МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВА РОБОТА**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Бази даних»

на тему:

**« Розробка інтернет магазину Apple»**

студента ІI курсу групи ІПЗ-20-2

спеціальності 121 «Інженерія програмного забезпечення»

Волинця Владислава Михайловича

(прізвище, ім’я та по-батькові)

Керівник: ст. вкл. каф. ІПЗ Чижмотря О.В.

Дата захисту: " 1" серпня 2022 р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Чижмотря \_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Коротун.

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ І.І. Сугоняк\_

(підпис) (прізвище та ініціали)

Житомир – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

В.о.зав. кафедри

\_\_\_\_\_\_\_\_\_\_\_\_ А.В.Морозов

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2022 р.

ЗАВДАННЯ

НА КУРСОВИЙ ПРОЕКТ СТУДЕНТУ

Волинцю Владиславу Михайловичу

1. Тема роботи: Розробка "Інтернет магазин Apple".
2. Керівник роботи: ст. вкл. каф. ІПЗ Чижмотря О.В.
3. Строк подання студентом: “ 1” серпня 2022р.
4. Вихідні дані до роботи: Розробити Базу Даних.
5. Зміст розрахунково-пояснювальної записки(перелік питань, які підлягають розробці)
   * + 1. Постановка завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми
       4. Опис роботи програми
       5. Програмне дослідження
6. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)

1. Презентація до КП\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2. 2. Посилання на репозиторій: <https://gitlab.com/courseworks_/applestore>

1. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
| 1,2,3,4,5 | Чижмотря О.В., ст. вик. каф. ІПЗ |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання “ 20 ” лютого 2022 р.

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсового проекту | Строк виконання етапів проекту | Примітки |
| 1 | Постановка задачі | 20.02- 22.02 | Виконано |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 15.06-  20.06 | Виконано |
| 3 | Формулювання технічного завдання | 21.06-  24.06 | Виконано |
| 4 | Опрацювання літературних джерел | 26.06-  28.06 | Виконано |
| 5 | Проектування структури | 29.06-  2.07 | Виконано |
| 6 | Написання програмного коду | 3.07-  23.07 | Виконано |
| 7 | Відлагодження | 24.07-  28.07 | Виконано |
| 8 | Написання пояснювальної записки | 29.07 | Виконано |
| 9 | Захист | 1.08 | Виконано |

**КАЛЕНДАРНИЙ ПЛАН**

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Волинець В.М.

(підпис) (прізвище та ініціали)

**Керівник проекту** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чижмотря О.В.

(підпис) (прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до курсового проекту на тему «Розробка інтернет магазину Apple» складається зі вступу, чотирьох розділів, висновків, списку використаної літератури та додатків.

Загальний обсяг роботи – 44 сторінок.

Пояснювальна записка має 10 сторінок додатків.

Список використаних джерел містить 10 найменувань і займає 1 сторінку.

У роботі наведено 18 рисунків.

Ключові слова: МАГАЗИН, ВЕБ, ДИЗАЙН, ІНТЕРНЕТ, ПОКУПКИ, веб додаток, бд, АДМІН, користувачі.

## ЗМІСТ

[ВСТУП 6](#_Toc110442847)

[РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ІНТЕРНЕТ МАГАЗИНУ 7](#_Toc110442848)

[1.1 Аналіз інформаційних потреб та визначення предметної області дослідження 7](#_Toc110442849)

[1.2 Обґрунтування вибору засобів реалізації 9](#_Toc110442850)

[Висновки до 1 розділу: 11](#_Toc110442851)

[РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ ІНТЕРНЕТ МАГАЗИНУ 12](#_Toc110442852)

[2.1 Аналіз інформаційних процесів 12](#_Toc110442853)

[2.2 Проектування структури бази даних сайту 14](#_Toc110442854)

[Висновки до 2 розділу: 19](#_Toc110442855)

[РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ ІНТЕРНЕТ МАГАЗИНУ 20](#_Toc110442856)

[3.1 Проектування інтерфейсу обробки даних 20](#_Toc110442857)

[3.2 Реалізація операцій обробки даних в БД 24](#_Toc110442858)

[3.3 Організація звітності системи 26](#_Toc110442859)

[Висновки до 3 розділу: 29](#_Toc110442860)

[РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ 30](#_Toc110442861)

[4.1 Розробка заходів захисту інформації в БД. 30](#_Toc110442862)

[4.2 Налаштування параметрів роботи сервера 31](#_Toc110442863)

[Висновки до 4 розділу: 31](#_Toc110442864)

[ВИСНОВКИ 32](#_Toc110442865)

[СПИСОК ЛІТЕРАТУРИ 33](#_Toc110442866)

[ДОДАТКИ 34](#_Toc110442867)

[Додаток А 35](#_Toc110442868)

[Додаток Б 41](#_Toc110442869)

## ВСТУП

**Актуальність теми**: На сьогоднішній день в сучасному світі вже неможливо уявити життя без веб-технологій. Кожний бізнес має свій інетрнет магазин. Це набагато спрощує доступ до продуктів та інформації. Також сайти допомагають замовляти товари із будь-яких куточків світу.

**Тема курсового проекту**: «Інтернет магазин Apple». Даний сайт зможе надавати користувачам доступ до каталогу сайту та змогу замовляти товари.

**Метою створення** даного курсового проекту є розробка веб додатку, який надає користувачам таки дії :

1. Перегляд каталогу товарів.
2. Перегляд корзини.
3. Відправку форми зворотного зв’язку.
4. Замовлення товарів.
5. Перегляд галереї та контактів.

**Об’єктом дослідження** є розробка бази даних для інтернет магазину Apple використовуючу MySql та Php.

**Предметом дослідження**: аналіз інших сайтів, реалізація свого, здобуття навичок із роботою базами даних, мовами програмування.

## РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ІНТЕРНЕТ МАГАЗИНУ

# **1.1 Аналіз інформаційних потреб та визначення предметної області дослідження**

На початку роботи я ознайомимося із поставленою задачею курсової роботи. Це дозволить більш ефективно та якісно продумати план розробки веб додатку.

Задачею даної роботи є створення функціонального веб додатку із створенням власної БД.

Для її вирішення використовується мова програмування Php в середовищі програмування PhpStorm.

Проведемо аналіз декількох сайтів із схожою тематикою. Перший такий сайт буде ЯБКО (рис. 1.1). За допомогою цього ми можемо переглянути інформацію про наявні товари, дізнатися більше інформації про конкретний товар, при покупці товару одразу здійснюється замовлення де ми вказуємо відповідні для зв’язку та вказуємо метод доставки.

Переваги: сучасний дизайн, легкість у використанні.

Недоліки: відсутність особистого кабінету.

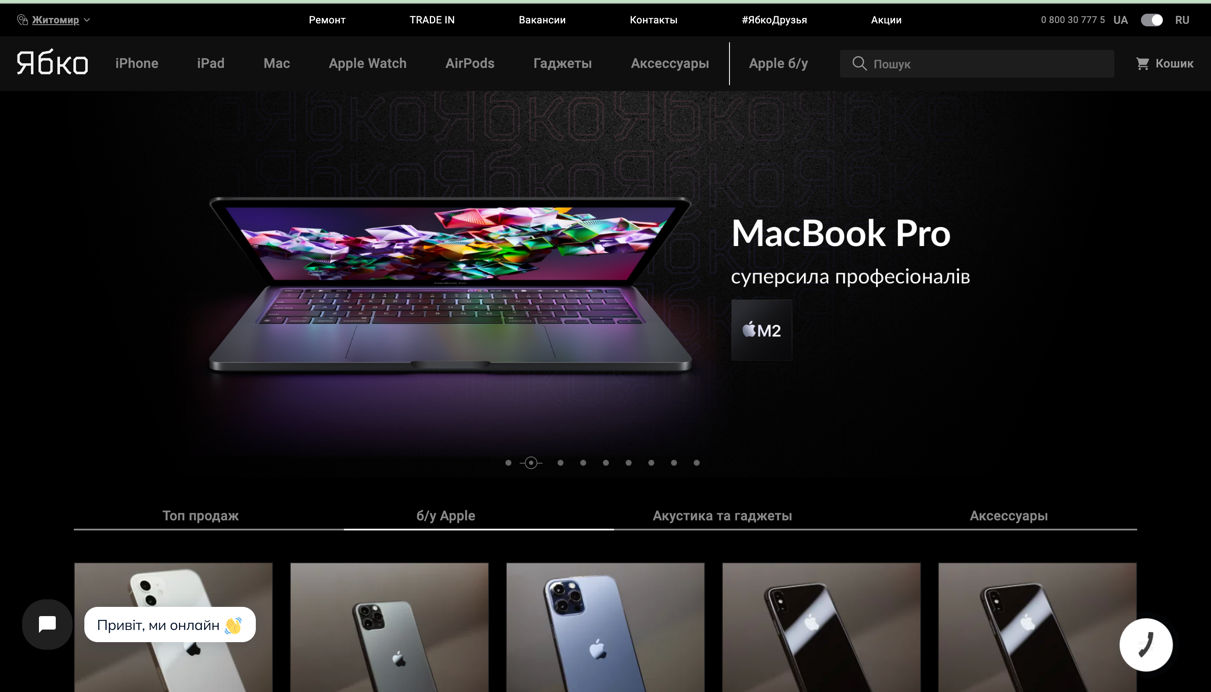


Рис. 1.1 Інтерфейс сайту ЯБКО

Наступний сайт – офіційний сайт Apple (рис. 1.2).

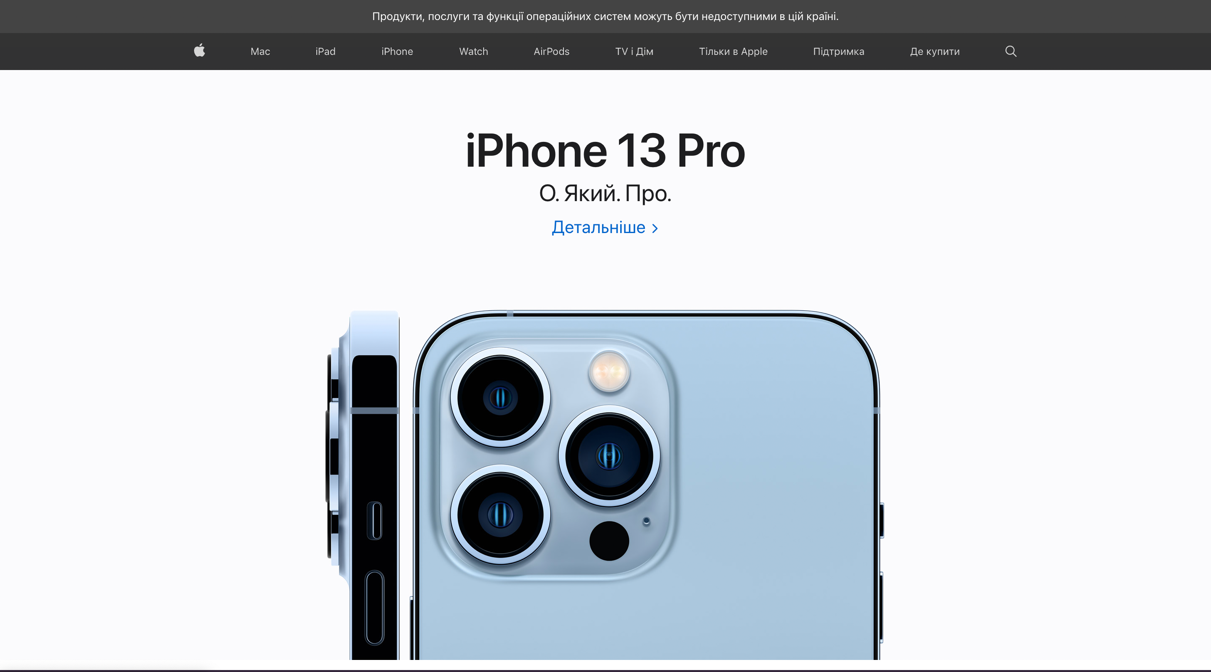


Рис. 1.2 Інтерфейс сайту Apple.com

Переваги: красивий дизайн, легкий у використанні, чудова галерея.

Недоліки: відсутність особистого кабінету.

Проаналізувавши ці головні сайти, створимо свій, який буде виконувати головні функції для користувачів.

# **1.2 Обґрунтування вибору засобів реалізації**

Систе́ма управління ба́зами да́них (СУБД, СКБД англ. Database Management System, DBMS) — набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних . Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

Перед реалізацією курсової роботи було проведено порівняльну характеристику декількох розповсюджених СУБД.

Таблиця 1.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Харктеристика | Microsoft SQL Server | Oracle | MongoDB | SQLite | MySQL |
| Адміністративне керування | Добре | Відмінно | Відмінно | Відсутнє | Добре |
| Графічні інструменти | Відмінно | Добре | Добре | Відсутнє | Відмінно |
| Простота обслуговування | Відмінно | Відмінно | Відмінно | Відмінно | Відмінно |
| Механізм даних | Добре | Відмінно | Відмінно | Відмінно | Відмінно |
| Одночасний доступ декількох користувачів | Добре | Відмінно | Відмінно | Відмінно | Відмінно |
| Повнотекстовий пошук | Добре | Відмінно | Задовільно | Відмінно | Добре |
| Інтеграція з іншими СУБД | Добре | Добре | Задовільно | Відсутня | Відсутня |
| Мова SQL | Відмінно | Відмінно | Відмінно | Відсутня | Відсутня |
| Засоби завантаження | Відмінно | Відмінно | Відмінно | Відмінно | Відмінно |
| Засоби аналізу | Відмінно | Відмінно | Відмінно | Задовільно | Відмінно |

В якості мови програмування було обрано MySQL, а для адміністрування phpMyAdmin, який вже є вбудованим у MAMP.

Для написання серверної частини було обрано мову програмування PHP версії 8.0 із використанням фреймворку Laravel, для браузерної: HTML, CSS, JavaScript.

Середовищем розробки для написання веб-додатку слугував PhpStorm.

# **Висновки до 1 розділу:**

Було проаналізовано схожі інтернет-мазазини та сформовано їх позитивні та негативні сторони. Також було зроблено порівняння різних СУБД та обрання мов програмування веб додатку.

## РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ ІНТЕРНЕТ МАГАЗИНУ

# **2.1 Аналіз інформаційних процесів**

Побудуємо блок схему загального алгоритму нашого веб додатку.

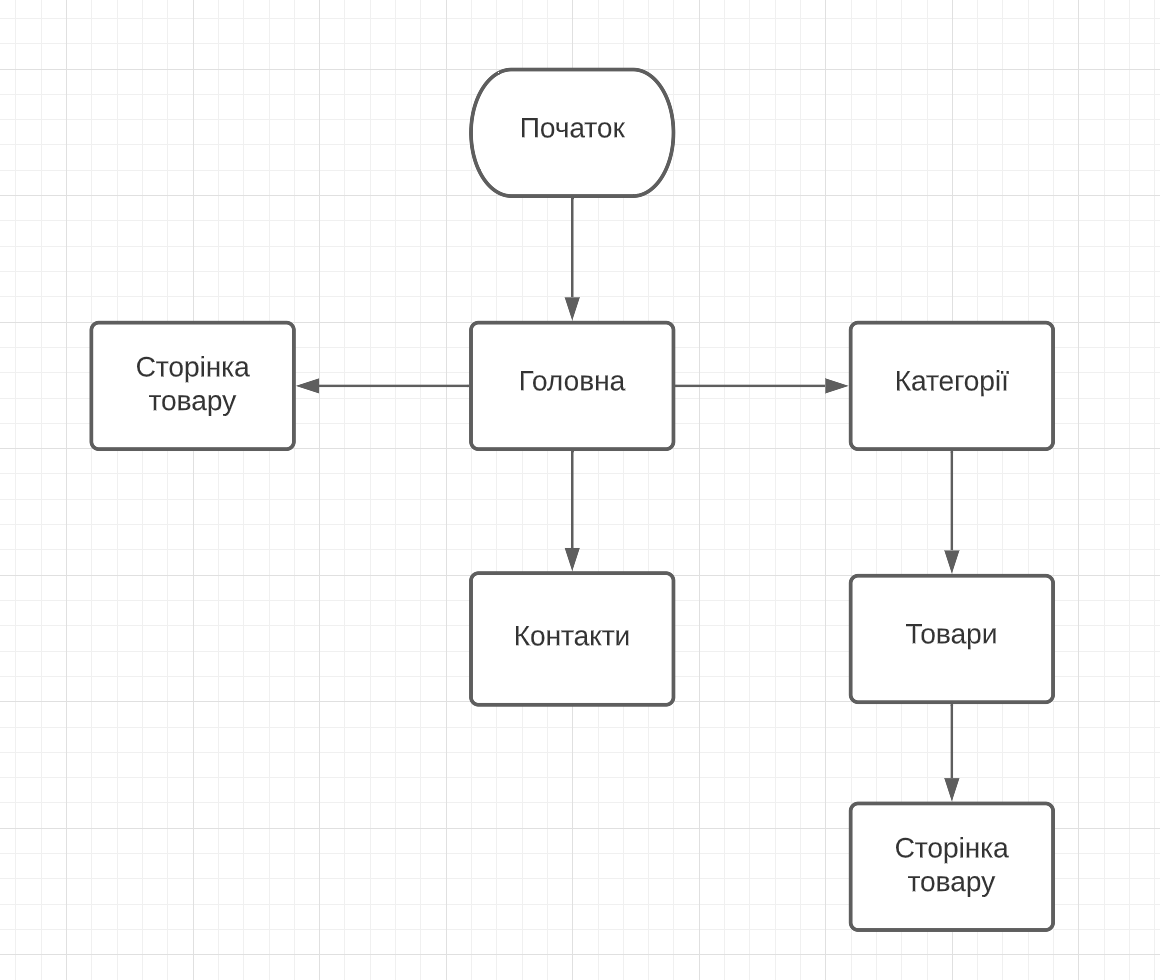


Рис. 2.1 Блок-схема роботи веб додатку (рис. 2.1).

Побудуємо блок схему адмінської частини нашого веб додатку (рис 2.2).

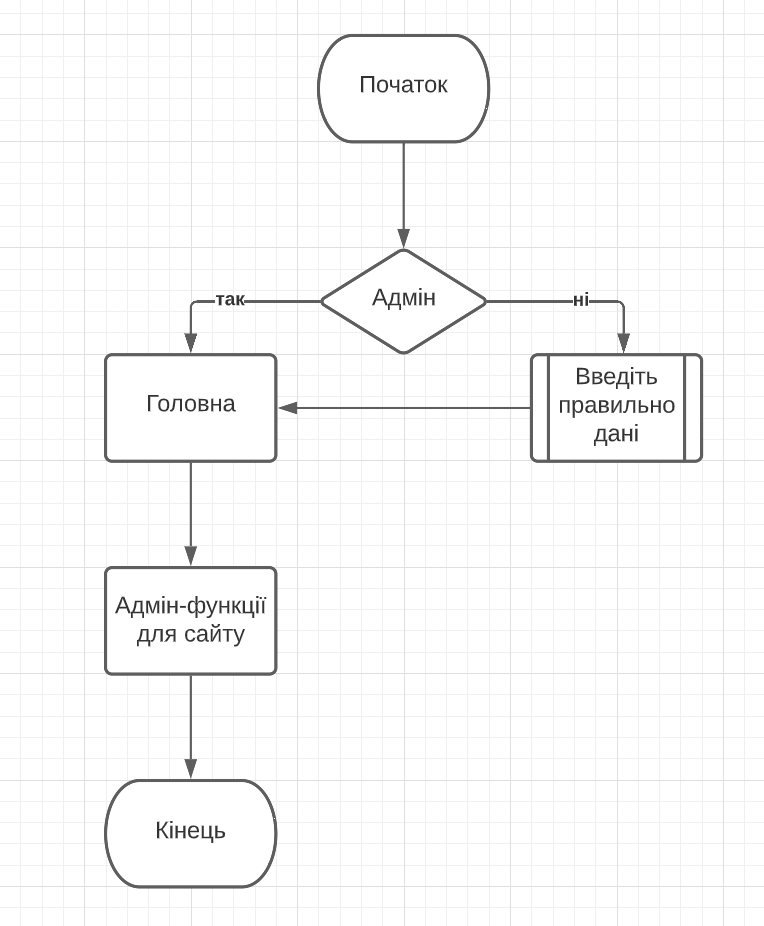


Рис. 2.2 Блок-схема роботи адмінської частини (рис. 2.2).

Побудуємо блок схему замовлення товарів нашого веб додатку (рис 2.3).



Рис. 2.3 Блок-схема роботи замовлення товарів (рис. 2.3).

# **2.2 Проектування структури бази даних сайту**

Проектування бази даних відбувалось на сайті phpMyAdmin, де були створені таблиці, які будуть потрібні у подальшій розробці.

Таблиця 2.1

Таблиця “categories”

|  |  |  |
| --- | --- | --- |
| Назва | Тип даних | Опис поля |
| id | bigint(20) | Id категорії |
| title | varchar(255) | Назва категорії |
| description | text | Опис категорії |
| img | varchar(255) | Шлях до картинки |
| alias | varchar(255) | Адреса категорії |

Таблиця 2.2

Таблиця “products”

|  |  |  |
| --- | --- | --- |
| Назва | Тип даних | Опис поля |
| id | bigint(20) | Id товару |
| title | varchar(255) | Назва товару |
| description | text | Опис товару |
| img | varchar(255) | Шлях до картинки |
| alias | varchar(255) | Адреса товару |
| price | int(11) | Ціна |
| in\_stock | tinyint(1) | В наявності |
| category\_id | bigint(20) | Id категорії |
| color | varchar(255) | Колір |

Таблиця 2.3

Таблиця “product\_images”

|  |  |  |
| --- | --- | --- |
| Назва | Тип даних | Опис поля |
| id | bigint(20) | Id товару |
| img | varchar(255) | Шлях до картинки |
| product\_id | bigint(20) | Id продукту |

Таблиця 2.4

Таблиця “orders”

|  |  |  |
| --- | --- | --- |
| Назва | Тип даних | Опис поля |
| id | bigint(20) | Id замовлення |
| fname | varchar(255) | Ім’я |
| lname | varchar(255) | Прізвище |
| phone | varchar(255) | Номер телефону |
| email | varchar(255) | Пошта |
| country | varchar(255) | Країна |
| city | varchar(255) | Місто |
| address | varchar(255) | Адреса |
| pincode | varchar(255) | Поштовий індекс |
| status | tinyint(4) | Статус |
| message | varchar(255) | Повідомлення |
| tracking\_no | varchar(255) | ТТН |

Таблиця 2.5

Таблиця “order\_items”

|  |  |  |
| --- | --- | --- |
| Назва | Тип даних | Опис поля |
| id | bigint(20) | Id замовленого товару |
| order\_id | varchar(255) | Id замовлення |
| product\_id | varchar(255) | Id товару |
| qty | varchar(255) | Кількість |
| price | varchar(255) | Ціна |

Таблиця 2.6

Таблиця “galleries”

|  |  |  |
| --- | --- | --- |
| Назва | Тип даних | Опис поля |
| id | bigint(20) | Id слайду |
| title | varchar(255) | Заголовок |
| desc | text | Опис |
| img | varchar(255) | Шлях до картинки |

Таблиця 2.7

Таблиця “features”

|  |  |  |
| --- | --- | --- |
| Назва | Тип даних | Опис поля |
| id | bigint(20) | Id особливості |
| title | varchar(255) | Заголовок |
| desc | text | Опис |
| icon | varchar(255) | Шлях до іконки |

Таблиця 2.8

Таблиця “information”

|  |  |  |
| --- | --- | --- |
| Назва | Тип даних | Опис поля |
| id | bigint(20) | Id контактної інформації |
| title | varchar(255) | Заголовок |
| phone | varchar(255) | Телефон |
| email | varchar(255) | Пошта |

# **Висновки до 2 розділу:**

Складено загальний алгоритм роботи програми, алгоритм адмінської частини та алгоритм придбання товарів. Створена база даних із необхідними таблицями та описом до них.

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ ІНТЕРНЕТ МАГАЗИНУ

# **3.1 Проектування інтерфейсу обробки даних**

Для виконання цієї роботи я використовував PhpStorm. Дане середовище надає зручний інтерфейс та має зручну систему контролю версій. Після того як користувач заходить на сайт його зустрічає головна сторінка. На головній сторінці ми бачимо галерею, товари із знижкою та особливості фірми.

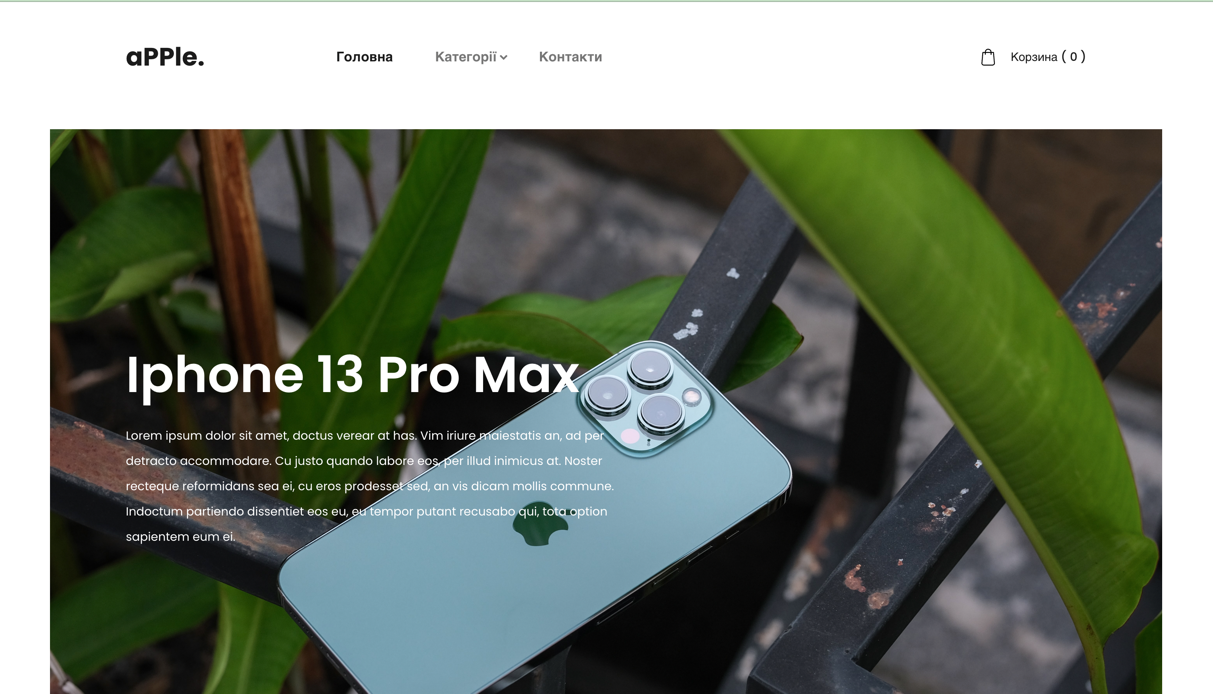


Рис. 3.1 Головна сторінка

Після цього ми можемо перейти на бажану нами категорію. Наприклад, це буде Macbook.

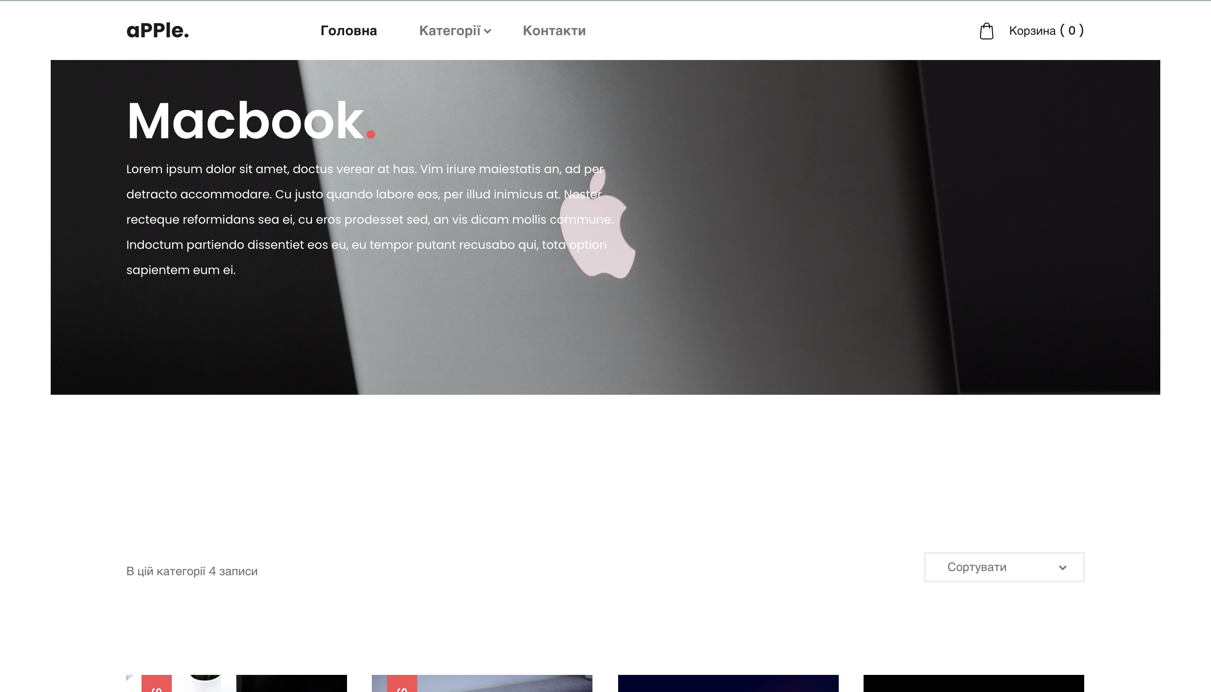


Рис. 3.2 Сторінка категорії Macbook

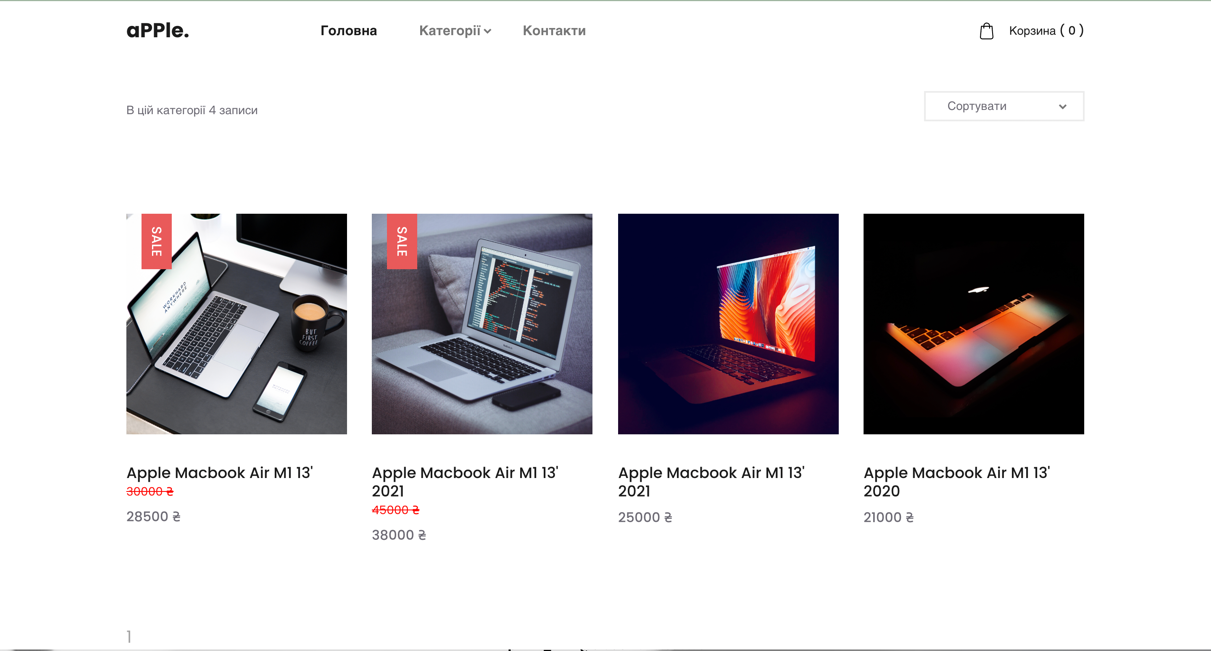


Рис. 3.3 Перегляд товарів в категорії Macbook

Перейдемо на якийсь товар та замовимо його.

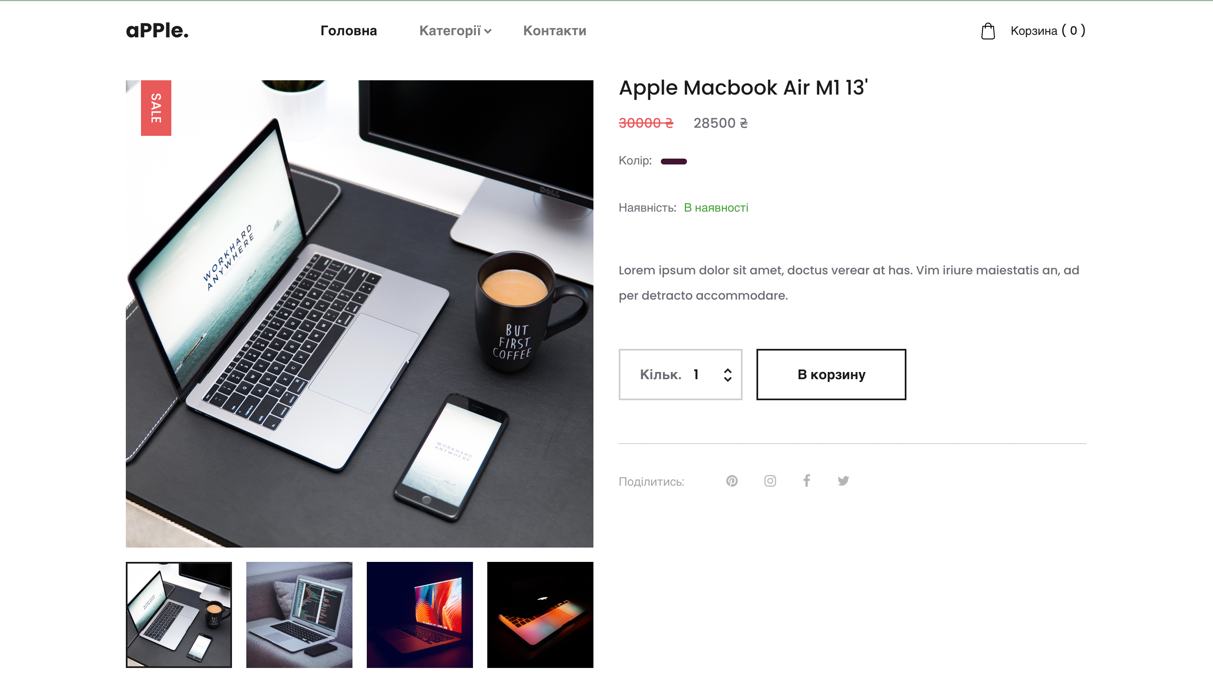


Рис. 3.4 Сторінка товара

Добавимо цей товар у корзину.

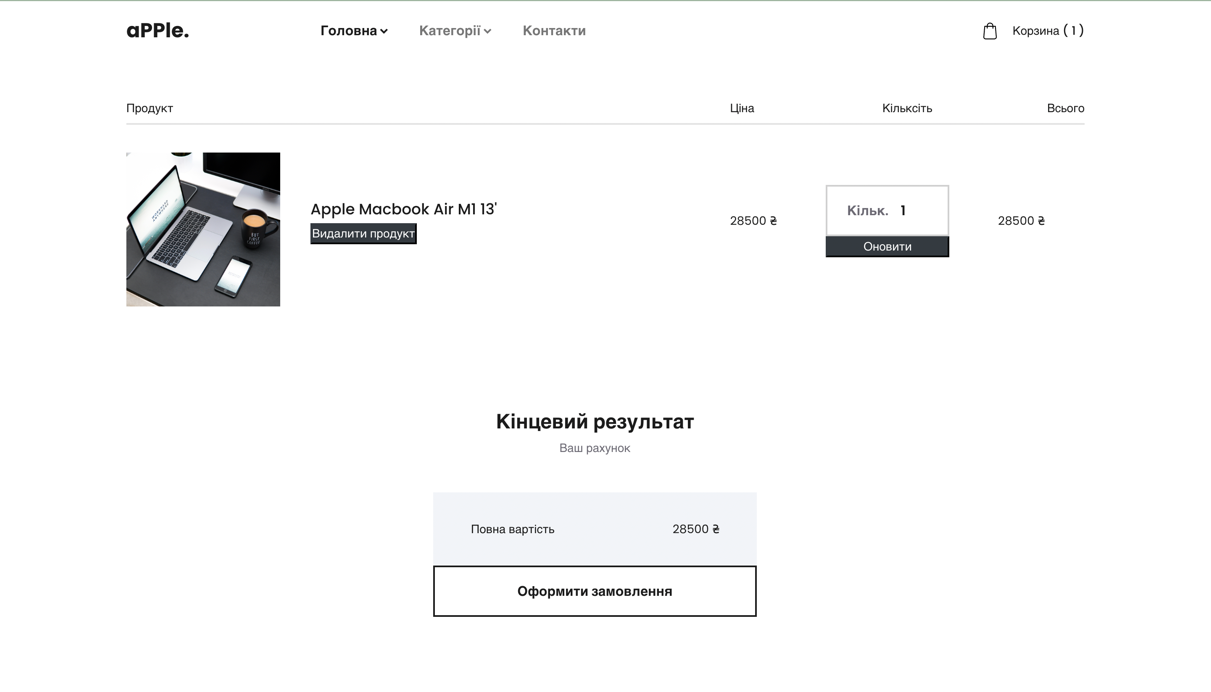


Рис. 3.5 Корзина

Перейдемо до оформлення замовлення.

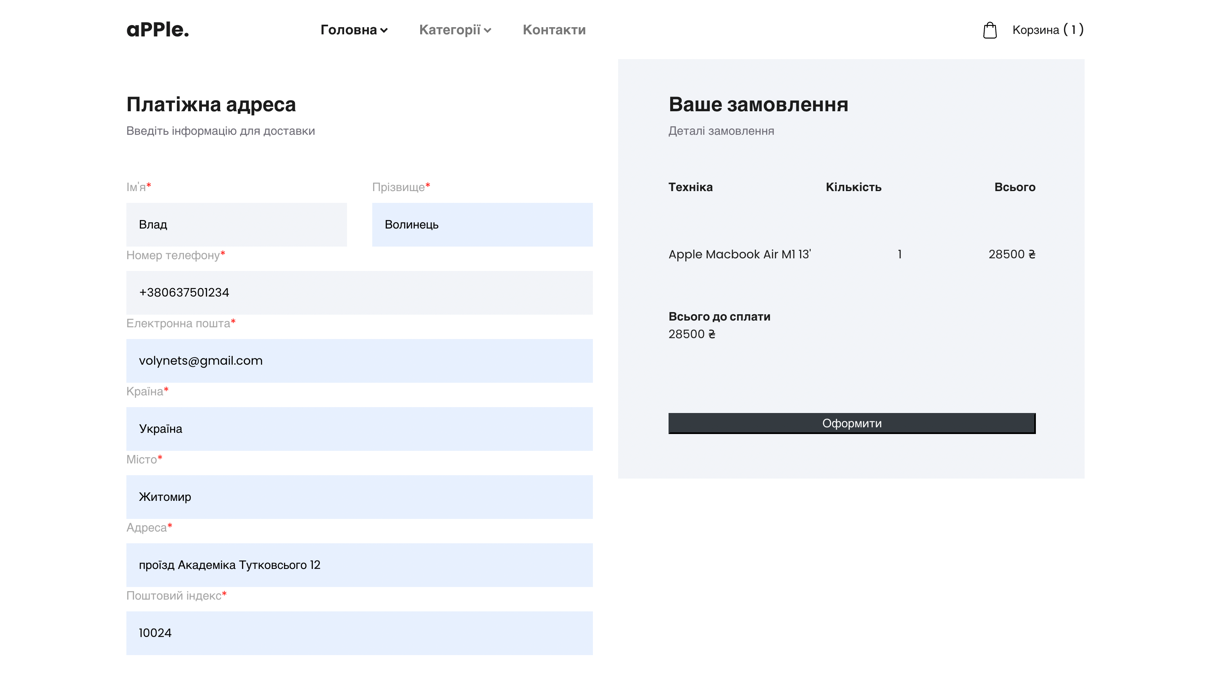


Рис. 3.6 Оформлення

Також на сайті є форма зворотнього зв’язку.

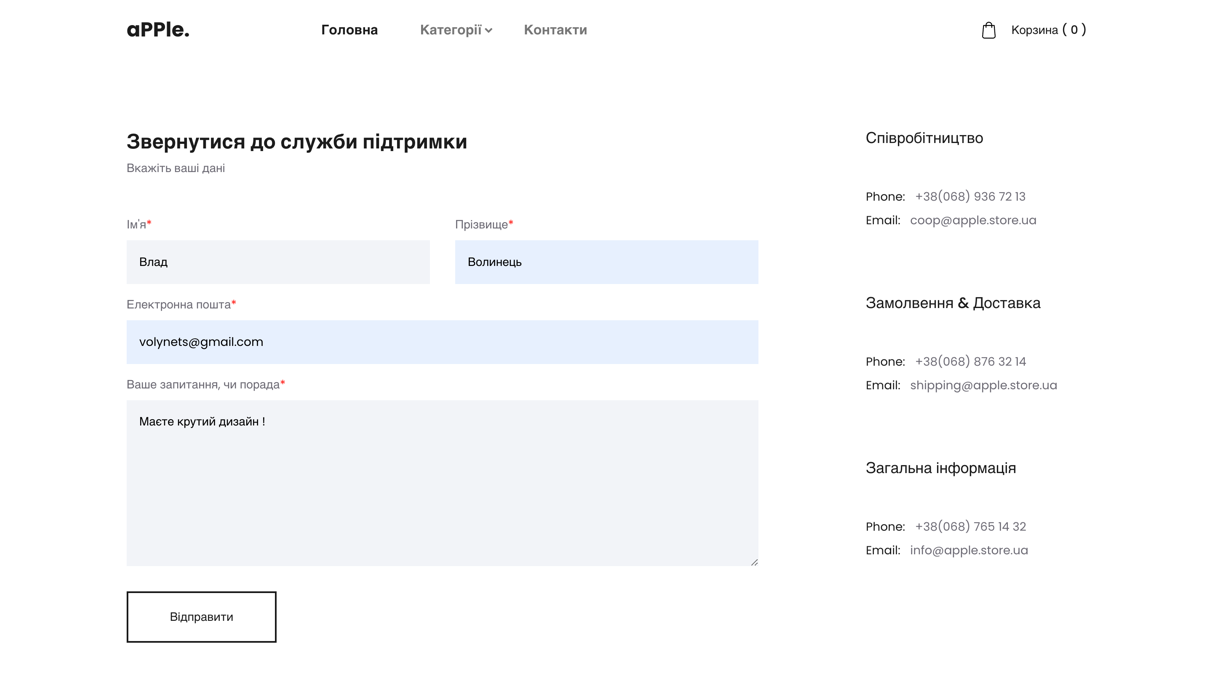


Рис. 3.7 Форма зворотнього зв’язку

# **3.2 Реалізація операцій обробки даних в БД**

Для реалізації проекту було використано СУБД MySql. Робота з бд здійсновалася через сайт phpMyAdmin. Реалізація була здійснена у програмі PhpStorm мовою програмування PHP.

Реалізуємо підключення до бази даних.

DB\_CONNECTION=mysql  
DB\_HOST=127.0.0.1  
DB\_PORT=8889  
DB\_DATABASE=applestore  
DB\_USERNAME=root  
DB\_PASSWORD=root

Відображення товарів.

public function show($cat, $product\_alias) {  
 $item = Product::*where*('alias', $product\_alias)->first();  
 $category = Category::*where*('id', $item->category\_id)->first();  
 $products = Product::*where*('category\_id', $item->category\_id)->WhereNot('alias', $product\_alias)->orderBy('created\_at')->take(4)->get();  
 return view('product/show', ['item' => $item, 'category' => $category, 'products' => $products]);  
}

Відображення каталогів.

public function showCategory(Request $request, $cat\_alias) {  
 $cat = Category::*where*('alias',$cat\_alias)->first();  
  
 $paginate = 4;  
 $products = Product::*where*('category\_id',$cat->id)->paginate($paginate);  
  
 if(isset($request->orderBy)){  
 if($request->orderBy == 'price-low-high'){  
 $products = Product::*where*('category\_id',$cat->id)->orderBy('price')->paginate($paginate);  
 }  
 if($request->orderBy == 'price-high-low'){  
 $products = Product::*where*('category\_id',$cat->id)->orderBy('price','desc')->paginate($paginate);  
 }  
 }  
  
 if($request->ajax()){  
 return view('ajax.order-by',[  
 'products' => $products  
 ])->render();  
 }  
  
 return view('categories.index',[  
 'cat' => $cat,  
 'products' => $products,

])  
  
  
}

Створення замовлення.

public function placeorder(Request $request) {  
 $order = new Orders();  
 $order->fname = $request->input('fname');  
 $order->lname = $request->input('lname');  
 $order->phone = $request->input('phone');  
 $order->email = $request->input('email');  
 $order->country = $request->input('country');  
 $order->pincode = $request->input('pincode');  
 $order->city = $request->input('city');  
 $order->address = $request->input('address');  
 $order->tracking\_no = 'apple\_store'.rand(1111,9999);  
 $order->save();  
  
 $order->id;  
  
 if(!isset($\_COOKIE['cart\_id'])) setcookie('cart\_id', uniqid());  
 $cart\_id = $\_COOKIE['cart\_id'];  
 \Cart::*session*($cart\_id);  
 foreach(\Cart::*getContent*() as $item){  
 OrderItems::*create*([  
 'order\_id' => $order->id,  
 'product\_id' =>$item->id,  
 'qty' => $item->quantity,  
 'price'=>$item->price \* $item->quantity  
 ]);  
 }  
  
  
  
 \Cart::*clear*();  
  
 return redirect('/cart/checkout')->with('success', 'Успішно! Код вашого замовлення - '.$order->tracking\_no);  
}

Форма зворотнього зв'язку.

public function store() {  
 $data = request()->validate([  
 'name' => 'required',  
 'last\_name' => 'required',  
 'email' => 'required|email',  
 'message' => 'required',  
 ]);  
  
 Mail::to('applestore@gmail.com')->send(new ContactFormMail($data));  
  
 return redirect(route("contact\_form"))->with('success', 'Форма успішно відправлена!');  
}

# **3.3 Організація звітності системи**

Для користувача у якого є доступ до прав адміністратора відобразяться додатковий функціонал сайту, а саме робота з товарами, категоріями, замовленями і так далі. Також буде можливо переглянути статистику зароблених грошей за день.

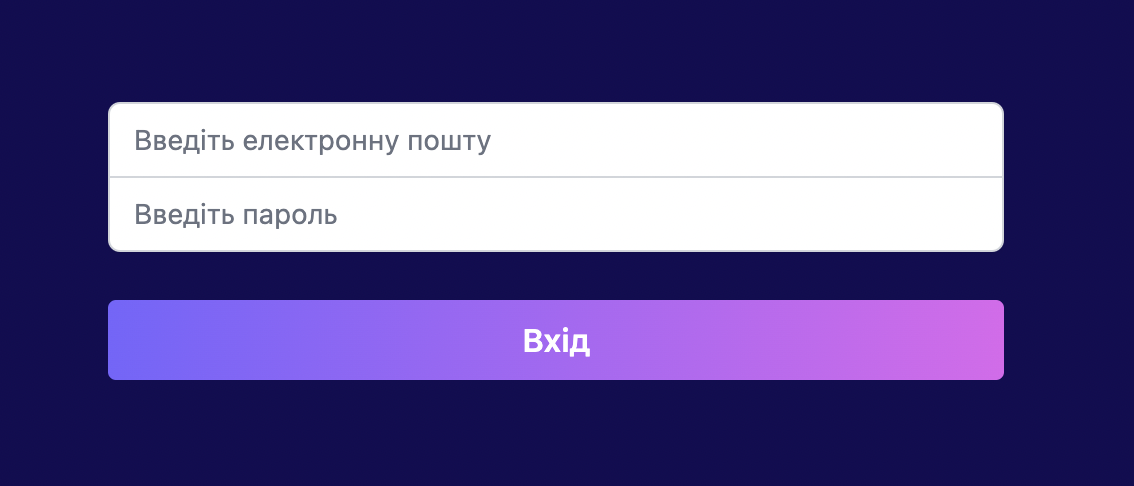


Рис. 3.8 Вхід до адміністратора

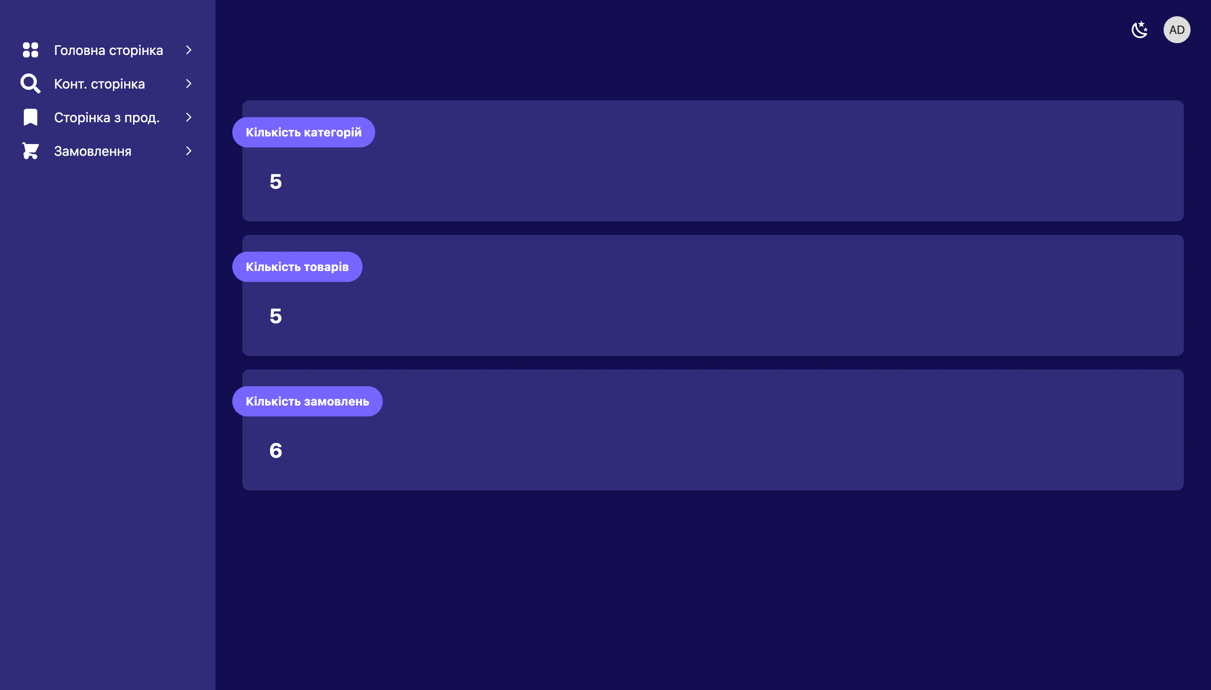


Рис. 3.9 Головна сторінка Адмін-панелі

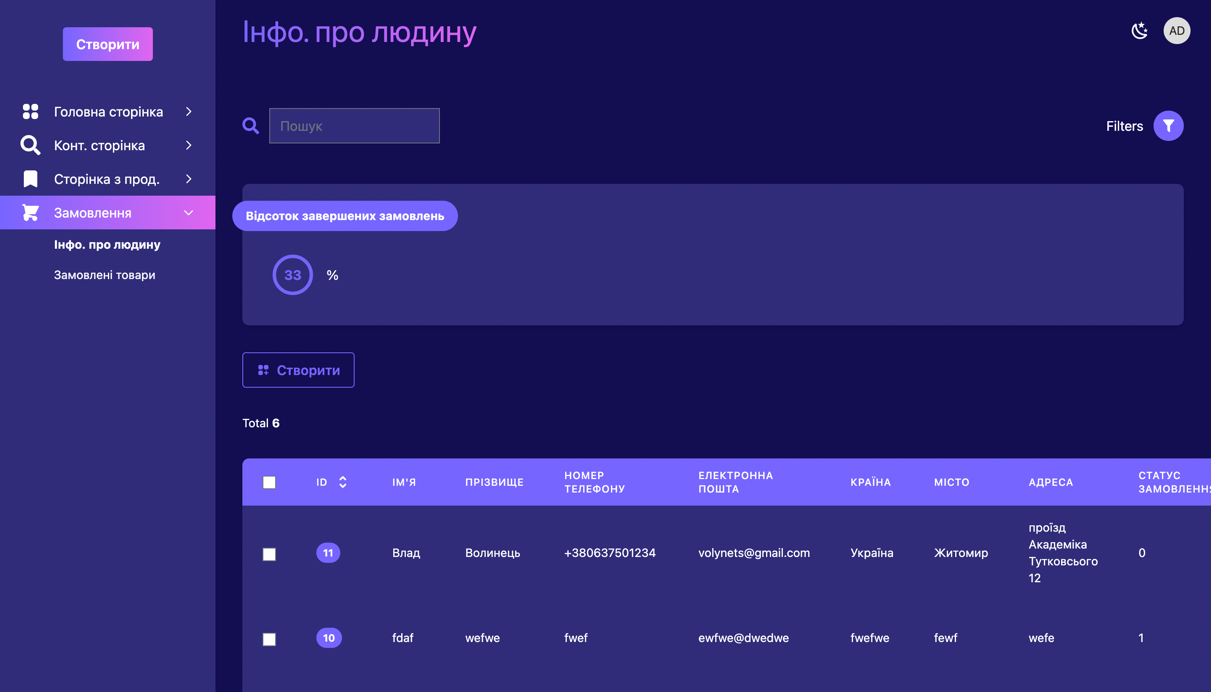


Рис. 3.10 Сторінка про замовлення

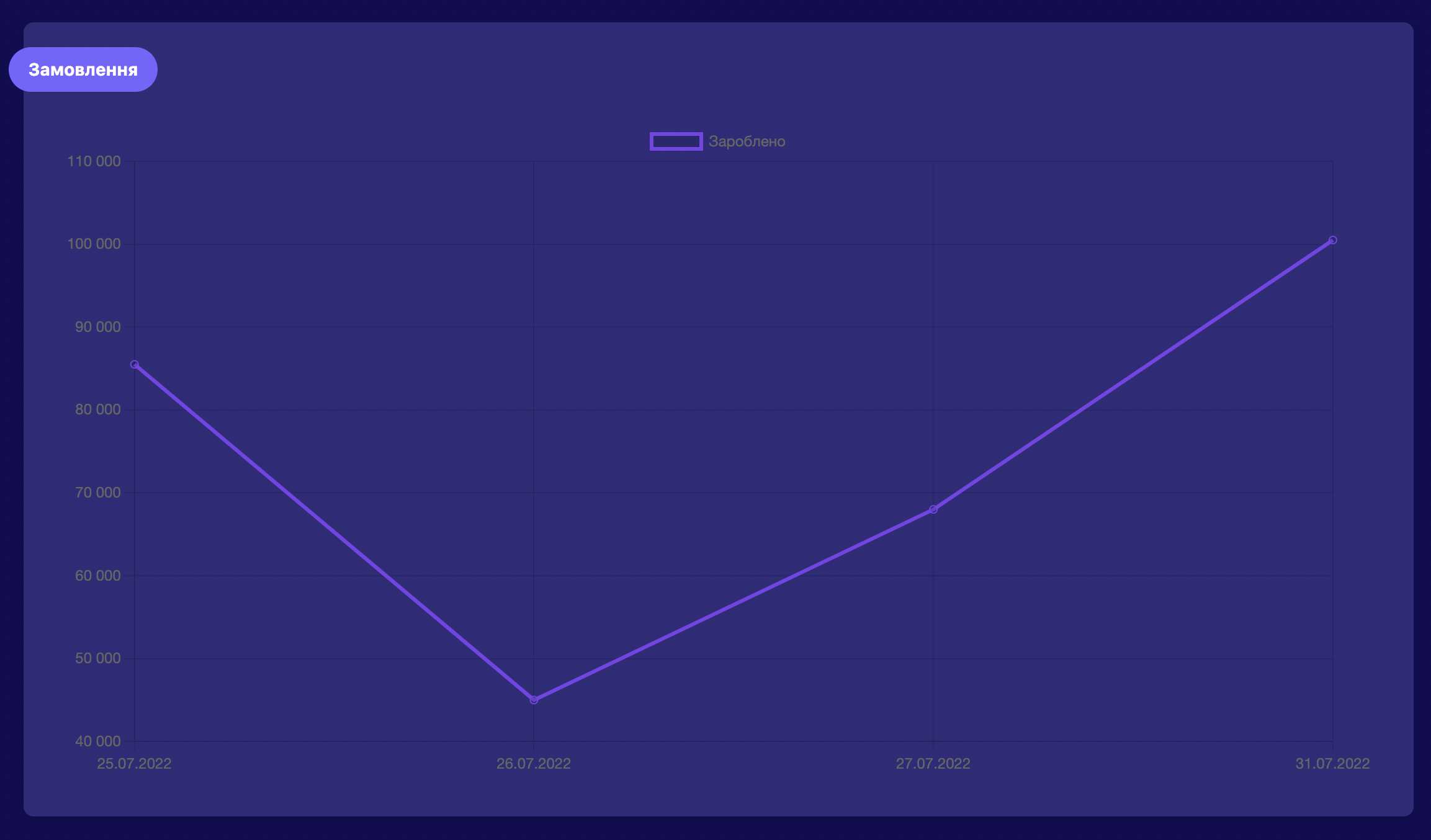


Рис. 3.11 Статистика зароблених грошей за кожний день

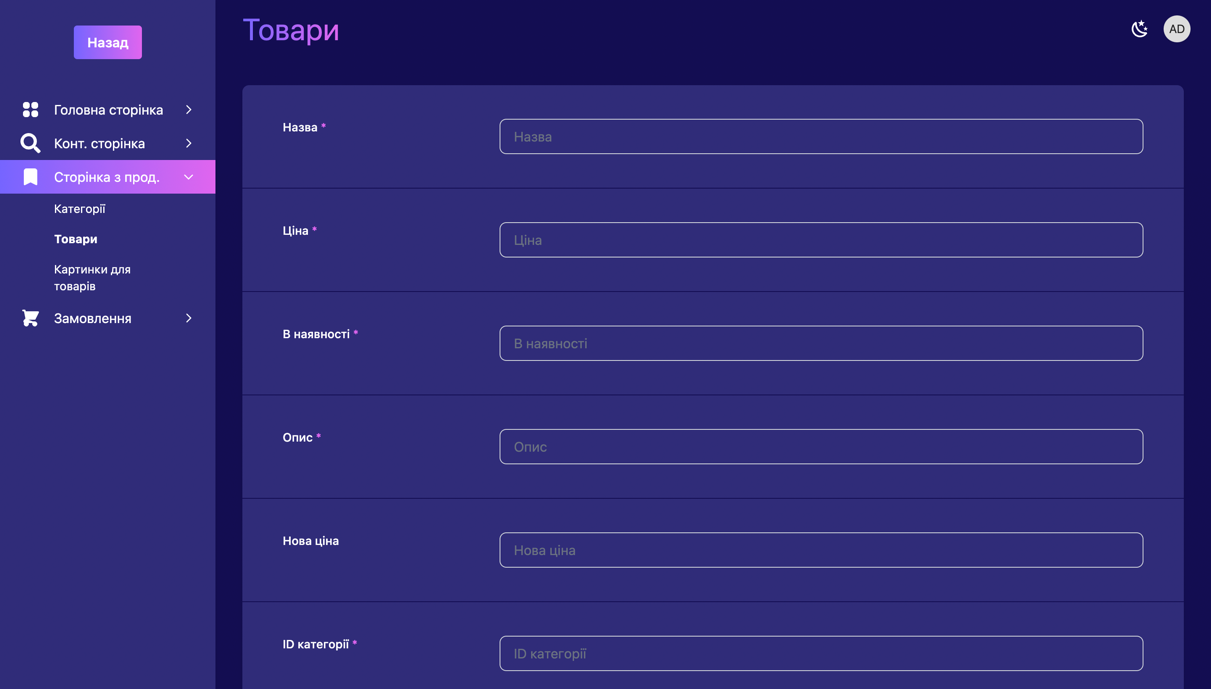


Рис. 3.12 Додавання нових товарів

# **Висновки до 3 розділу:**

Було розроблено інтерфейс сайту та продемонстровано його рисунками. Веб додаток був протестований на практиці та виявився легким у використанні із сучасним дизайном. Було реалізовано функції із базою даних.

## РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ

# **4.1 Розробка заходів захисту інформації в БД.**

В мене реалізовано тільки 2 ролі: незареєстровані користувачі та адміністартор. Звичайні користувачі можуть тільки переглядати товари, робити замовлення та відсилати форму зворотнього зв’язку. Тільки адміністратору призначається окремо доступ до команд CRUD.

У таблиці 4.1 наведено наведено матрицю доступу.

Доступні можливості: 1 – перегляд, 2 – додавання, 3 – редагування, 4 – видалення, 5 – робити завмовлення. 6 – відправка форми, 7 – весь доступ.

Таблиця 4.1

|  |  |  |
| --- | --- | --- |
| Матриця доступу | | |
| Роль  Таблиця | Користувач | Адміністратор |
| galleries | 1 | 7 |
| features | 1 | 7 |
| information | 1, 6 | 7 |
| categories | 1 | 7 |
| products | 1 | 7 |
| orders | 1,5 | 7 |

# **4.2 Налаштування параметрів роботи сервера**

Я використовував MAMP, тому для того, щоб увійти в phpMyAdmin потрібно було ввести пароль. Після чого, нам буде доступно перегляд усіх баз даних, які ми створювали.

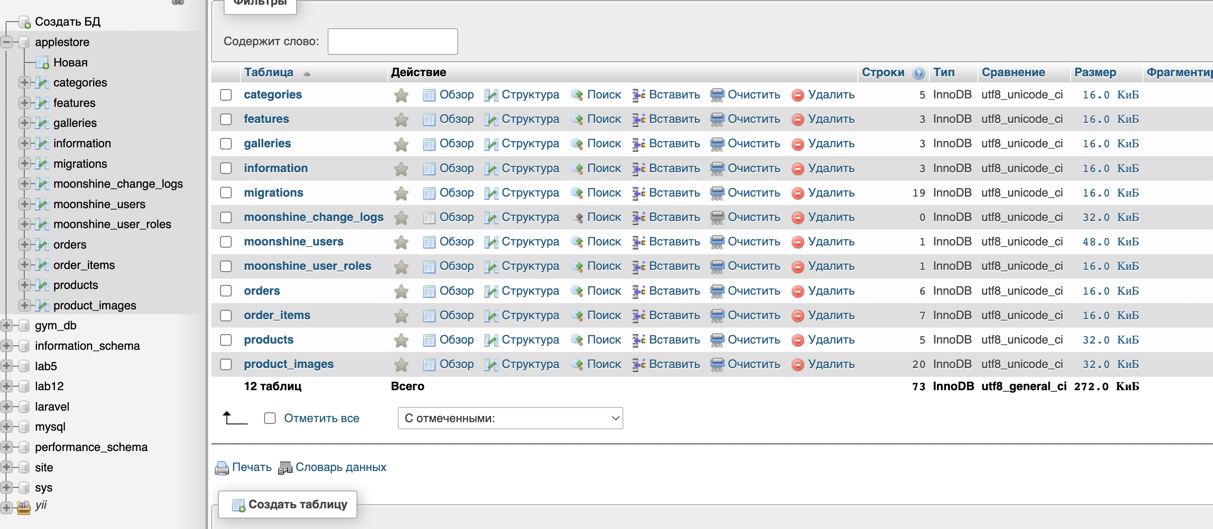


Рис. 4.1 Перегляд таблиць в базі даних

# **Висновки до 4 розділу:**

Для захисту інформації в базі даних було створено два рівні доступу до інформації: користувач, адміністратор. Адміністратор має найбільші привілеї. Була створена матриця доступу для відображення можливостей ролей.

## ВИСНОВКИ

В ході виконання курсового проекту було обрано тему для курсового проекту, проведено аналіз схожих проектів та побудова плану реалізації веб додатку.

Наступним кроком було обрано технології за допомогою яких і був розроблений додаток: фреймворк Laravel, мови програмування PHP та СУБД MySql.

Після обирання технологій, які використовувались для реалізації веб додатку були сформовані загальні алгоритми програми та продумані сторінки сайту із якими може взаємодіяти користувач. Також було реалізовано роль адміністратора. Було створено бази даних та таблиці до них.

Було реалізовано необхідні функції роботи з базою даних. Реалізовано статистики за допомогою яких, адміністратор може проводити аналіз замовлень, товарів, тощо.. Для надійності бази даних було реалізовано створення бекапів, що надає змогу мати копії даних і не втратити їх. Створений додаток вийшов простим та зручним.

В кінці була створена матриця доступу для візуалізації відмінностей між звичайними користувачами та адміністратором.

## СПИСОК ЛІТЕРАТУРИ

1. Бази даних – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85>.
2. Документація по Php – Режим доступу до ресурсу: <https://www.php.net/manual/ru/index.php>.
3. Документація по фреймворку Laravel - Режим доступу до ресурсу:

<https://uk.wikipedia.org/wiki/Laravel>.

1. Модель-вид-контролер - Режим доступу до ресурсу:

<https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%B2%D0%B8%D0%B4-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80>.

1. MySql документація - Режим доступу до ресурсу:

<https://wiki.gentoo.org/wiki/MySQL/Startup_Guide/ru>.

1. Youtube.com. 2022. MySQL CRUD – Режим доступу до ресурсу:

<https://www.youtube.com/watch?v=7xiv3tALliQ.>

1. Youtube.com. 2022. Основи phpMyAdmin - Режим доступу до ресурсу:

<https://www.youtube.com/watch?v=bpcs_o2WEQU>.

1. MAMP - Режим доступу до ресурсу:

<https://www.mamp.info/en/windows/>.

1. Lucidchart - Режим доступу до ресурсу:

<https://www.lucidchart.com/pages/>.

1. W3schools.com – Режим доступу до ресурсу: <https://www.w3schools.com/js/default.asp>.

# 

## ДОДАТКИ

# **Додаток А**

**ТЕХНІЧНЕ ЗАВДАННЯ**

1. **Загальне положення**
   1. **Найменування програмного засобу**

Повне найменування програмної системи: «База даних Інтернет магазину Apple» (надалі "програма"). Коротка назва програмної системи – «магазин».

* 1. **Призначення розробки та область застосування**

Програма призначена для відображення даних про товари Apple. Реалізована база днаих де зберігаються товари, замовлення та категорії. Система може бути покращена та впроваджена у роботу офлайн-магазину.

* 1. **Мета**

Програмна система «База даних Інтернет магазину» дозволить підвищити ефективність, швидкість та продуктивність роботи працівників.

* 1. **Найменування розробника та замовника**

Розробник даного продукту - студент групи ІПЗ-20-2 (надалі "розробник"). Замовник програмного продукту - кафедра інженерії програмного забезпечення Житомирського державного технологічного університету в межах виконання курсової з дисципліни «Бази даних» (надалі замовник).

1. **Підстава для розробки**

**2.1. Документ на підставі якого ведеться розробка**

Робота ведеться на підставі навчального плану за спеціальністю 121 – «Інженерія програмного забезпечення» наказу про закріплення тем курсових робіт за студентами

1. **Вимоги до програми**

**3.1. Вимоги до функціональних характеристик**

**3.1.1. Загальні вимоги**

- Можливість створення Баз Даних

- Створювати сутності в базі даних (Таблиці, Представлення, Процедури)

- Проводити вставку, оновлення, вибірку та видалення даних за умовами

- Забезпечувати базову підтримку підтримки цілісності даних

**3.1.2. Склад виконуваних функцій**

Розробити базу даних інтернет-магазину техніки Apple, що підтримує виконання наступних функцій:

1. Перегляд усіх товарів, згрупованих по категоріям

2. Якщо у користувача є дані для адміністратора – з'являється доступ до режиму адміністрування

3. Вставка, перегляд та видалення товарів та категорій.

4. Зберігання даних про замовлення користувачів

**3.1.3. Організація вхідних і вихідних даних**

Вхідними даними є дані про товари, користувачів, категорії та замовлення. Організація вхідних і вихідних даних повинна відповідати інформаційній структурі виконуваних з операцій. Введення оперативних даних повинно виконуватися з використанням діалогових екранних форм, побудованих на основі візуальних компонентів. Введення даних виконується в режимі online.

**3.1.4. Часові характеристики і розмір пам'яті, необхідної для роботи програми.**

Час реакції програми на дії користувача (маніпуляції з пристроями введення даних) не повинен перевищувати 0,25 с. Час виконання команд меню не більше 1 с. Відображення масивів даних за запитами не більше 3 хвилин. Доступність БД – 90% цілодобово. Операції з’єднання з БД не більше 1 хвилини. Обсяг оперативної пам'яті, необхідний для роботи програми не менше 1Гб.

**3.2. Вимоги до надійності**

**3.2.1. Вимоги до надійного функціонування**

Програма повинна нормально функціонувати при безперебійній роботі ПК. Доступність БД 90% при одночасному доступі 30 користувачів. При апаратних збоях, відновлення нормальної роботи програми повинне виконуватися після:

а) апаратні збої сервера - перезавантаження ОС сервера, запуск сервера БД (запуску резервного сервера, використання технологій RAID для збереження даних);

б)апаратні збої робочої станції – перезавантаження ОС ПК, запуск виконуваного файлу програми.

в) БД повинна повертатись в найближчий несуперечний стан – передбачити точки відновлення.

При збоях програмного забезпечення:

а) система повинна забезпечувати можливість відновлення даних та фіксацію і «відкат» транзакцій.

б) в системі має бути реалізована коректна обробка виняткових ситуацій.

**3.2.2. Контроль вхідної і вихідної інформації**

Для контролю коректності вхідної інформації та захисту від помилок оператора:

• Перевірка відповідності даних доменам інформаційних атрибутів;

• Використанням механізму авто заповнення та вибору за переліком для зв’язаних даних;

Визначені некоректні дії повинні супроводжуватись повідомленнями про помилку і блокуванням операцій оновлення даних. В системі має бути передбачений захист від загального блокування.

Для вихідної інформації передбачити:

• відсутність блокування даних через багато користувальницький доступ;

• постійне оновлення даних у відображених на екрані звітних формах.

• повідомлення про використання режиму монопольного доступу

супервайзерами БД.

**3.2.3. Час відновлення після відмови**

Час відновлення після відмови, не пов'язаною з роботою програми, повинен складатися із: часу перезапуску операційної системи; часу запуску сервера БД (підключення до сервера ) запуску виконуваного

файлу, часу повторного введення або зчитування даних.

**3.3. Вимоги до інформаційної і програмної сумісності.**

**3.3.1. Вимоги до методів рішення і мов програмування**

Вибір методів рішення здійснюється розробникам без узгодження з замовником. СУБД обирається у відповідності до характеристик визначених в п.3. З замовником погоджується вибір варіанту за вартісною ознакою.

**3.3.2. Вимоги до системи програмних засобів. Вимоги до програмного забезпечення сервера**

Вимоги до програмного забезпечення сервера: ОС – серверна версія ОС Windows не нижче Windows Server 2014 (або інший сервер), СУБД визначається встановлюється та налаштовується розробником (ліцензування СУБД виконується замовником). Вимоги до програмного забезпечення робочої станції: ОС - Windows не нижче Windows 7, або MacOS.

**3.4. Вимоги до складу і параметрів технічних засобів.**

Вимоги до складу технічних засобів:

1. Сервер:  
   • сервер у базовій конфігурації із підтримкою RAID;  
   • засоби для під’єднання до локальної мережі.  
   2. Робоча станція:  
   • ПК на базі процесорів Intel, AMD у стандартній комплектації;  
   • засоби для під’єднання до локальної мережі. засоби для під’єднання до локальної мережі.  
   • Pentium III з тактовою частотою 750 МГц;

• оперативна пам’ять – 2048Мбайт;  
• об’єм дискової пам’яті - 20Гбайт.

**4. Вимоги до програмної документації**

1. "Інструкція по інсталяції (встановленню) програми", складається з опису інсталяційного пакету, переліку етапів інсталяції та їх послідовності, опису встановлених програмних компонентів та режимів їх роботи після інсталяції. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться у 4 розділі.

2. Керівництво користувача" складається з опису послідовності завантаження програми, основних режимів роботи, опису основних екранних форм, переліку виняткових ситуацій та реакції користувача на них, порядку виконання завдань в системі. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 3 розділі курсової роботи.

3. "Керівництво адміністратора баз даних", складається з опису складу таблиць бази даних та правил доступів до них, опису та послідовності робіт по обслуговуванню бази даних (архівування, резервне копіювання, з вказівкою періодичності виконання та засобів, що для цього використовуються), порядку налаштування серверу та клієнтських додатків. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 2,3 та 4 розділах курсової роботи згідно плану.

# **Додаток Б**

Приклад коду для добавлення товарів у корзину:

class CartController extends Controller  
{  
 public function index() {  
  
 return view('cart.index');  
 }  
  
 public function store(Request $request) {  
 $product = Product::*where*('id', $request->id)->first();  
  
 if(!isset($\_COOKIE['cart\_id'])) setcookie('cart\_id', uniqid());  
 $cart\_id = $\_COOKIE['cart\_id'];  
 \Cart::*session*($cart\_id);  
  
 \Cart::*add*([  
 'id' => $product->id,  
 'title' => $product->title,  
 'price' => $product->new\_price ? $product->new\_price : $product->price,  
 'alias' => $product->alias,  
 'quantity' => (int) $request->qty,  
 'attributes' => [  
 'img' => isset($product->images[0]->img) ? $product->images[0]->img : 'no\_image.jpeg'  
 ]  
 ]);  
  
 return response()->json(\Cart::*getContent*());  
 }  
  
 public function destroy(Request $request) {  
 $cart\_id = $\_COOKIE['cart\_id'];  
 \Cart::*session*($cart\_id)->remove($request->id);  
 return redirect()->route('cartIndex')->with('success', 'Successfully !');;  
 }  
  
 public function update(Request $request) {  
 $cart\_id = $\_COOKIE['cart\_id'];  
 \Cart::*session*($cart\_id)->update(  
 $request->id,  
 [  
 'quantity' => [  
 'relative' => false,  
 'value' => $request->quantity  
 ],  
 ]  
  
 );  
 if($request->quantity == 0) {  
 \Cart::*session*($cart\_id)->remove($request->id);  
 return redirect()->route('cartIndex');  
 } else {  
 return redirect()->route('cartIndex');  
 }  
 }

public function clear() {  
 $cart\_id = $\_COOKIE['cart\_id'];  
 \Cart::*session*($cart\_id)->clear();  
  
  
 return redirect()->route('cartIndex');  
 }  
}

@extends('layouts.main')  
  
@section('title', 'Корзина')  
  
@section('custom\_css')  
 <link rel="stylesheet" type="text/css" href="/styles/cart.css">  
 <link rel="stylesheet" type="text/css" href="/styles/cart\_responsive.css">  
@endsection  
  
  
@section('content')  
  
 <!-- Home -->  
  
  
 <--Cart::session($\_COOKIE['cart\_id'])->getContent()  
 <div class="home">  
 <div class="home\_container">  
 <div class="home\_background" style="background-image:url(images/forCart/cart\_1.jpg)"></div>  
 </div>  
 </div>  
  
 <!-- Cart Info -->  
  
 <div class="cart\_info">  
 <div class="container">  
 <div class="row">  
 <div class="col">  
 <!-- Column Titles -->  
 <div class="cart\_info\_columns clearfix">  
 <div class="cart\_info\_col cart\_info\_col\_product">Продукт</div>  
 <div class="cart\_info\_col cart\_info\_col\_price">Ціна</div>  
 <div class="cart\_info\_col cart\_info\_col\_quantity">Кільксіть</div>  
 <div class="cart\_info\_col cart\_info\_col\_total">Всього</div>  
 </div>  
 </div>  
 </div>  
 <div class="row cart\_items\_row">  
 <div class="col">  
  
 @foreach(Cart::session($\_COOKIE['cart\_id'])->getContent() as $item)

<!-- Cart Item -->  
 <div class="cart\_item d-flex flex-lg-row flex-column align-items-lg-center align-items-start justify-content-start">  
 <!-- Name -->  
 <div class="cart\_item\_product d-flex flex-row align-items-center justify-content-start">  
 <div class="cart\_item\_image">  
 <div><img src="/images/forProducts/{{$item->attributes->img}}" alt=""></div>  
 </div>  
 <div class="cart\_item\_name\_container">  
 <div class="cart\_item\_name"><a href="{{route('showProduct', ['category', $item->alias])}}">{{$item->name}}</a></div>  
 <form action="{{ route('cartDestroy') }}" method="POST">  
 @csrf  
 <input type="hidden" value="{{ $item->id }}" name="id">  
 <button class="btn-dark cart\_item\_edit">Видалити продукт</button>  
 </form>  
 </div>  
 </div>  
 <!-- Price -->  
 <div class="cart\_item\_price">{{$item->price}} ₴</div>  
 <!-- Quantity -->  
 <form action="{{ route('cartUpdate') }}" method="POST">  
  
  
 <div class="cart\_item\_quantity">  
 <div class="product\_quantity\_container">  
 <div class="product\_quantity clearfix">  
 @csrf  
 <input type="hidden" name="id" value="{{ $item->id}}" >  
 <span>Кільк.</span>  
 <input type='text' name='quantity' id="quantity\_input" type="text" pattern="[0-9]\*" value="{{$item->quantity}}">  
 </div>  
 </div>  
 </div>  
 <button type="submit" style='width:100%;'class="btn-dark">Оновити</button>  
 </form>  
 <!-- Total -->  
 <div class="cart\_item\_total">{{$item->price \* $item->quantity}} ₴</div>  
 </div>  
 @endforeach  
  
 </div>  
 </div>  
  
 <div class="row row\_extra">  
 <div class="col-lg-6 offset-lg-2">  
 <div class="cart\_total">  
 <div class="section\_title">Кінцевий результат</div>  
 <div class="section\_subtitle">Ваш рахунок </div>  
 <div class="cart\_total\_container">

<ul>  
 <li class="d-flex flex-row align-items-center justify-content-start">  
 <div class="cart\_total\_title">Повна вартість</div>  
 <div class="cart\_total\_value ml-auto">{{ Cart::getTotal()}} ₴</div>  
 </li>  
 </ul>  
 </div>  
 @if(Cart::session($\_COOKIE['cart\_id'])->getTotalQuantity() == 0)  
  
 @else  
 <div class="button checkout\_button"><a href="{{route('productCheckout')}}">Оформити замовлення</a></div>  
 @endif  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 <!-- Icon Boxes -->  
  
  
  
  
@endsection  
  
@section('custom\_js')  
 <script src="/js/cart.js"></script>  
  
@endsection