2.2 Висновки до розділу .....                                       | 26 |
| 3 Програмування .....   | 27 |
| 3.1 Створюємо проєкт.....   | 27 |
| 3.2 Інтерфейс.....  | 36 |
| Висновки .....  | 43 |
| Перелік посилань.....   | 44 |
| Додаток А Controller та маршрути сторінок у файлі web.php.....      | 45 |

## ВСТУП

Так вийшло, що сьогодні ми навіть не можемо себе уявити без того, щоб подивитися зранку новини у телефоні, подивитись щось цікаве у YouTube за сніданком. І хоч є люди, яким не подобається те, скільки часу люди почали проводити у Інтернеті, але навіть 15 – 20 років тому ми не могли навіть уявити, як же сильно він зробить наше життя легшим. Ми проводимо у Інтернеті більшу частину свого життя.

Сьогодні, навіть якщо тобі здається, що в тебе постійно нічого не виходить, можна не виходячи з дому, витрачаючи по одній – дві години на день, ти зможеш навчитися робити сайти. І коли не знаєш чим почати займатись у житті, почти зі свого маленького інтернет магазину. По перше це підніме твої навички у програмуванні і зможеш заробляти у фрілансі чи влаштуватися в якусь фірму. По друге якщо в тебе вже є якийсь бізнес, то так ти зможеш розширити охоплення аудиторії та покупців.

Важливо також і якість написаного коду, бо чим швидше сайт буде працювати, тим вище він у пошуку.

Електроніка це одне з того, що потребує просування у інтернеті, тому що завжди знайдеш свого покупця через великий асортимент та великий охоплення покупців. Ти навіть можеш не виходити нікуди, та шукати покупців на свій сайт, а перенаправляти заявки до постачальників.

Мета даної роботи є створення сайту для інтернет – магазину електроніки, з зручним зв'язком між покупцями та продавцями та зручністю у використанні.

Основні завдання:

- розкрити теоретичні основи фреймворку Laravel;
- проаналізувати діяльність магазину електроніки;
- побудувати моделі інформаційної системи;
- спроектувати інформаційну систему;
- реалізувати та протестувати інформаційну систему;

– оформити документацію до проєкту.

Об'єкт дослідження це інтернет магазин електроніки.

Предмет дослідження: є створення системи, яка забезпечить необхідне користувачам якісну та зручну роботу з пошуком товару та замовленням.

Методи дослідження це збір вимог до ПЗ, метод моделювання, метод проєктування, аналітичний метод та програмний.

Структурно робота складається зі вступу, трьох розділів, висновку, переліку посилань.

Перший розділ це аналіз технічного завдання, вимог до системи, розбір Laravel та опис системи [3].

У другому розділі проєктуються моделювання програмного забезпечення.

Третій розділ це реалізація програмування інформаційної системи та опис роботи інтернет магазину електроніки.



# 1 АНАЛІЗ

## 1.1 Характеристика і функціонал інтернет магазину електроніки

Інтернет-магазин електроніки подібний до інших онлайн-магазинів, які продають одяг, квіти або запчастини для автомобілів [5]. Вони всі мають спільні особливості, такі як урахування додаткових витрат, наприклад, на доставку і обробку замовлень, а також уважність до вимог покупця щодо характеристик товару. У такому магазині електроніки може бути широкий асортимент товарів, включаючи як велику побутову техніку для дому, так і аксесуари для смартфонів. Користувачі шукають конкретний товар і потрібні характеристики, а в разі потреби їм допомагають менеджери. При створенні такого інтернет-магазину використовуються клієнтська та адміністративна панелі.

## 1.2 Вимоги

Вимоги будуть складатися з декількох типів. Як мінімум розділимо на клієнтську і адмін частини. У них різні: функціонал, призначення, дизайн та використання.

Що вимагає адмін панель. Функціонал та зручність використання. Дизайн не так важливий, як для клієнтської частини.

- головне можливість користувачу, чиї дані є в БД, авторизуватися;
- редагувати замовлення;
- швидко відповісти клієнтові та допомогти йому;
- змінювати замовлення та відповідати на відгуки.

У всіх працівників своя «зона відповідальності». Та коли є нові замовлення, то вони йдуть до найменш зайнятого менеджера.

Менеджер вибирає стан замовлення із прийнято, на опрацьовано та

виконано. Адмін – панель повинна бути досить проста у використанні. Менеджер повинен швидко орієнтуватися, та швидко приймати замовлення або відповісти клієнту та допомогти обрати товар.

Відправлення замовлень:

Клієнт обирає потрібний йому товар по опису та характеристикам. Обирає товар та додає його до корзини. Вводить свої данні для відправки замовлення та сплачує товар. Отримує підтвердження на пошту.

Що вимагає клієнтська частина сайту:

- перегляд товарів;
- зручний пошук товарів за категоріями (у каталозі повинні бути всі описи товарів та їх категорії);
- відгуки (інші клієнти сайту повинні мати змогу прочитати реальний відгук про товар та обрати найкраще);
- робити замовлення (потрібно зробити форму де клієнт введе свої дані для замовлення та побажання до товару);
- оплата (потрібно підключити сервіси оплати товару).

## 2 МОДЕЛЮВАННЯ

### 2.1 Дослідження предметної області

Предметна область проекту є товар на сайті, який обирає клієнт на основі своїх потреб. Менеджер завжди зможе підказати чи допомогти з вибором правильних характеристик того чи іншого товару, що потребує клієнт.

Це будь що. Велика техніка для дому, маленькі аксесуари, запчастини до машини, телевізори, ноутбуки та інше. Клієнт лише робе замовлення, залишає свій номер телефону та електронну адресу для зв'язку. Кожне замовлення продивляється дуже уважно.

Для кожного товару є свої характеристики:

- назва товару: ідентифікує конкретний продукт, який продається;
- ціна товару: вказує вартість товару для покупця;
- опис: надає детальну інформацію про товар, його особливості, функції та характеристики;
- короткий опис: надає коротку інформацію про товар, яка допомагає швидко ознайомитись з основними відомостями про нього;
- картинки: зображення товару, що допомагають візуально оцінити його зовнішній вигляд та деталі;
- категорія: визначає класифікацію товару, розташовує його в певній групі або категорії, щоб полегшити навігацію покупця;
- рейтинг (з можливістю залишення відгуку): покупці можуть оцінювати товар і залишати свої відгуки, що допомагає іншим користувачам прийняти рішення про покупку;
- теги: ключові слова або фрази, що асоціюються з товаром і допомагають в редагуванні пошуку, спрощують пошукову навігацію покупця за інтересами чи потребами.

Ці елементи інформації допомагають покупцям знайти потрібний товар,

ознайомитись з його характеристиками, відгуками та зробити інформований вибір при покупці.

У замовленнях інтернет – магазину електроніки зазвичай включаються наступні компоненти:

- дані клієнта: інформація, яка стосується покупця, включаючи його особисті дані, такі як ім'я, адреса та контактна інформація;
- ціна замовлення: вартість товарів або послуг, які були обрані клієнтом;
- статус замовлення: показує етап обробки замовлення, який може змінюватися менеджером (наприклад, "В очікуванні оплати", "У процесі доставки", "Виконано" тощо);
- відповідь адміністратора: коментарі або відповіді від адміністратора магазину, які пов'язані з конкретним замовленням;
- обраний товар: конкретний товар або послуга, який був обраний клієнтом для замовлення;
- кожен клієнт має свого власного менеджера, який відповідає за його обслуговування.

Крім того, кожен товар може мати відгуки, які включають:

- назва відгука: заголовок або короткий опис відгуку;
- оцінка продукту: оцінка, яка відображає задоволення покупця товаром на певну шкалу (наприклад, від 1 до 5 зірок);
- текст відгуку: детальний коментар або опис вражень покупця щодо товару;
- ім'я користувача: ім'я або псевдонім особи, яка залишила відгук;
- товар, на який дається відгук: ідентифікатор або назва товару, до якого стосується відгук;
- час створення відгуку: дата та час, коли відгук був залишений.

Менеджери магазину мають свої особисті дані, такі як ім'я, телефон, електронна пошта, посада. Вони також мають паролі, токени та інформацію про час створення та зміни облікового запису, а також про поточну кількість замовлень, якими вони керують.

На сайті такого магазину важливо мати наступний функціонал:

- фільтрація: можливість фільтрувати товари за різними критеріями, такими як ціна, категорія, характеристики тощо;
- купівля: можливість клієнтам придбати обрані товари та оформити замовлення;
- можливість залишати відгуки: можливість покупцям залишати свої враження та оцінки щодо придбаних товарів.

Функціонал панелі адміністратора включає:

- можливість редагування замовлень: зміна статусу замовлень, оновлення інформації про замовлення, комунікація з клієнтами тощо;
- відправка замовлення на електронну пошту користувача: можливість адміністратору надсилати повідомлення про статус замовлення або іншу важливу інформацію на електронну пошту клієнта.

### **2.1.1 Початок проєктування**

Перед тим, як почати програмувати, потрібно підійти до такої ж важливої частини. До проєктування. Таким чином, ти подивишся на свій проєкт збоку та зрозумієш, що треба переробити. Спочатку робимо повний опис проєкту і взаємодії. Потім декомпозиція. Ділимо все на підсистеми і кожен описуємо.

Спочатку іде функціональне проєктування. Це методологія IDEF0 (див. рис. 2.1). Це побудова ієрархічної системи діаграм. Окрема діаграма і буде описом частинок.

Потрібно зрозуміти для діаграми IDEF0, що є на початку і що виходить в результаті.

Спочатку є лише потреба клієнта у чомусь. Його особисті побажання, гроші.

Замовлення обробляє менеджер. Для цього йому потрібні або інструкції, або добре вивчити опис товарів. Не забуваємо і про законі України, та права

покупців. Кінцева точка це сформоване замовлення та можливі поради менеджера.

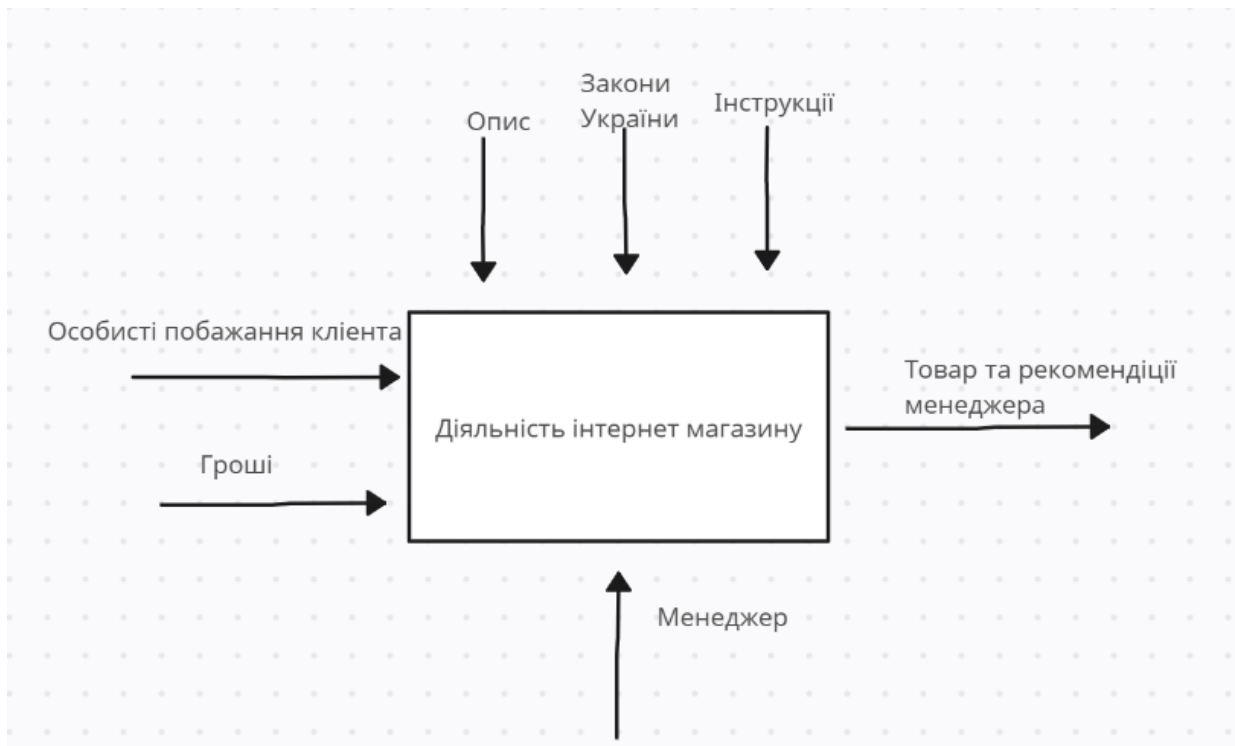


Рисунок 2.1 – Контекстна діаграма IDEF0

Для функціонального проектування даного проекту можна виділити кілька підсистем, які допоможуть розбити його на складові частини.

**Підсистема вибору товару:** ця підсистема дозволяє клієнту обирати товари за їхніми побажаннями. Вона має включати функціонал для пошуку, фільтрації та перегляду товарів з урахуванням характеристик, категорій та інших параметрів.

**Підсистема оплати і купівлі товару:** ця підсистема забезпечує процес оплати і оформлення замовлення. Вона повинна мати інтеграцію з платіжними системами для прийому платежів, а також функціонал для заповнення форми замовлення та обробки платежу.

**Підсистема формування замовлень:** ця підсистема відповідає за збір і обробку інформації про замовлення. Вона включає функціонал для зберігання даних клієнтів, обробки вибраних товарів, розрахунку загальної вартості

замовлення та формування відповідного запису замовлення.

**Підсистема сповіщення клієнта:** ця підсистема відповідає за надсилання повідомлення клієнту щодо підтвердження замовлення та інших важливих оновлень. Вона може використовувати електронну пошту, SMS або інші канали зв'язку для доставки повідомлень клієнту.

Кожна з цих підсистем має свої функціональні можливості та взаємодії з іншими частинами системи, але разом вони утворюють повну функціональність інтернет – магазину електроніки.

Нижче наведено діаграму декомпозиції інтернет – магазину електроніки (див. рис. 2.2).

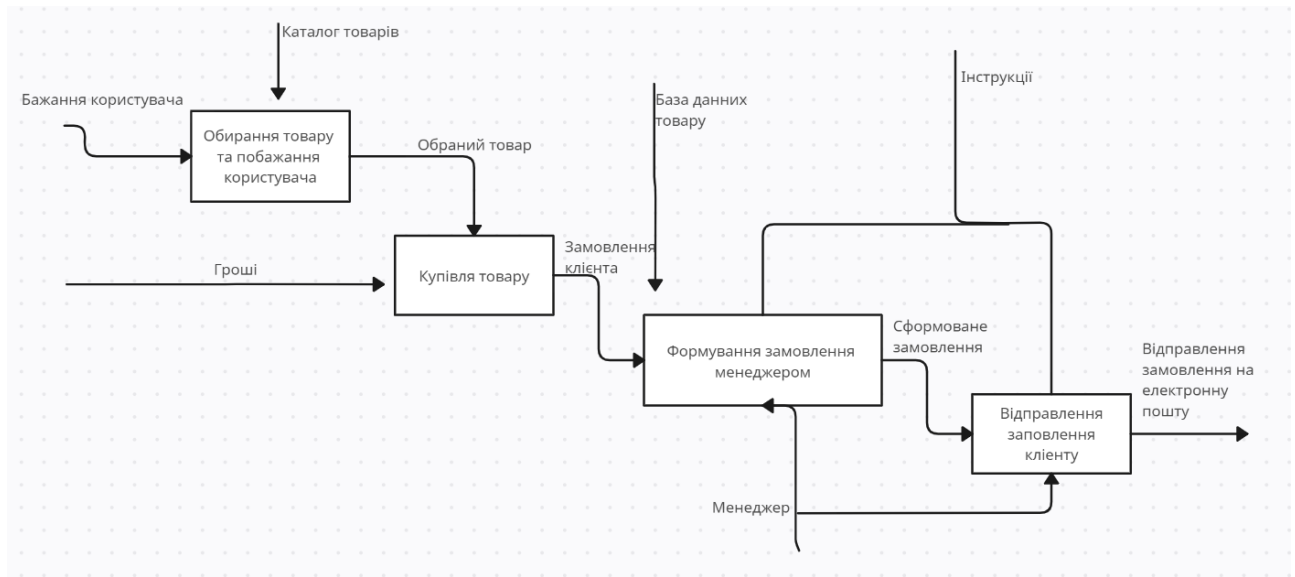


Рисунок 2.2 – Діаграма декомпозиції інтернет – магазину електроніки

### 2.1.2 Проектування інформаційної моделі

Наведені дані можна візуалізувати за допомогою діаграми типу ER (сутність – зв'язок). Ця модель даних надає загальний огляд взаємозв'язків між сутностями без надання подробиць. Це дозволяє швидко оцінити масштаб проекту та його сутність.

Варто зауважити, що ER-діаграми зазвичай використовуються для

реляційних баз даних, таких як MySQL. Ця методологія передбачає чітку структурування даних і розбиття їх на поля, стовпці та рядки.

Опишемо концептуальну ER-діаграму (див. рис. 2.3).

В даному випадку, можна розпочати з виділення основних сутностей, починаючи з сутності "Продукт".

Це основа для магазину. Вона має:

- первинний ключ id;
- ім'я, опис;
- короткий опис;
- рейтинг;
- ціна та акційна ціна;
- певний перелік картинок та категорію.

Винесемо категорію товарів у окремо: первинний ключ id та назва категорії.

Для зберігання картинок, можна використати файловий менеджер, де при завантаженні зображення будуть зберігатися. У базі даних будуть зберігатися посилання на ці картинки, а також назви картинок, які будуть використовуватися для src-атрибуту зображень. Кожній картинці буде відповідати свій унікальний ідентифікатор (первинний ключ id).

"Order" – це сутність, яка використовується для формування та обробки замовлень користувачів. Вона містить різні атрибути, які дозволяють зберігати необхідну інформацію про замовлення. Наприклад, атрибути включають ім'я клієнта, контактну інформацію (телефон та електронну пошту), вартість замовлення. Крім того, є атрибути, що служать для відстеження змін та додавання замовлення, такі як час зміни та час додавання. Також сутність "Order" пов'язана з менеджером магазину.

Наступною сутністю, яку розглянемо, є "Status". Вона використовується для контролю та регулювання процесу обробки замовлень, а також для ефективного розподілу завдань між адміністраторами. Сутність "Status" містить два атрибути – первинний ключ id та назву статусу.



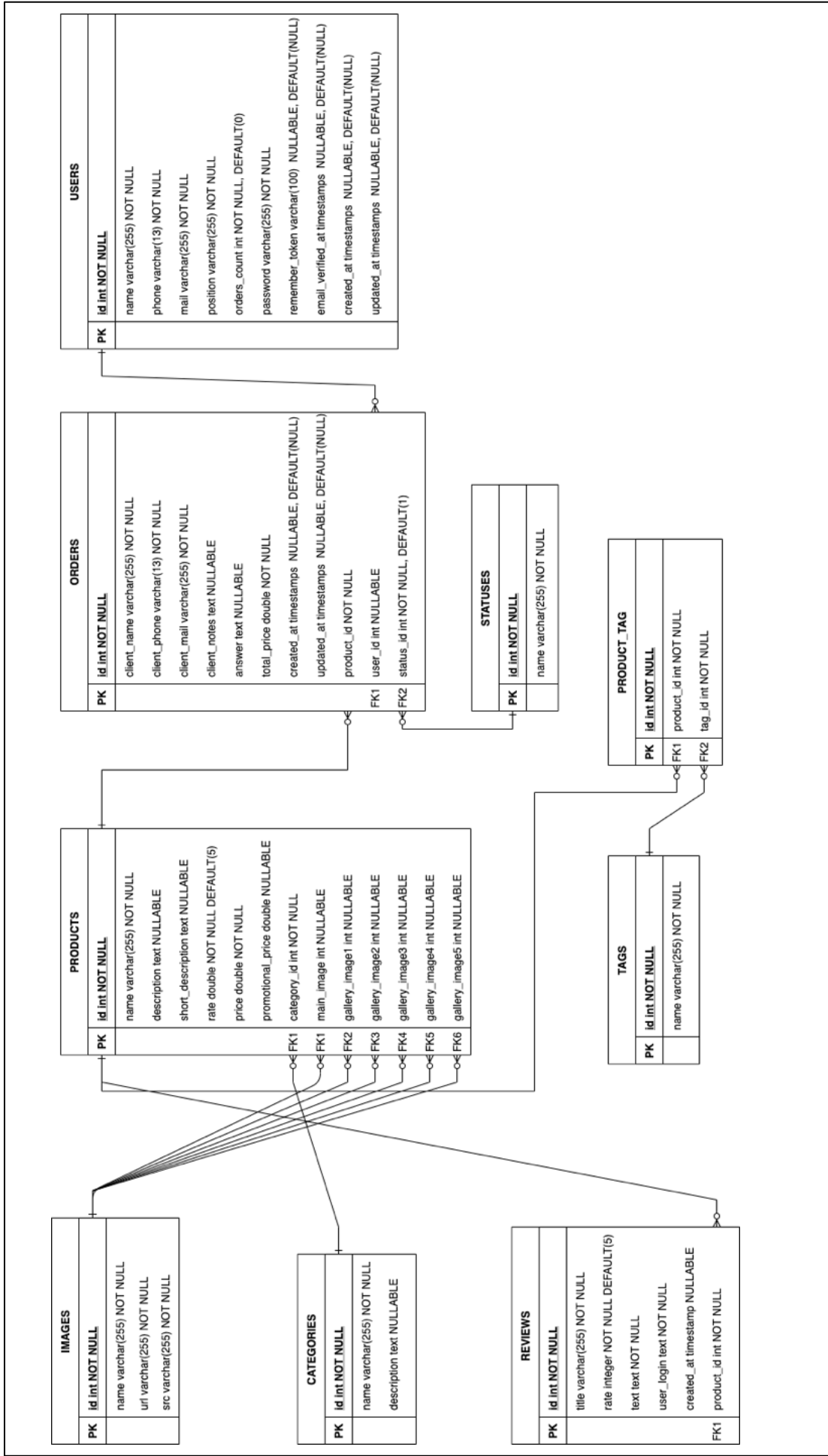


Рисунок 2.3 – ER – діаграма

Ці атрибути дозволяють ідентифікувати конкретний статус замовлення та надають інформацію про його поточний стан.

Остання сутність для розгляду – "User" або менеджер. Кожне замовлення обов'язково обробляється менеджером, тому ця сутність містить атрибути, що відображають особисту інформацію менеджера, такі як його ім'я, телефонний номер, електронна пошта та посада. Додатково, для автентифікації та безпеки, фреймворк Laravel автоматично створює певні атрибути:

- пароль;
- токен;
- час зміни облікового запису менеджера;
- час верифікації – важливий для токену;
- кількість замовлень.

### **2.1.3 Проектування інформаційної моделі**

Мета логічного проектування є створення детальної логічної моделі даних, яка відповідає предметній області і вимогам СУБД [1]. У цій моделі кожна таблиця відповідає сутностям ER – діаграми (див. рис. 2.3).

Таблиця "Product" містить інформацію про товари, які є важливими для інтернет-магазину. Ці товари відносяться до категорії електроніки. Перше це первинний ключ (див. табл. 2.1).

Атрибути:

- первинний ключ;
- "name";
- "description";
- "short\_description"
- "rate";
- "price";
- "promotional\_price".

Крім цих атрибутів, таблиця також містить зовнішні ключі:

- "category\_id";
- "main\_image";

Таблиця 2.1 – Структура таблиці «Products»

| Поле              | Тип даних               | Розмір поля | Властивості        |
|-------------------|-------------------------|-------------|--------------------|
| id                | Числовий                |             | PK, Auto_increment |
| name              | Символьний              | 255         |                    |
| description       | Текстовий               |             | nullable           |
| short_description | Текстовий               |             | nullable           |
| rate              | Число з плаваючою комою |             | default(5)         |
| price             | Число з плаваючою комою |             |                    |
| promotional_price | Число з плаваючою комою |             | nullable           |
| category_id       | Числовий                |             | FK                 |
| main_image        | Числовий                |             | FK                 |
| gallery_image1    | Числовий                |             | FK                 |
| gallery_image2    | Числовий                |             | FK                 |
| gallery_image3    | Числовий                |             | FK                 |
| gallery_image4    | Числовий                |             | FK                 |
| gallery_image5    | Числовий                |             | FK                 |

Таблиця «User» визначає поля в менеджера, який відповідає за на відповіді на замовлення та написання їх (див. табл. 2.2).

Атрибути:

- name;
- phone;
- mail;
- електронної пошти менеджера;
- position;
- order\_count;
- remember\_token;
- verified\_at);
- updated\_at;
- created\_at.

Таблиця 2.2 – Структура таблиці «User»

| Поле           | Тип даних       | Розмір поля | Властивості              |
|----------------|-----------------|-------------|--------------------------|
| id             | Числовий        |             | PK, Auto_increment       |
| name           | Символьний      | 255         |                          |
| phone          | Символьний      | 13          |                          |
| mail           | Символьний      | 255         |                          |
| position       | Символьний      | 255         |                          |
| order_count    | Числовий        |             | default(0)               |
| password       | Символьний      | 255         |                          |
| token_remember | Символьний      | 100         | NULLABLE, default (NULL) |
| verified_at    | Тимчасові мітки |             | NULLABLE, default (NULL) |

Продовження табл. 2.2

| Поле       | Тип даних       | Розмір поля | Властивості              |
|------------|-----------------|-------------|--------------------------|
| created_at | Тимчасові мітки |             | NULLABLE, default (NULL) |
| updated_at | Тимчасові мітки |             | NULLABLE, default (NULL) |

Таблиця «Orders» – таблиця замовлень (див. табл. 2.3).

Атрибути:

- первинний ключ;
- client\_name;
- client\_phone;
- client\_mail;
- client\_notes;
- answe;
- total\_price;
- created\_at;
- updated\_at.

Два зовнішні ключі:

- status\_id;
- user\_id.

Таблиця 2.3 – Структура таблиці «Orders»

| Назва поля | Тип даних | Розмір поля | Властивості | Зв'язок |
|------------|-----------|-------------|-------------|---------|
| id         | Числовий  |             | PK,         |         |

Продовження табл. 2.3

| Назва поля   | Тип даних               | Розмір поля | Властивості             | Зв'язок          |
|--------------|-------------------------|-------------|-------------------------|------------------|
|              |                         |             | Auto_increment          |                  |
| client_name  | Символьний              | 255         |                         |                  |
| client_phone | Символьний              | 13          |                         |                  |
| client_mail  | Символьний              | 255         |                         |                  |
| client_notes | Текстовий               |             | nullable                |                  |
| answer       | Текстовий               |             | nullable                |                  |
| total_price  | Число з плаваючою комою |             |                         |                  |
| user_id      | Числовий                |             | FK, nullable            | Від User (id)    |
| status_id    | Числовий                |             | FK, default(1)          | Від Status (id)  |
| product_id   | Числовий                |             | FK                      | Від Product (id) |
| created_at   | Тимчасові мітки         |             | Nullable, default(NULL) |                  |
| updated_at   | Тимчасові мітки         |             | Nullable, default(NULL) |                  |

#### 2.1.4 Діаграма прецедентів

На рисунку 2.4 представлено діаграму прецедентів. Розберемо акторів (див. табл. 2.4).

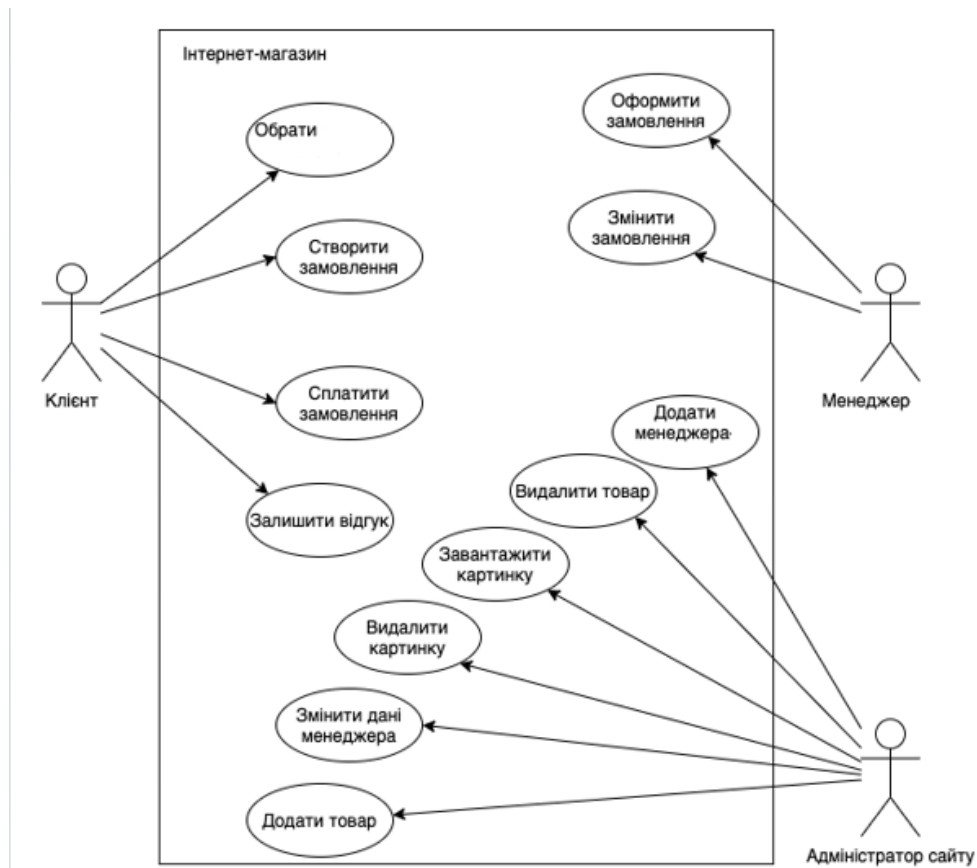


Рисунок 2.4 – Діаграма прецедентів

Таблиця 2.4 – Актори

| Назва актора        | Опис актора               |
|---------------------|---------------------------|
| Клієнт              | Користується сайтом       |
| Менеджер            | Обробляє заявки           |
| Адміністратор сайту | Має права «адміністратор» |

Розберемо можливості акторів (див. табл. 2.5).

Таблиця 2.5 – Можливості акторів

| Назва               | Опис                                |
|---------------------|-------------------------------------|
| Обрати              | Функція щоб клієнт міг обрати товар |
| Створити замовлення | Можливість створення замовлення     |
| Сплатити замовлення | Функція оплати                      |
| Залишити відгук     | Можливість поділитися відгуком з    |

## Продовження табл. 2.5

| Назва                  | Опис   |
|------------------------|--|
|                        | іншими клієнтами                                       |
| Оформити замовлення    | Функція менеджера прийняти замовлення                  |
| Змінити замовлення     | Можливість менеджера редагувати замовлення             |
| Додати менеджера       | Функція адміністратора додавати чи видаляти менеджерів |
| Видалити товар         | Можливість редагування товару                          |
| Завантажити картинку   | Можливість змінювати данні товару                      |
| Видалити картинку      | Можливість змінювати данні товару                      |
| Змінити дані менеджера | Функція редагування профілю менеджера                  |
| Додати товар           | Функція редагування товару                             |

### 2.1.5 Діаграма взаємодії та послідовності

Діаграма послідовності – це графічний інструмент, який дозволяє моделювати взаємодію між об'єктами або компонентами системи впродовж певного часового періоду. Дозволяє зобразити порядок виконання операцій та комунікацію між об'єктами системи. Вона допомагає розібратися в послідовності подій і взаємодії між об'єктами під час виконання певного процесу. Загалом, діаграма послідовності допомагає уявити та зрозуміти, як система працює у реальному часі, відображаючи взаємодію об'єктів та послідовність подій. Вона є потужним інструментом для аналізу та проектування систем, а також для комунікації між розробниками та зацікавленими сторонами [2].

Розглянемо діаграму активностей (див. рис. 2.5).



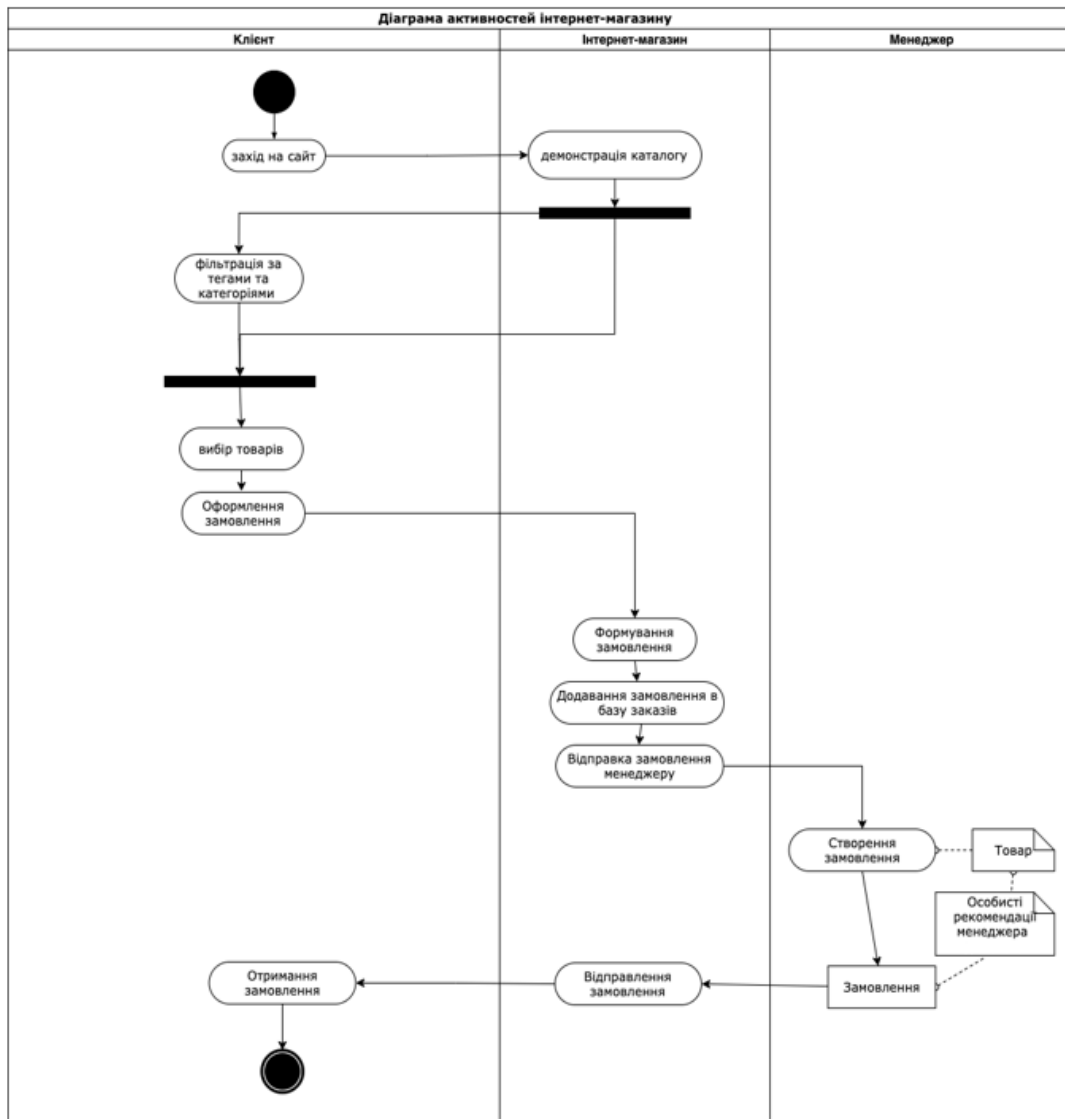


Рисунок 2.5 – Діаграма взаємодії інтернет-магазину

### 2.1.6 Діаграма розгорткування та компонентів

Ділимо на частини:

- а) база даних (сутності на ER-діаграмі (див. рис. 2.2));
- б) front – end:
  - 1) «About Us» та «Contact Us»;
  - 2) панель адміністратора.

Тож розглянемо діаграму розгорткування (див. рис. 2.6).

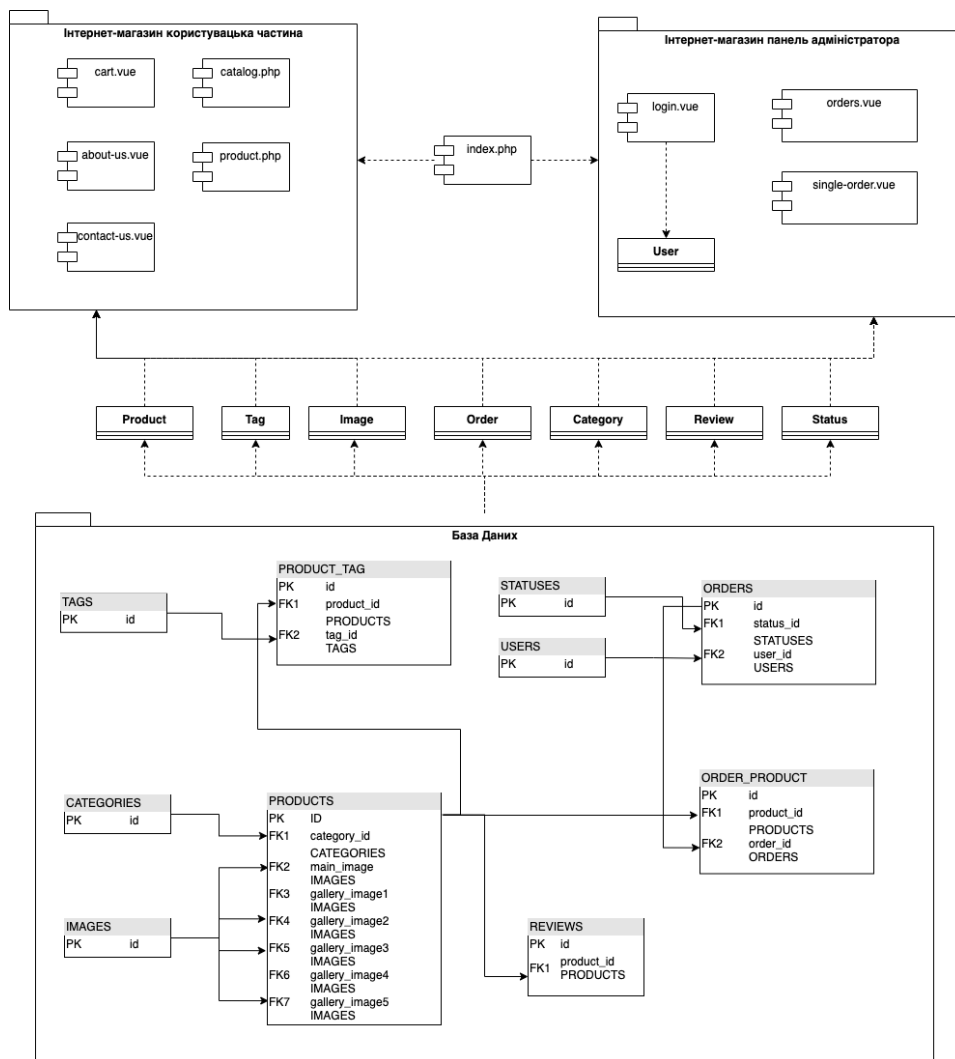


Рисунок 2.6 – Діаграма розгортання

## 2.2 Висновки до розділу

Проектування програмного забезпечення є важливим етапом розробки, який допомагає визначити основні аспекти проекту і підготувати його до реалізації. У даному проекті, за допомогою діаграм, вдалося отримати уявлення про його складність та необхідність функціоналу. Виявилось, що проект має середню складність і включає в себе вісім сутностей. Адміністративна панель, яка має більший функціонал і потребує авторизації, виявилась складнішою за клієнтську частину. Завдяки проектуванню програмного забезпечення, розробники отримали більш чітке уявлення про проект, його складність та обсяг робіт, що сприяє більш точному плануванню і успішному виконанню проекту.

## 3 ПРОГРАМУВАННЯ

### 3.1 Створюємо проєкт

Для початку роботи встановлюємо OpenServerPanel. Це графічний інтерфейс, який надає зручний спосіб керування сервером без необхідності вручного введення конфігураційних файлів або використання командного рядка. За допомогою OpenServerPanel можна швидко встановити та налаштувати локальне середовище розробки для вебпроєктів [8].

Відкриваємо консоль (див. рис. 3.1).

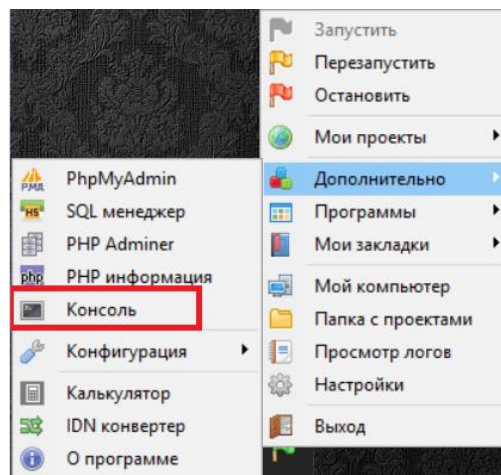


Рисунок 3.1 – Консоль OpenServe

У консолі нам потрібно відкрити папку «domains» за допомогою команди «cd D:/OpenServe/domains» та створюємо проєкт, за допомогою команди (див. рис. 3.2). Замість «example - app» – пишемо назву нашого нового проєкту – «shop».

```
composer create-project laravel/laravel example-app
```

Рисунок 3.2 – Створити новий проєкт Laravel

Через OpenServe – мої проєкти заходимо в проєкт «shop» (див. рис. 3.3).

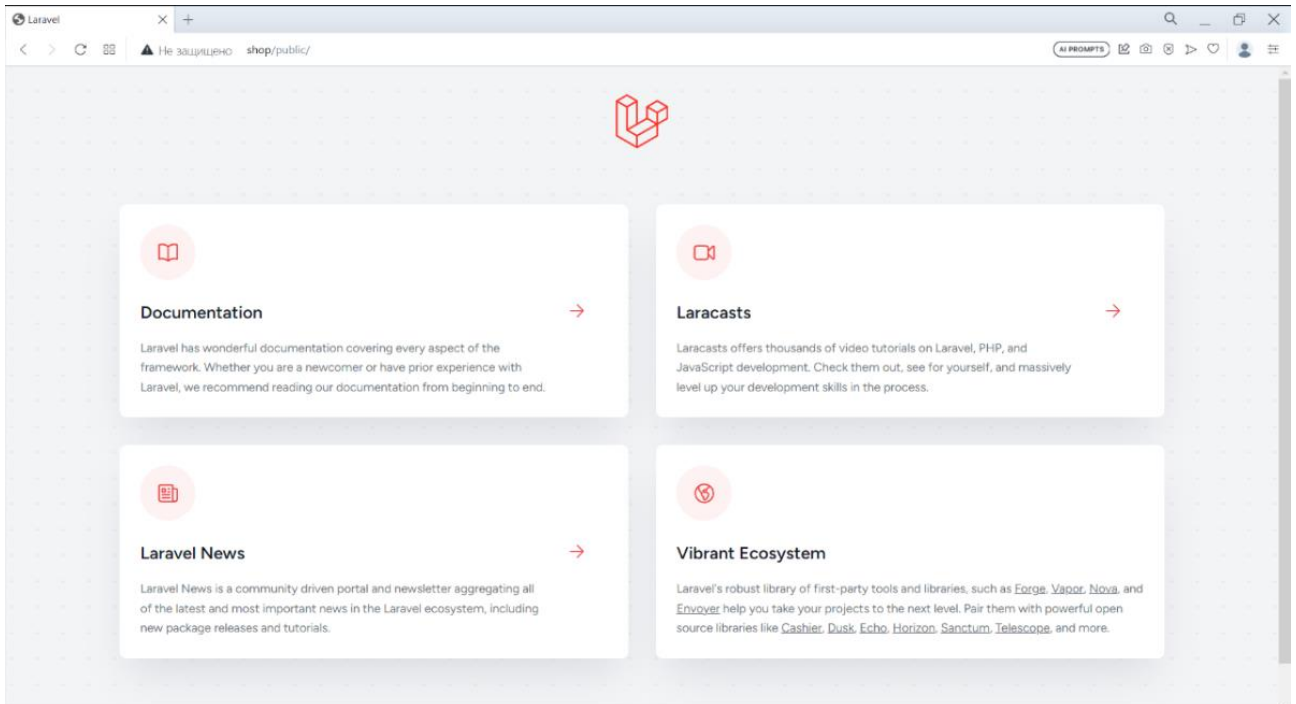


Рисунок 3.3 – Головна сторінка Laravel

Запускаємо локальний сервер розробки Laravel за допомогою команди Laravel Artisan CLI `serve`: (див. рис. 3.4).

```
php artisan serve
```

Рисунок 3.4 – Запуск локального серверу

Встановлюємо зручне IDE для PHP – phpStorm. Запускаємо наш проєкт «shop» (див. рис. 3.5).

Заходимо через OpenServe в SQL Менеджер та створюємо (див. рис. 3.6 і рис. 3.7).

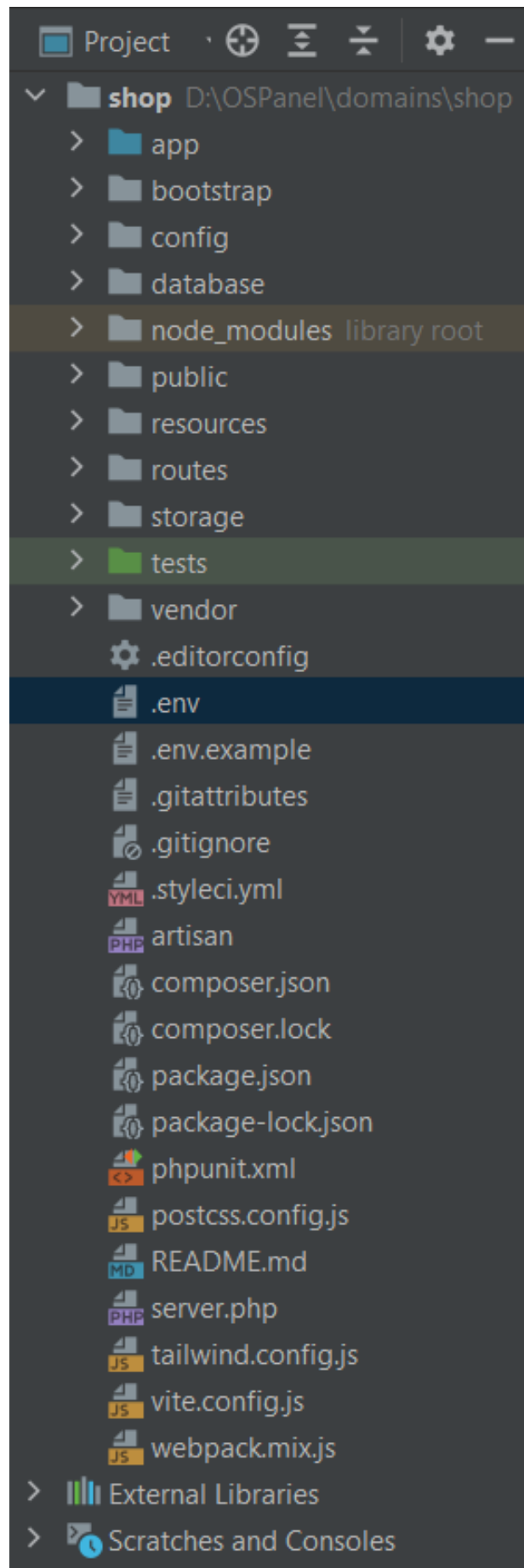


Рисунок 3.5 – Каталог проекта

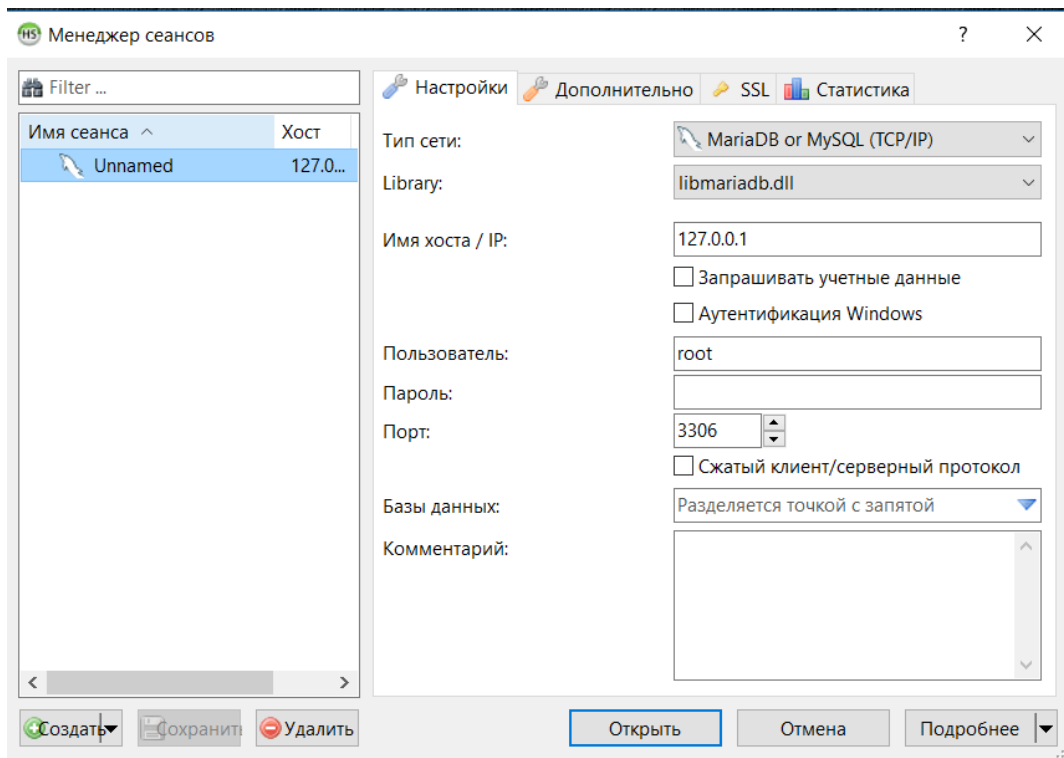


Рисунок 3.6 – SQL Менеджер

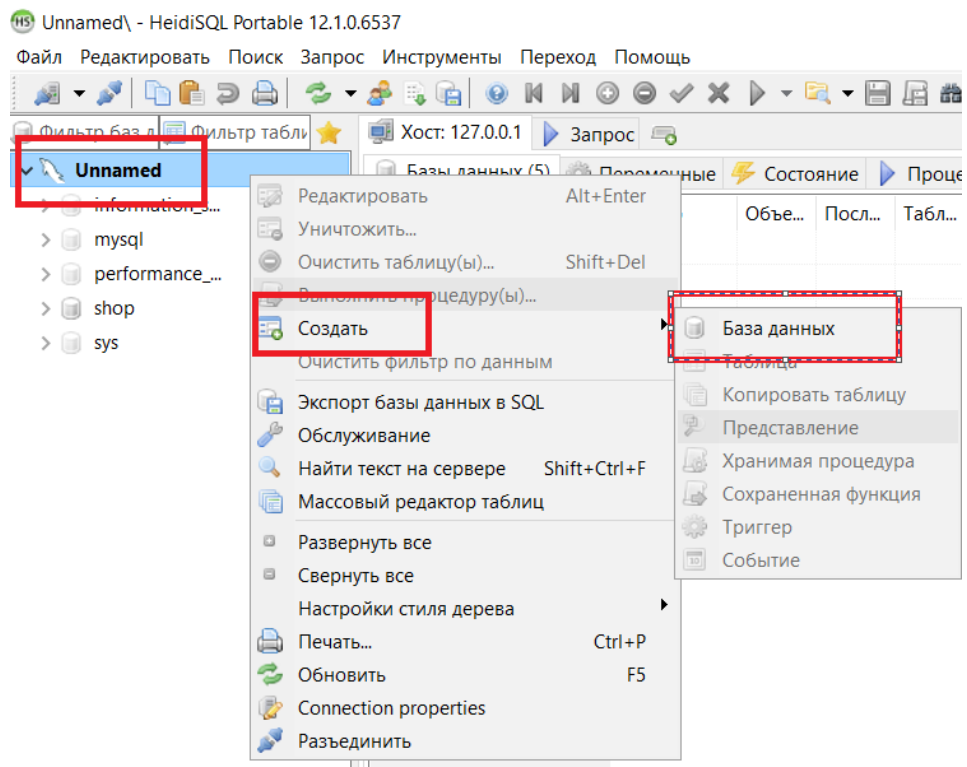


Рисунок 3.7 – Створюємо базу даних

Підключаємо базу даних, редагуємо файл .env (див. рис. 3.8).

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=shop
DB_USERNAME=root
DB_PASSWORD=
```

Рисунок 3.8 – Конфігурація бази даних

Після усіх приготувань, за допомогою команди «php artisan serve» можемо відкривати сайт по посиланню <http://127.0.0.1:8000/>.

Для того, щоб на нашому сайті з'явилась форма реєстрації, відкриваємо консоль OpenServe та вводимо команди (див. рис. 3.9).

|  |
|--|
| <code>composer require laravel/breeze --dev</code> |
| <code>php artisan breeze:install blade</code>      |
| <code>php artisan migrate</code>                   |

Рисунок 3.9 – Створення форми реєстрації

Для того щоб форма реєстрації виглядала красиво (рис. 3.10), потрібно підгрузити JavaScript код [7].

The image shows a registration form with a light gray background. At the top center is a logo consisting of three stacked cubes. Below the logo is a white rounded rectangle containing the form fields. The 'Email' field has a blue border and a cursor. The 'Password' field is empty. Below the password field is a checkbox labeled 'Remember me'. At the bottom right of the form area, there is a link 'Forgot your password?' and a dark gray button with white text 'LOG IN'.

Рисунок 3.10 – Форма реєстрації

Для цього встановлюємо Node.js та відкриваєм звичайний cmd і вводимо команди (див. рис. 3.11) [6].

|             |
|-------------|
| npm install |
| npm run dev |

Рисунок 3.11 – Загрузка JS

Одним із популярних способів підключення адмін-панелі в Laravel – це Voyager (див. рис. 3.12 і рис. 3.13) [4].

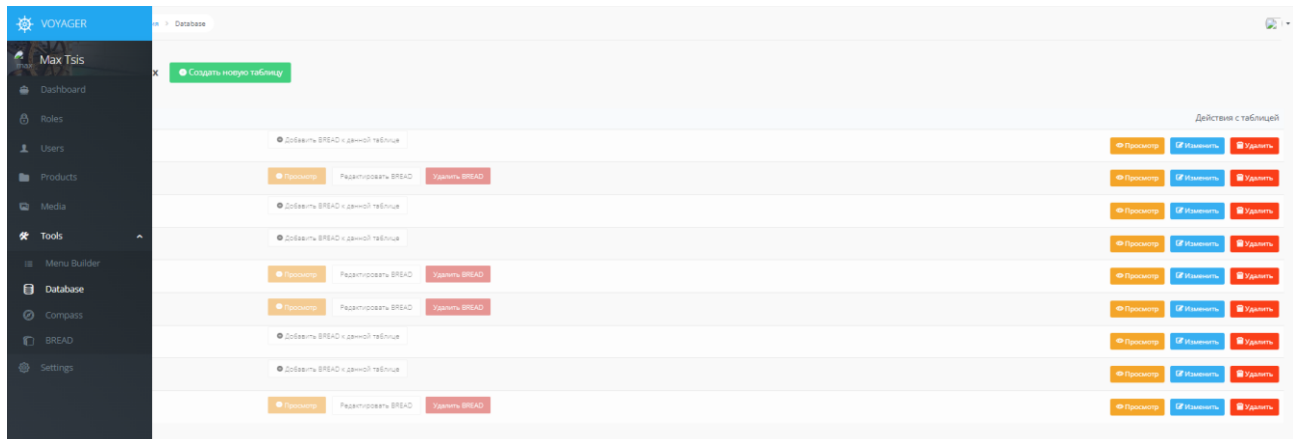


Рисунок 3.12 – Адмін панель voyager

```
Route::group(['prefix' => 'admin'], function () {
    Voyager::routes();
});
```

Рисунок 3.13 – Частина коду про Voyager у файлі web.php

Для цього відкриваємо в OpenServe консоль і вводимо команди (див. рис. 3.14).

|                              |
|------------------------------|
| composer require tcg/voyager |
| php artisan voyager:install  |

Рисунок 3.14 – Підключення Voyager.

Запустіть локальний сервер розробки за допомогою «php artisan serve» і відвідайте <http://localhost:8000/admin>. Для того, щоб нам стати адміном на сайті, потрібно авторизуватися на сайті. Відкрити консоль і ввести команду (див. рис. 3.15).



php artisan voyager:адміністратор your@gmail.com

Рисунок 3.15 – Реєстрація адміністратора в адмін панелі Voyager

Приклад, як можна зробити таблицю (див. рис. 3.16) та додати в неї товари (рис. 3.17) за допомогою Voyager.

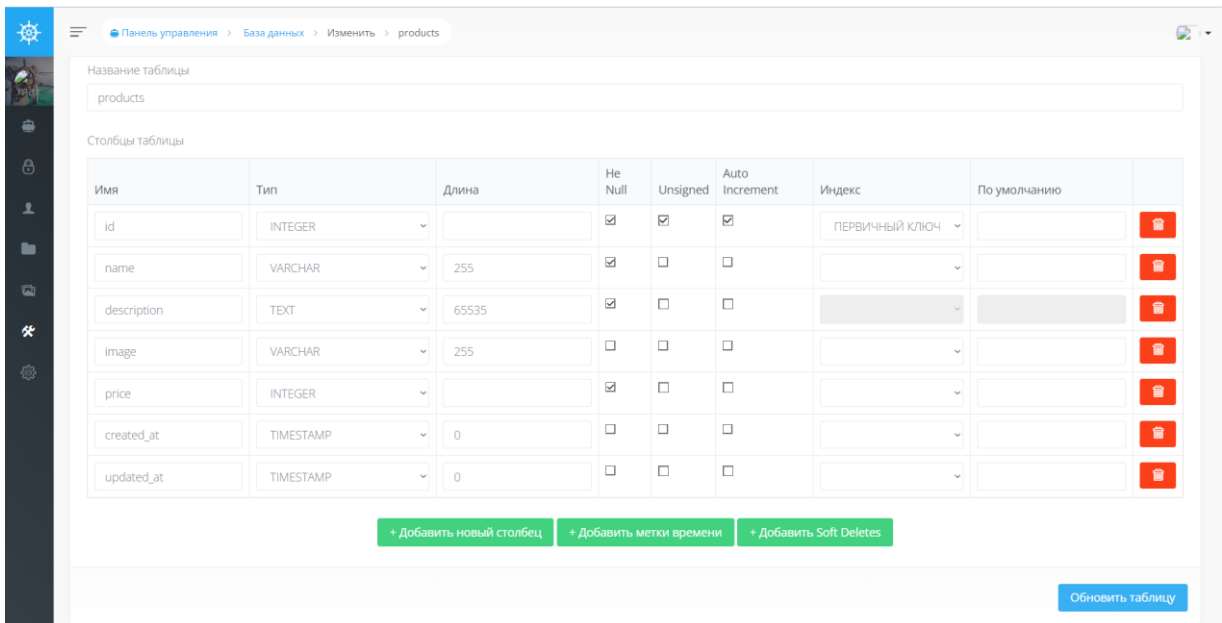


Рисунок 3.16 – Таблица за «products» допомогою адмін панелі voyager

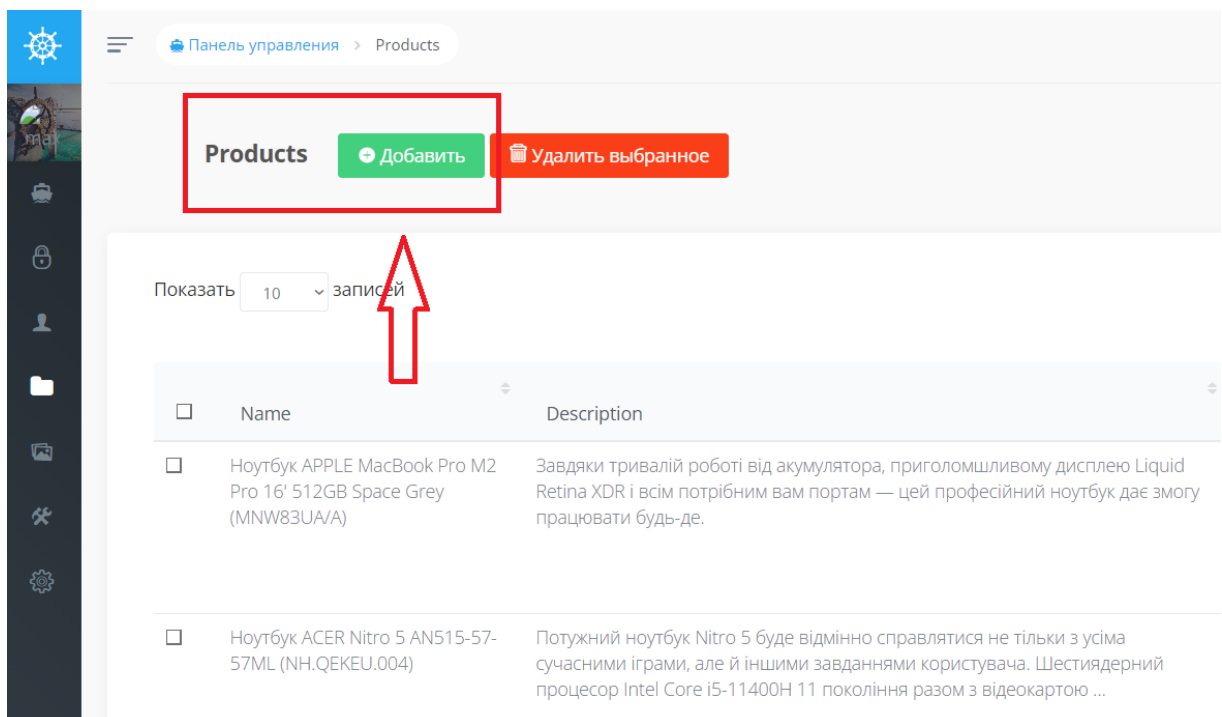


Рисунок 3.17 – Приклад роботи панелі управління в voyager

Для того, щоб ми могли взаємодіяти з нашою базою даних нам потрібно прописати Controller (див. рис. 3.18).

Для цього відкриваємо консоль в OpenServe та вводимо команду.

```
Php artisan make: controller назваController
```

Рисунок 3.18 – Команда для Controller

Приклади усіх прописаних Controller'ів та усі прописані маршрути сторінок у файлі web.php (див. додаток А).

Приклад роботи Controller. Частина коду, у яку можна «вкласти» стільки товару, скільки хочеш (рис. 3.19). Товар виводить на сторінку з лімітом 3 і пропускає першу строчку (рис. 3.21). Товар можна добавляти через адмін-панель voyager.

```
class ProductController extends Controller
{
    1 usage
    public function shopList()
    {
        $products = Product::query()->limit( value: 3)->offset( value: 1)->get();

        $sessionId = Session::getId();

        \Cart::session($sessionId);

        $cart = \Cart::getContent();

        $sum = \Cart::getTotal('price');

        return view( view: 'electronic-shop/shop-page', [
            'products' => $products,
            'cart' => $cart,
            'sum' => $sum,
        ]);
    }
}
```

Рисунок 3.19 – Controller у якому прописана функція списку товарів shopList

```

@foreach($products as $product)
<div class="product-width col-lg-6 col-xl-4 col-md-6 col-sm-6">
  <div class="product-wrapper mb-10">
    <div class="product-img">
      <a href="product-details.blade.php">
        
      </a>
    <div class="product-action">
      <a title="Quick View" data-toggle="modal"
        data-target="#exampleModal" href="#">
        <i class="ti-plus"></i>
      </a>
      <a title="Add To Cart" href="{{route('electronic-shop/add-cart', ['id' => $product->id])}}">
        <i class="ti-shopping-cart"></i>
      </a>
    </div>
    <div class="product-action-wishlist">
      <a title="Wishlist" href="#">
        <i class="ti-heart"></i>
      </a>
    </div>
  </div>
  <div class="product-content">
    <h4><a href="product-details.blade.php">{{$product->name}}</a></h4>
    <div class="product-price">
      <span>{{$product->price}}</span>
    </div>
  </div>
  <div class="product-list-content">

```

Рисунок 3.20 – Код блоку товарів

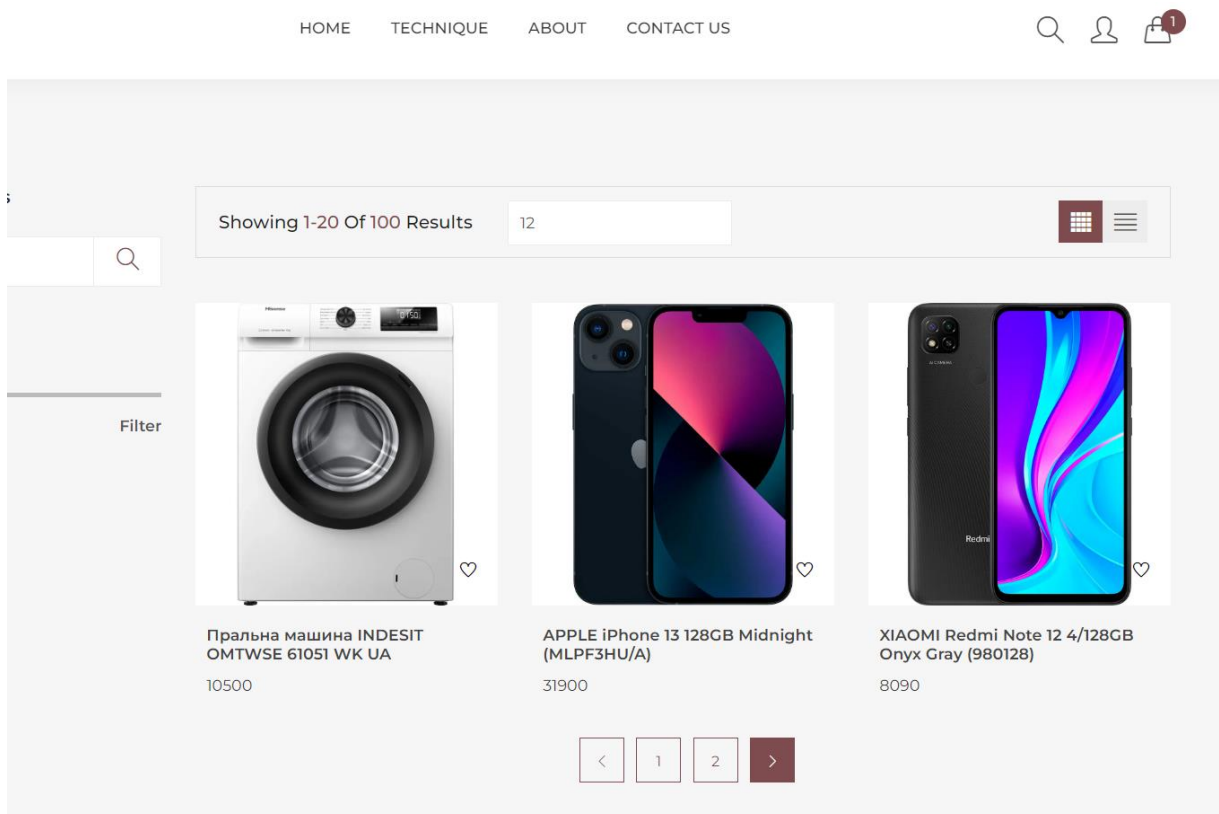


Рисунок 3.21 – Результат функції shopList у Controller

```

class Orders extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create( table: 'orders', function (Blueprint $table) {
            $table->integer( column: 'id')->index()->autoIncrement();
            $table->integer( column: 'user_id')->index();
            $table->longText( column: 'cart_data');
            $table->string( column: 'address');
            $table->float( column: 'total_sum');
            $table->string( column: 'phone');
            $table->timestamps();

            $table->primary( columns: 'id');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
}

```

Рисунок 3.22 – Міграція замовлень

## 3.2 Інтерфейс

Перший блок сайту це головна сторінка зі слайдером (див. рис. 3.23). Можна побачити, тематику сайту.

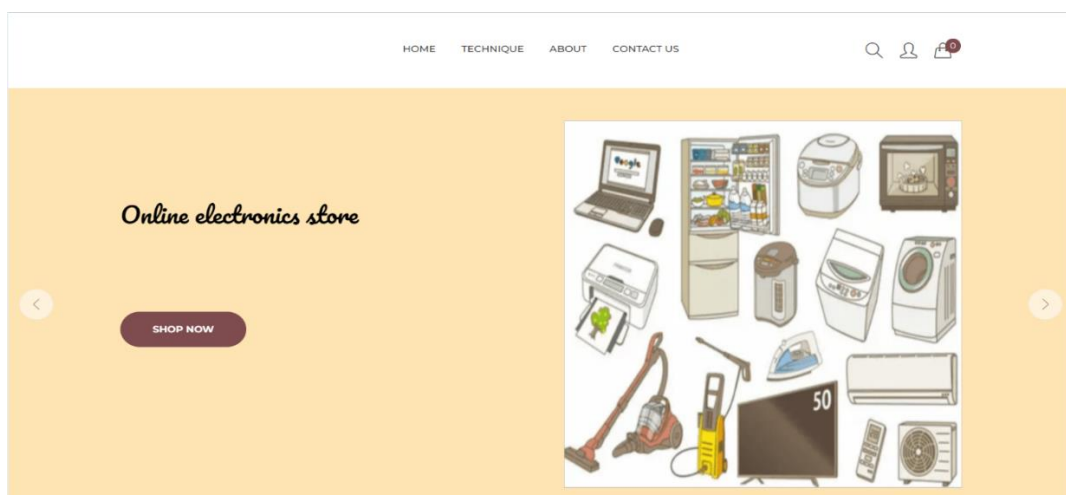
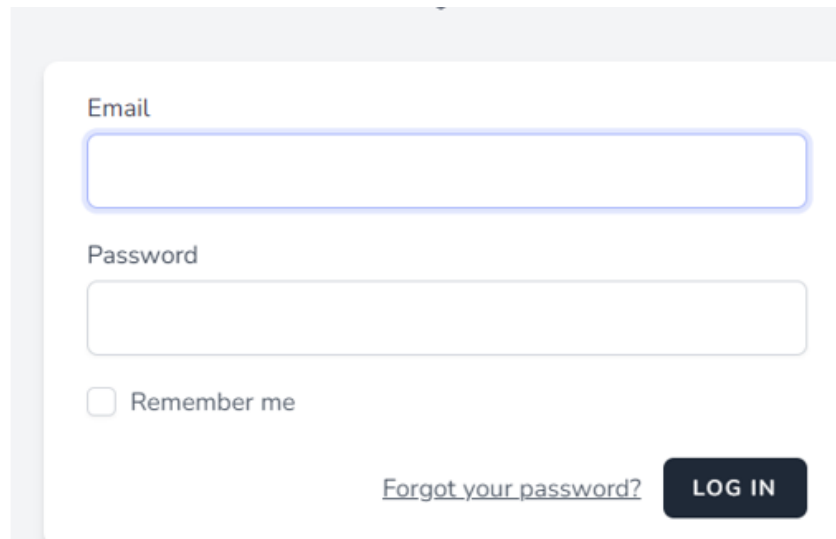


Рисунок 3.23 – Перший блок сайту зі слайдером

Для роботи с товаром, потрібно зареєструватися.



Email

Password

Remember me

[Forgot your password?](#) **LOG IN**

Рисунок 3.24 – Форма реєстрації

Другий блок сайту це частина голосної сторінки, де наведеного найпопулярніший товар. При оновленні сторінки товар міняється (див. рис. 3.25).

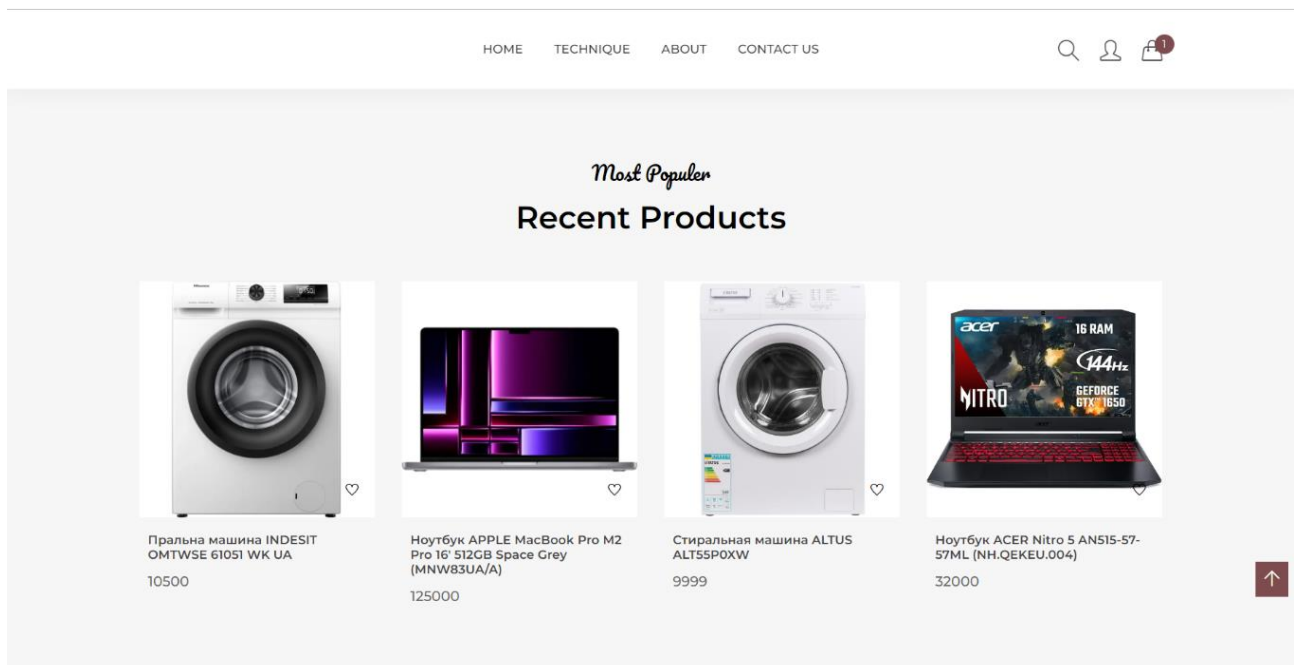


Рисунок 3.25 – Другий блок сайту


Третій блок сайту показує нам угоду тижня (див. рис. 3.26). Можна одразу прочитати характеристики товару, не переходячи на сторінку з товаром.

HOME    TECHNIQUE    ABOUT    CONTACT US

🔍 👤 🛒

*Best Product*

## Deal of the Week



Пральна машина INDESIT OMTWSE 61051 WK UA 10500

Лінійка пральних машин MyTime створена для прискореного прання одягу. Тепер в вашому розпорядженні буде декілька швидких циклів тривалістю менше години. Серед них: Бавовна 30 °C/30 хв., Синтетика (59 хв.), Шовк & Делікатна білизна (59 хв.), Вовна (59 хв.), Експрес (15 хв.)

0 Days    0 Hour    00 Min    00 Sec

**SHOP NOW**


↑

Рисунок 3.26 – Третій блок сторінки

Переходимо на сторінку будь – якого товару (рис. 3.27). Тут можемо ознайомитись з усіма характеристиками, обрати те що нам треба і додати товар у кошик.

HOME    TECHNIQUE    ABOUT    CONTACT US

🔍 👤 🛒



Ноутбук ACER Nitro 5 AN515-57-57ML (NH.QEKEU.004)  
\$32000

Color: Choose An Option

QTY: - 2 +

**ADD TO CART** 🛒

📧 📱 📺 📷 📹

**DESCRIPTION** | MORE INFORMATION | REVIEWS (2)

Популярний ноутбук Nitro 5 буде відмінно справлятися не тільки з усіма сучасними іграми, але й іншими завданнями користувача. Шестиядерний процесор Intel Core i5-11400H II покоління разом з відеокартою NVIDIA GeForce GTX 1650 4 GB GDDR6 видаватимуть вражаючу продуктивність, а якісна матриця з частотою оновлення 144 Гц забезпечить ідеальну плавність у будь-якій анімації на екрані. Високошвидкісна оперативна пам'ять об'ємом 16 Гб (з можливістю розширення до 32 Гб завдяки додатковим слотам) дасть змогу працювати з кількома складними завданнями одночасно, а швидкий диск SSD на 512 Гб значно зменшить час старту операційної системи та всіх встановлених програм. Продумана система охолодження з двох кулерів та працююча з технологією CoolBoost не дозволить ноутбуку перегріватися навіть після тривалої роботи під навантаженням. За допомогою фирмової програми Nitro Sense ви зможете зручно контролювати всі аспекти роботи системи. А яскраве підсвічування клавіатури дозволить вам у будь-яких умовах бачити потрібні для вас клавіші, і дозволить ноутбуку виглядати ще красивішим і стильнішим.

Рисунок 3.27 – Сторінка з товаром

Переходимо до останнього, четвертого, блоку головної сторінки (рис. 3.28). Тут є інформація про наші переваги при замовленні товару у нас і футер сайту.

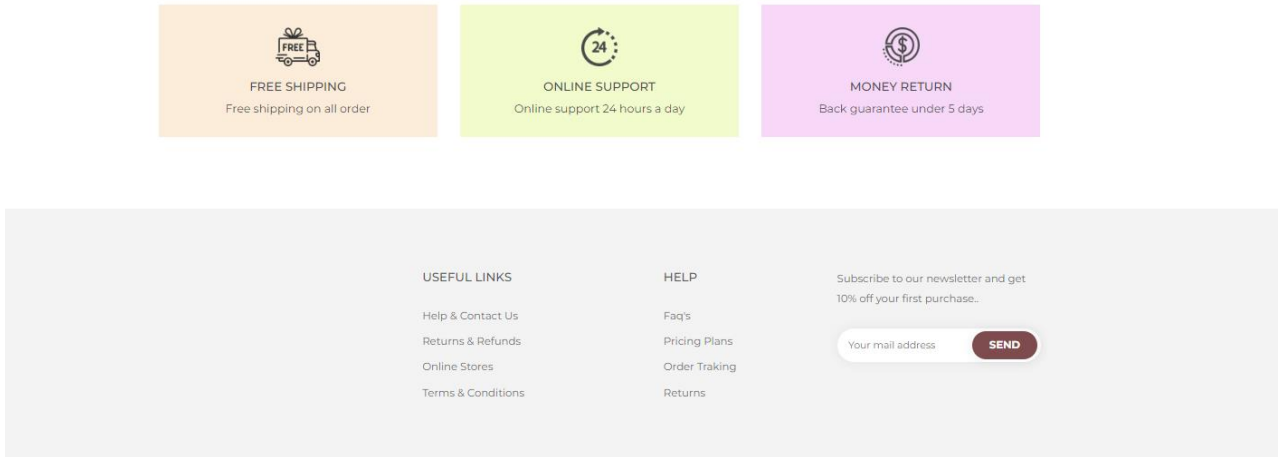


Рисунок 3.28 – Четверий блок сайту

Переходимо до сторінки з товаром (див. рис. 3.29). На сторінці ми бачимо можливість здійснити пошук за назвою, або вибрати за допомогою фільтра необхідні характеристики товару.

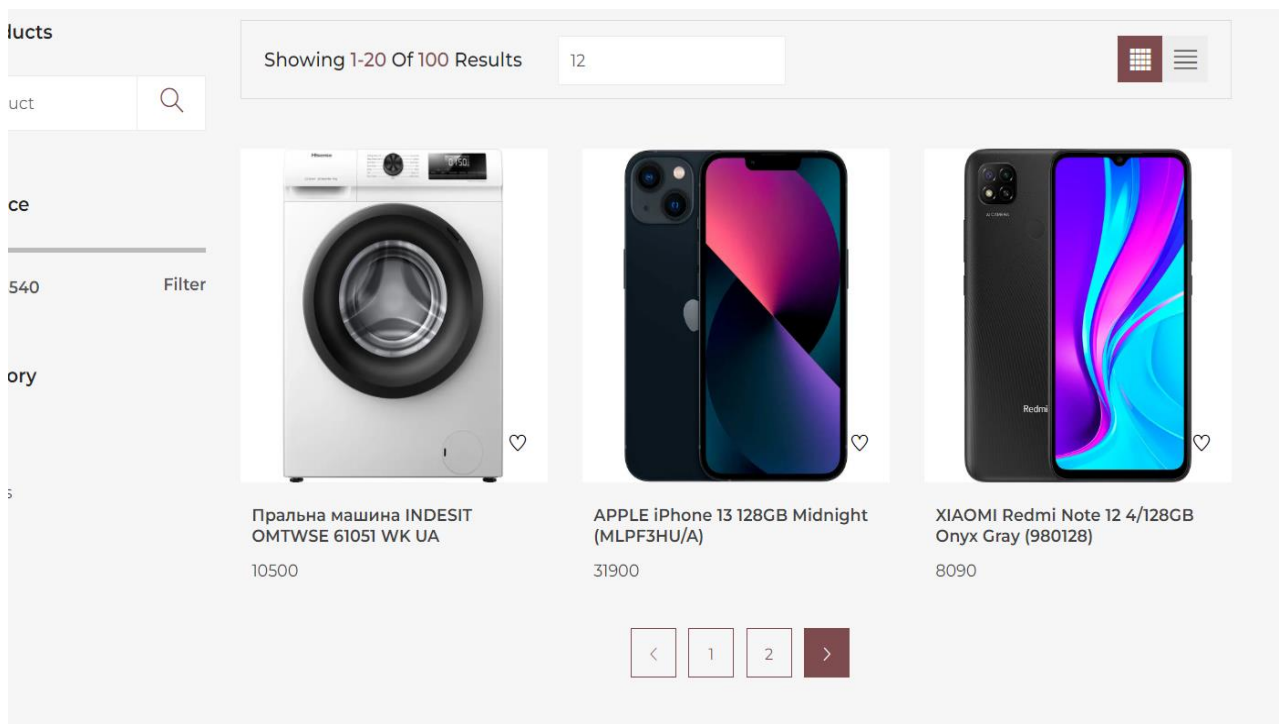


Рисунок 3.29 – Сторінка товару

Після того як ми обрали потрібні нам товари та добавили їх до корзини (див. рис. 3.30), всі вибрані товари відображаються у ній, пишеться загальна сума та кнопка з переходом до форми оплати товару (рис. 3.31).

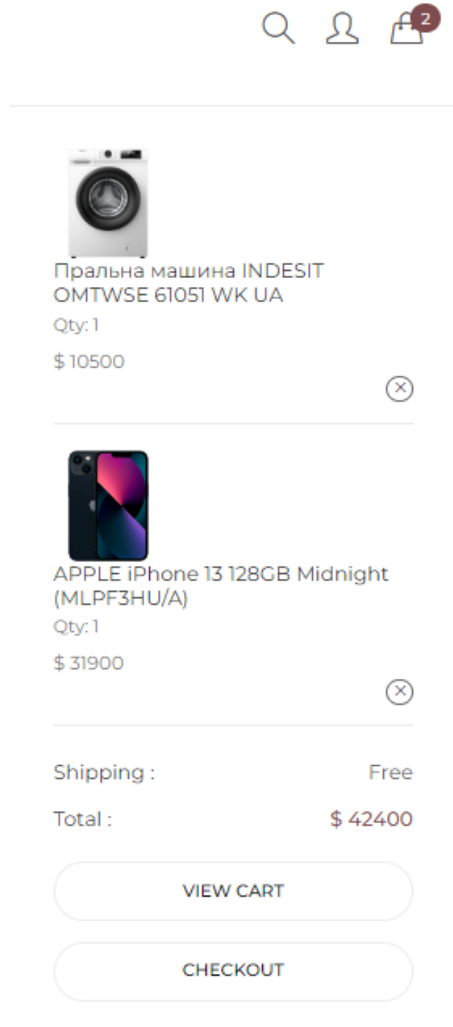


Рисунок 3.30 – Корзина сайту

Сторінка з формою оплати товару після корзини в інтернет-магазині виконує наступні функції:

- завершення процесу замовлення;
- вибір способу оплати;
- введення необхідних даних;
- безпека операцій;
- підтвердження оплати.



## CHECKOUT

1 BILLING INFORMATION
▼

First Name

Email Address

Address

City

State/Province

Zip/Postal Code

Telephone

| Product Name                               | Price    | Qty | Subtotal        |
|--|----------|-----|-----------------|
| APPLE iPhone 13 128GB Midnight (MLPF3HU/A) | \$ 31900 | 1   | 31900           |
| <b>Grand Total</b>                         |          |     | <b>\$ 31900</b> |

[Forgot an Item? Edit Your Cart.](#)

MAKE AN ORDER

Рисунок 3.31 – Форма оплати товару

Переходимо до сторінки «Contact us» (див. рис. 3.32). Тут клієнти сайту можуть знайти всю необхідну інформацію для зв'язку з сайтом і менеджерами.


Сторінка "Contact us" в інтернет-магазині необхідна для того, щоб забезпечити зручний і простий спосіб зв'язку між клієнтами і командою магазину. На цій сторінці можуть бути надані контактні дані, такі як електронна пошта, телефонний номер або форма зворотного зв'язку, за допомогою яких клієнти можуть звернутися зі своїми запитаннями, проблемами або пропозиціями. Сторінка "Contact us" дозволяє клієнтам:

- задати питання;
- повідомити про проблеми;
- залишити відгук.


---

HOME    TECHNIQUE    ABOUT    CONTACT US


🔍 👤 📧<sup>1</sup>



**Our Location**  
012 345 678 / 123 456 789  
info@example.com



**Contact us Anytime**  
Mobile: 012 345 678  
Fax: 123 456 789



**Write Some Words**  
Support24/7@example.com  
info@example.com

**GET IN TOUCH**

Full Name     Email Address

Subject

Message

**SEND MESSAGE**

Рисунок 3.32 – Сторінка «Contact us»

## ВИСНОВКИ

Інтернет магазин – досить складний проєкт через велику кількість сутностей та взаємодію з ними. В результаті було отримано досить якісний продукт. Клієнт може замовляти товари, оформлювати замовлення, а менеджер може редагувати замовлення та зберігати. Було автоматизовано досить багато процесів.

Проєкт реалізований за допомогою Laravel. Збоку частини клієнтської, використана бібліотека Node.js. Адмін частина за допомогою Voyager.

У проєкту є досить великий потенціал для розвитку. Було реалізовано сервіси для сплати та відправлення повідомлення користувачу в зручній йому месенджер або на пошту. Було зроблено перевірки та авторизацію менеджера.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Суркова Н. Е., Остроух А. В. Методологія структурного проєктування інформаційних систем. Харків: Науково-інноваційний центр, 2014. 184 с.
2. Wikipedia. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_послідовності](https://uk.wikipedia.org/wiki/Діаграма_послідовності) (дата звернення 29.04.2023).
3. Laravel. URL: <https://laravel.com/docs/10.x> (дата звернення 30.04.2023).
4. Repository Voyager. URL: <https://github.com/the-control-group/voyager>. (дата звернення 05.04.2023).
5. ChatGpt. URL: <https://chat.openai.com/?model=text-davinci-002-render-sha> (дата звернення 05.04.2023).
6. Node.js. URL: <https://dan-it.com.ua/uk/blog/chto-jeto-takoe-node-js-> (дата звернення 06.04.2023).
7. Java. URL: <https://uk.wikipedia.org/wiki/Java> (дата звернення 07.05.2023).
8. Wikipedia. URL: [https://www.wikiwand.com/uk/Діаграма\\_прецедентів](https://www.wikiwand.com/uk/Діаграма_прецедентів). (дата звернення 09.04.2023).

## ДОДАТОК А

### Controller та маршрути сторінок у файлі web.php

```
public function shopIndex()
{
    $randProducts = Product::query()->inRandomOrder()->limit( value: 4)->get();

    $product = Product::query()->select()->inRandomOrder()->limit( value: 1)->get();

    $sessionId = Session::getId();

    \Cart::session($sessionId);

    $cart = \Cart::getContent();

    $sum = \Cart::getTotal('price');

    return view( view: 'electronic-shop/index', [
        'randProducts' => $randProducts,
        'product' => $product,
        'cart' => $cart,
        'sum' => $sum
    ]);
}
```

Рисунок А.1 – Controller функція shopIndex

```
public function shopList()
{
    $products = Product::query()->limit( value: 3)->offset( value: 1)->get();

    $sessionId = Session::getId();

    \Cart::session($sessionId);

    $cart = \Cart::getContent();

    $sum = \Cart::getTotal('price');

    return view( view: 'electronic-shop/shop-page', [
        'products' => $products,
        'cart' => $cart,
        'sum' => $sum,
    ]);
}
```

Рисунок А.2 – Controller функція shopList

```

public function productDetails(Request $request)
{
    $product = Product::query()->where(['id' => $request->id])->get();

    $sessionId = Session::getId();

    \Cart::session($sessionId);

    $cart = \Cart::getContent();

    $sum = \Cart::getTotal('price');

    return view('electronic-shop/product-details', [
        'product' => $product,
        'cart' => $cart,
        'sum' => $sum,
    ]);
}

```

Рисунок А.3 – Controller функція productDetails

```

public function addCart(Request $request)
{
    $product = Product::query()->where(['id' => $request->id])->first();

    $sessionId = Session::getId();

    \Cart::session($sessionId)->add([
        'id' => $product->id,
        'name' => $product->name,
        'price' => $product->price,
        'quantity' => $request->qty ?? 1,
        'attributes' => [
            'image' => $product->image,
        ],
    ]);

    $cart = \Cart::getContent();

    return redirect()->back();
}

```

Рисунок А.4 – Controller функція addCart

```

public function contact()
{
    $sessionId = Session::getId();

    \Cart::session($sessionId);

    $cart = \Cart::getContent();

    $sum = \Cart::getTotal('price');

    return view( view: 'electronic-shop/contact', [
        'cart' => $cart,
        'sum' => $sum,
    ]);
}

```

Рисунок А.5 – Controller функція contact

```

Route::get( uri: '/', [\App\Http\Controllers\ProductController::class, 'shopIndex']->name( name: 'home'));
Route::get( uri: 'electronic-shop/food', [\App\Http\Controllers\ProductController::class, 'shopList'])
->name( name: 'electronic-shop/food');
Route::get( uri: 'electronic-shop/contact', [\App\Http\Controllers\ProductController::class, 'contact'])
->name( name: 'electronic-shop/contact');
Route::get( uri: 'electronic-shop/about', [\App\Http\Controllers\PetController::class, 'about'])
->name( name: 'electronic-shop/about');
Route::get( uri: 'electronic-shop/product-details', [\App\Http\Controllers\ProductController::class, 'productDetails'])
->name( name: 'electronic-shop/product-details');
Route::get( uri: '/dashboard', function () {
    return view( view: 'dashboard');
})->middleware(['auth']->name( name: 'dashboard');
Route::get( uri: 'electronic-shop/add-cart', [\App\Http\Controllers\ProductController::class, 'addCart'])
->name( name: 'electronic-shop/add-cart');
Route::get( uri: 'electronic-shop/checkout', [\App\Http\Controllers\ProductController::class, 'checkout'])
->name( name: 'electronic-shop/checkout')->middleware( middleware: 'auth');
Route::get( uri: 'electronic-shop/profile', [\App\Http\Controllers\ProductController::class, 'profile'])
->name( name: 'electronic-shop/profile')->middleware( middleware: 'auth');
Route::post( uri: 'electronic-shop/make-order', [\App\Http\Controllers\ProductController::class, 'makeOrder'])
->name( name: 'electronic-shop/make-order')->middleware( middleware: 'auth');

```

Рисунок А.6 – Усі прописані маршрути з Controller