

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Вебсистема "Drevo" для керування інтернет-магазином одягу»

Ступінь вищої освіти – бакалавр  
Спеціальність – 122 «Комп’ютерні науки»  
Освітня програма «Комп’ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: здобувач 4 курсу  
групи КН-21

Даниїл НОСУЛЬКО

Керівник: викладач кафедри інформаційного  
менеджменту, математики та  
статистики

Олег МУШИНСЬКИЙ

Засвідчую, що кваліфікаційна  
робота оформлена відповідно  
до ДСТУ 3008:2015 та не  
містить запозичень з праць  
інших авторів без відповідних  
посилань.

Здобувач: \_\_\_\_\_  
(підпис)

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»»

ЗАТВЕРДЖУЮ:  
завідувач кафедри  
комп'ютерних наук  
\_\_\_\_\_Сергій МІЧКІВСЬКИЙ  
«\_\_\_»\_\_\_20\_\_р

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Носулько Даниїл Вікторович

Тема роботи	Вебсистема "Drevo" для керування інтернет-магазином одягу
Номер та дата наказу про затвердження теми	№121-7 від 24 грудня 2024 року
Коротка постановка завдання	Розробити панель управління для обробки базових бізнес-процесів, таких як управління користувачами, замовленнями, товарами, фінансовий облік, аналітика.
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	1. Голей, Ю., & Дрік, І. (2023). Аналіз використання штучного інтелекту в системах управління бізнес-процесами: переваги та недоліки. <i>Challenges and Issues of Modern Science</i> , 1, 382-386. <a href="https://cims.fti.dp.ua/j/article/view/73">https://cims.fti.dp.ua/j/article/view/73</a> 2. Петрова І. Л., Мушинський О. Ю. Інноваційні підходи до управління командами у гібридному середовищі. <i>Теоретичні та прикладні питання економіки</i> . 2024. № 49. DOI <a href="https://doi.org/10.17721/tpre.2024.49.12">https://doi.org/10.17721/tpre.2024.49.12</a>
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має передбачити теоретичне, системотехнічне або експериментальне дослідження складного спеціалізованого завдання або практичної проблеми в галузі комп'ютерних наук, яке характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.

Дата видачі завдання 27 грудня 2024 р.

Керівник

Олег МУШИНСЬКИЙ

Здобувач освітнього ступеня бакалавра

Даниїл НОСУЛЬКО

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
<b>Підготовчий етап</b>			
1	Вибір напрямку дослідження	02.12.2024 р.	<i>виконано</i>
2	Формування теми та призначення керівника	16.12.2024 р.	<i>виконано</i>
3	Затвердження теми кваліфікаційної роботи	23.12.2024 р.	<i>виконано</i>
4	Затвердження завдання на кваліфікаційну роботу	27.12.2024 р.	<i>виконано</i>
<b>Основний етап</b>			
5	Розробка концепції кваліфікаційної роботи	13.01.2025 р.	<i>виконано</i>
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів	20.01.2025 р.	<i>виконано</i>
7	Затвердження розширеної постановки завдання. Підготовка та подання керівникові розділу 1 кваліфікаційної роботи	10.03.2025 р.	<i>виконано</i>
8	Проектування. Підготовка та подання керівникові розділу 2 кваліфікаційної роботи	24.03.2025 р.	<i>виконано</i>
9	Підготовка доповіді для експертизи стану виконання кваліфікаційної роботи (проміжний контроль)	31.03-04.04.2025 р.	<i>виконано</i>
10	Реалізація. Підготовка та подання керівникові розділу 3 кваліфікаційної роботи	07.04.2025 р.	<i>виконано</i>
11	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	14.04.2025 р.	<i>виконано</i>
12	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівникові доопрацьованого варіанту кваліфікаційної роботи	21.04.2025 р.	<i>виконано</i>
<b>Завершальний етап</b>			
13	Представлення рукопису для перевірки на плагіат	28.04-04.05.2025 р.	<i>виконано</i>
14	Підготовка презентації та доповіді на передзахист	05.05-11.05.2025 р.	<i>виконано</i>
15	Передзахист кваліфікаційної роботи	12.05-16.05.2025 р.	<i>виконано</i>
16	Доопрацювання роботи за результатами передзахисту	19.05-06.06.2025 р.	<i>виконано</i>
17	Експертиза роботи керівником та зовнішнім експертом	09.06-15.06.2025 р.	<i>виконано</i>
18	Доопрацювання доповіді та презентації для захисту	09.06-15.06.2025 р.	<i>виконано</i>
19	Захист кваліфікаційної роботи	16.06-22.06.2025 р.	<i>виконано</i>

Керівник

Олег МУШИНСЬКИЙ

Здобувач освітнього ступеня бакалавра

Даниїл НОСУЛЬКО

*Носулько Д. В. Вебсистема "Drevo" для керування інтернет-магазином одягу*

Пояснювальна записка кваліфікаційної роботи за спеціальністю 122 – Комп'ютерні науки (освітня програма – Комп'ютерні науки) СО Бакалавр. – ВНЗ «Університет економіки та права «КРОК», Навчально-науковий інститут інформаційних та комунікаційних технологій, кафедра комп'ютерних наук, Київ, 2024.

Описано розробку панелі управління для магазину одягу, яка спрощує управління замовленнями, товарами, підвищує ефективність персоналу, а також покращує безпеку даних.

Ключові слова: веб-застосунок, інтернет-магазин, панель управління, PHP, JS.

Рис. 17. Бібліограф.: 21 найм.

*Nosulko D. V. Web system "Drevo" for managing an online clothing store*

Explanatory note of qualification work in the specialty 122 - computer science (educational program - computer science) SO Bachelor.

The development of a control panel for a clothing store that simplifies the management of orders, goods, increases the efficiency of staff, and improves data safety.

Keywords: web support, online store, control panel, PHP, JS.

Rice. 17. Bibliographer: 21 hiring.

## ЗМІСТ

ВСТУП .....	6
РОЗДІЛ 1 ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ВЕБСИСТЕМИ "DREVO" ДЛЯ КЕРУВАННЯ ІНТЕРНЕТ-МАГАЗИНОМ ОДЯГУ " .....	8
1.1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.2 АНАЛІЗ ПОТЕНЦІЙНИХ КОНКУРЕНТНИХ ПЕРЕВАГ ПРОГРАМНОГО ПРОДУКТУ	11
1.3 ПОСТАНОВКА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ.....	12
Висновки до розділу 1 .....	12
РОЗДІЛ 2 ПРОЕКТУВАННЯ ПЛАТФОРМИ ІНТЕРНЕТ МАГАЗИНУ З ПАНЕЛЬЮ УПРАВЛІННЯ.....	14
2.1 Моделювання поведінки продукту.....	14
2.2 Моделювання структури продукту.....	16
2.3 ОПИС АРХІТЕКТУРИ ПРОДУКТУ .....	17
Висновок до розділу 2.....	18
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБСИСТЕМИ "DREVO" ДЛЯ КЕРУВАННЯ ІНТЕРНЕТ-МАГАЗИНОМ ОДЯГУ .....	20
3.1 ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ СИСТЕМИ.....	20
3.2 ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	27
3.3 ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	29
Висновок до розділу 3.....	34
ВИСНОВОК.....	36
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	37
ДОДАТОК А.....	39

## ВСТУП

**Актуальність теми.** У сучасних умовах автоматизація бізнес-процесів є ключовим фактором підвищення ефективності управління компанією. Панель управління DREVO спрямована на спрощення роботи з користувачами, замовленнями, товарами, фінансовим обліком та аналітикою. Вона забезпечує централізований контроль та моніторинг ключових аспектів діяльності підприємства, що дозволяє приймати виважені управлінські рішення.

**Мета дослідження.** Метою дослідження є розробка панелі управління DREVO для оптимізації бізнес-процесів, підвищення продуктивності та автоматизації операцій компанії.

### **Завдання дослідження:**

1. Проаналізувати існуючі системи управління бізнес-процесами.
2. Визначити основні функціональні вимоги до панелі управління DREVO.
3. Розробити архітектуру та інтерфейс панелі управління.
4. Реалізувати модулі управління користувачами, замовленнями, товарами, фінансами та аналітикою.
5. Провести тестування та оцінку ефективності впровадження.

**Об'єкт дослідження.** Процеси управління бізнес-операціями в компанії, що потребують автоматизації та оптимізації.

**Предмет дослідження.** Панель управління DREVO, її структура, функціональність та вплив на ефективність бізнес-процесів.

### **Методи дослідження:**

- аналіз літератури та існуючих рішень у сфері управління бізнес-процесами;
- проектування та моделювання системи;
- уекспериментальне тестування панелі управління DREVO;
- аналіз ефективності впровадження за допомогою статистичних методів.

**Практичне значення.** Розробка панелі управління DREVO дозволить підприємствам покращити контроль над бізнес-процесами, зменшити витрати на адміністративні завдання та підвищити рівень прийняття рішень. Впровадження цієї системи сприятиме зростанню конкурентоспроможності компанії та покращенню якості обслуговування клієнтів.

# РОЗДІЛ 1

## ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ВЕБСИСТЕМИ "DREVO" ДЛЯ КЕРУВАННЯ ІНТЕРНЕТ-МАГАЗИНОМ ОДЯГУ "

### 1.1 Опис предметної області

Предметна область розробки охоплює систему внутрішнього адміністрування бізнес-процесів компанії, яка включає в себе управління користувачами та управління замовленнями. Такий програмний продукт є критично важливим для компаній, що прагнуть до автоматизації, безпеки й ефективної взаємодії між підрозділами.

У традиційній організації без автоматизованих рішень часто виникають такі проблеми:

- відсутність централізованого обліку користувачів;
- ризики витоку конфіденційної інформації через неправильно налаштовані рівні доступу;
- високе навантаження на персонал під час ручної обробки замовлень;
- часті помилки при передачі даних між відділами;
- низька швидкість обробки клієнтських заявок, що негативно впливає на репутацію компанії.

Для подолання цих проблем розробляється система DREVO, яка надає інструменти для ефективного управління користувачами та замовленнями.

Управління користувачами — ключовий компонент будь-якої сучасної бізнес-системи. Панель адміністратора DREVO дозволяє створювати, редагувати та видаляти користувачів, визначати їхні ролі та рівні доступу, що є основою інформаційної безпеки й структурованості робочих процесів.

Основні функції:

- реєстрація та автентифікація користувачів;
- призначення ролей (адміністратор, менеджер, оператор тощо);
- налаштування прав доступу відповідно до ролей;

- управління персональними налаштуваннями;
- можливість блокування або відновлення доступу.

Завдяки гнучкій системі контролю доступу, компанія може ефективно розмежовувати повноваження співробітників, що зменшує ризик зловживань і підвищує надійність роботи.

Система також дозволяє автоматизувати процес обробки замовлень — від моменту подачі клієнтської заявки до її остаточного виконання. Така автоматизація значно знижує адміністративне навантаження на персонал і підвищує якість обслуговування клієнтів.

Основні функції:

- створення, редагування й видалення замовлень;
- відстеження статусів замовлень;
- можливість пошуку й фільтрації замовлень;

Впровадження такої системи скорочує час обробки заявок, зменшує кількість помилок і дозволяє компанії підтримувати високий рівень сервісу, що безпосередньо впливає на задоволеність клієнтів і конкурентоспроможність на ринку.

Для кращого розуміння логіки функціонування інтернет-магазину та панелі управління використовується моделювання процесів за допомогою діаграм діяльності. Activity Diagram дозволяє візуально описати, як взаємодіють користувачі з системою та які етапи проходять ті чи інші процеси.

Основні процеси

1. Процес оформлення замовлення (для менеджера):

- авторизація в системі;
- перегляд наявних товарів;
- створення нового замовлення;
- додавання товарів до замовлення;
- збереження замовлення.

2. Процес редагування замовлення:

- пошук замовлення;

- перегляд деталей;
  - внесення змін (кількість товару, статус, контактна інформація);
  - збереження змін.
3. Процес роботи з товарами:
- додавання нового товару;
  - редагування наявного товару;
  - зміна статусу (активний/неактивний);
  - позначення як "хіт продажу".
4. Процес роботи з відгуками:
- перегляд відгуків користувачів;
  - можливість фільтрації за товарами;
  - модерація (за потреби);
5. Процес адміністрування (для адміністратора):
- авторизація;
  - управління користувачами (додавання, редагування);
  - призначення ролей;
  - редагування дозволів (permissions);
  - зміна системних налаштувань.

На головній сторінці, клієнт шукає необхідний йому товар через фільтри. Щоб додати товар до кошику, клієнт переходить в деталі товару. Далі відкривається модальне вікно, де клієнт вписує усю необхідну інформацію для замовлення. Далі усім менеджерам відправляється повідомлення про нове замовлення. Він авторизується в панелі та переглядає деталі замовлення (товар, кількість, дані клієнта). Менеджер може ввести зміни до замовлення (коригування кількості товарів, зміна статусу, редагування контактних даних клієнта). Всі зміни зберігаються.

Діаграми діяльності (Activity Diagrams) дозволяють наочно уявити, як користувачі та адміністратори взаємодіють з системою на різних етапах, від оформлення замовлення до його обробки і доставки. Це дозволяє оптимізувати робочі процеси і забезпечити швидку та зручну обробку замовлень, що

позитивно впливає на користувацький досвід і ефективність роботи менеджерів.

## 1.2 Аналіз потенційних конкурентних переваг програмного продукту

Перед розробкою інформаційної системи "DREVO" важливо проаналізувати існуючі рішення, які виконують схожі функції (рис 1.1 та рис. 1.2). Це дозволить визначити їхні сильні та слабкі сторони, а також виділити унікальні можливості, які можуть бути реалізовані в нашій системі.

*Odoo* – це модульна ERP-система з відкритим кодом, яка дозволяє налаштовувати функціонал відповідно до потреб компанії. Вона містить модулі для CRM, складського обліку, управління фінансами, електронної комерції та багато іншого. Її перевагою є гнучкість та наявність безкоштовної версії, однак недоліками є складність впровадження та необхідність глибокого технічного налаштування для адаптації під конкретний бізнес.

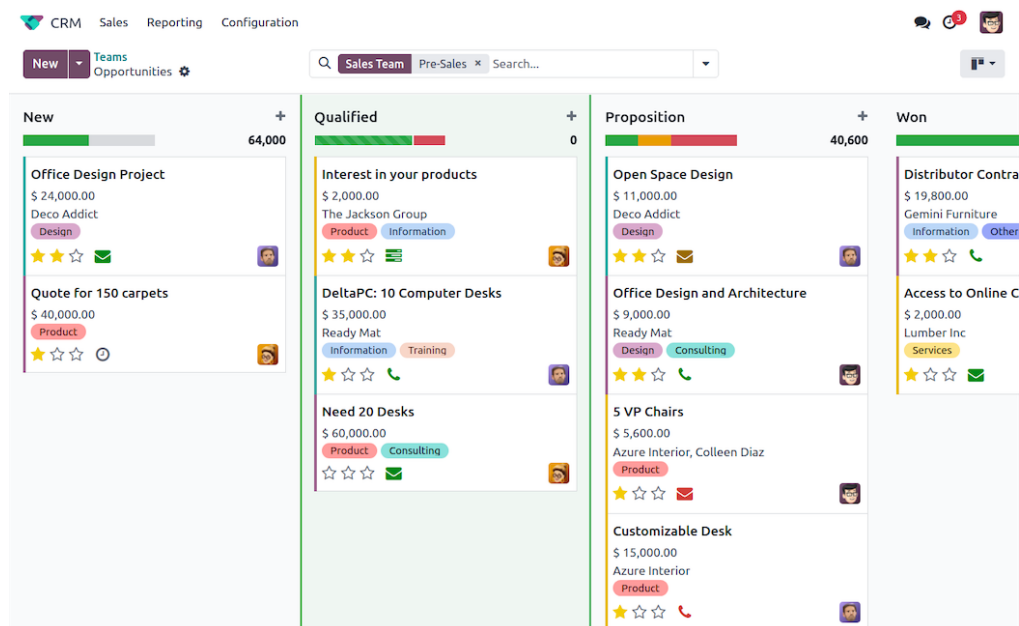
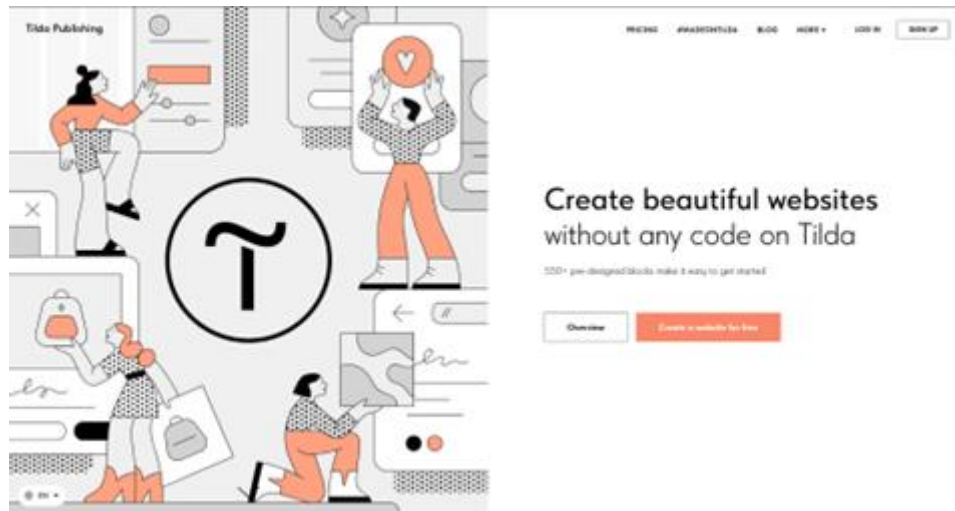


Рисунок 1.1 – Odoo

Джерело [4].



*Рисунок 1.2 – Tilda Publishing*

*Джерело [5].*

### **1.3 Постановка завдання на кваліфікаційну роботу**

Інформаційна система "DREVO" розробляється для автоматизації управління бізнесом, включаючи облік клієнтів, замовлень, складу та фінансів. Основна мета – створення ефективного інструменту, що спрощує роботу співробітників, мінімізує рутинні завдання та покращує контроль за бізнес-процесами.

Система буде складатися з веб-інтерфейсу для взаємодії користувачів та серверної частини для обробки запитів, збереження даних і виконання бізнес-логіки.

Основні завдання:

- управління замовленням: створення, оновлення, відстеження та обробка замовлень;
- управління товарами/послугами. додавання, редагування, видалення, категорії, фільтри.

### **Висновки до розділу 1**

У процесі аналізу існуючих рішень було виявлено, що більшість систем або є надто складними та дорогими для впровадження, або мають обмежений функціонал, що не дозволяє повною мірою автоматизувати бізнес-процеси. Це

підтверджує необхідність розробки інформаційної системи "DREVO", яка стане оптимальним рішенням для малого та середнього бізнесу.

Постановка задачі визначила ключові вимоги до системи, її функціональність та технологічний стек. Основний акцент зроблено на управлінні клієнтами, замовленнями, фінансами, що забезпечить комплексний підхід до автоматизації бізнесу.

Очікуваним результатом є створення ефективної, зручної та масштабованої інформаційної системи, яка допоможе підприємствам оптимізувати свої операційні процеси, зменшити ручну роботу та покращити контроль за фінансовими та товарами.

## РОЗДІЛ 2

### ПРОЕКТУВАННЯ ПЛАТФОРМИ ІНТЕРНЕТ МАГАЗИНУ З ПАНЕЛЬЮ УПРАВЛІННЯ

#### 2.1 Моделювання поведінки продукту

Моделювання поведінки продукту дозволяє заздалегідь визначити, як користувачі взаємодіятимуть з системою, та передбачити можливі сценарії її використання. Для цього використовувались діаграми активностей та сценарії використання (use case), що описують основні потоки дій користувачів як у публічній частині інтернет-магазину, так і в адміністративній панелі.

Поведінка користувача інтернет-магазину. Основними сценаріями поведінки користувача є:

- огляд каталогу товарів: користувач переглядає доступні товари, фільтрує їх за параметрами (категорія, ціна, рейтинг);
- перегляд детальної інформації про товар: після вибору товару користувач ознайомлюється з описом, характеристиками, умовами доставки, а також читає відгуки інших покупців;
- формування замовлення: додає товар у кошик, переходить до оформлення, заповнює персональні дані, надсилає замовлення;
- взаємодія з формами зворотного зв'язку: залишення відгуку або запит на зворотній дзвінок.

Поведінка співробітника компанії. В рамках адміністративної панелі були змодельовані наступні сценарії для менеджерів:

- перегляд списку замовлень, призначення себе відповідальним, редагування статусу;
- керування товарами та відгуками;
- редагування особистих даних;
- адміністратор;
- повний доступ до системних модулів: керування ролями, правами доступу, перегляд системної інформації;

- можливість створення, редагування та видалення користувачів;

Use case diagram - діаграма, що описує, який функціонал програмної системи, що розробляється, доступний кожній групі користувачів (рис 2.1 та рис 2.2).



Рисунок 2.1 – Use case diagram. Клієнт

Джерело: розроблено автором

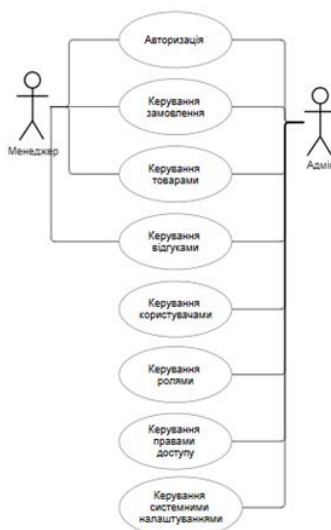


Рисунок 2.2 – Use case diagram. Менеджер, Адмін

Джерело: розроблено автором

## 2.2 Моделювання структури продукту

Моделювання даних є одним із ключових етапів розробки інформаційної системи. Воно передбачає визначення основних сутностей, їхніх властивостей і взаємозв'язків, що в сукупності формують логічну структуру бази даних майбутньої системи.

Діаграма класів (рис 2.3 та рис 2.4)) - статичне представлення структури моделі в UML. Відображає елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів.

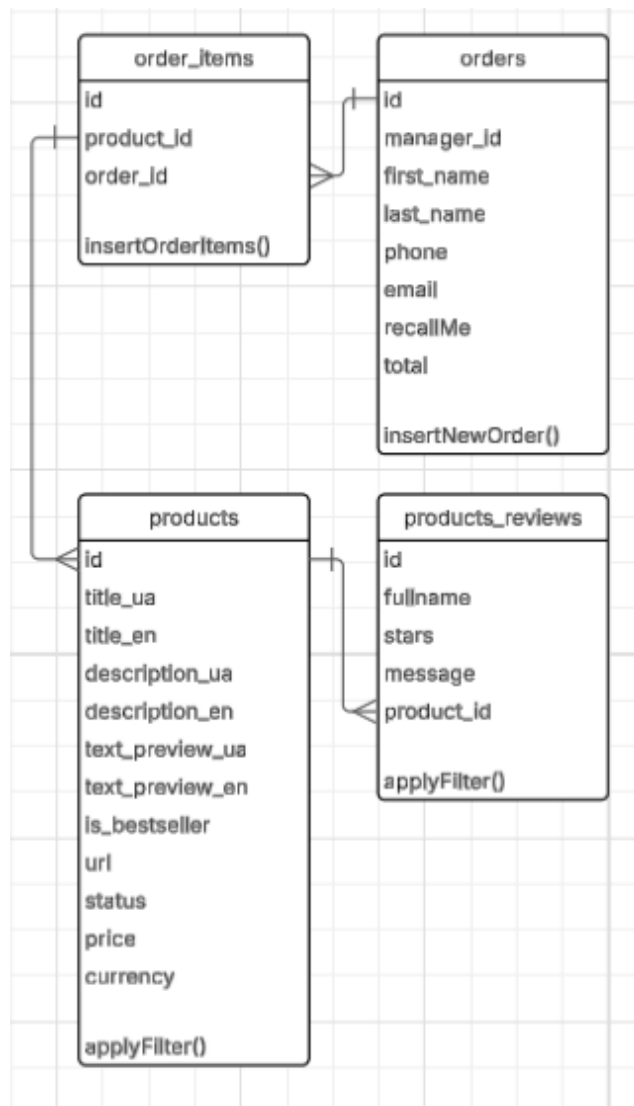


Рисунок 2.3 – Діаграма класів магазину.

Джерело: розроблено автором

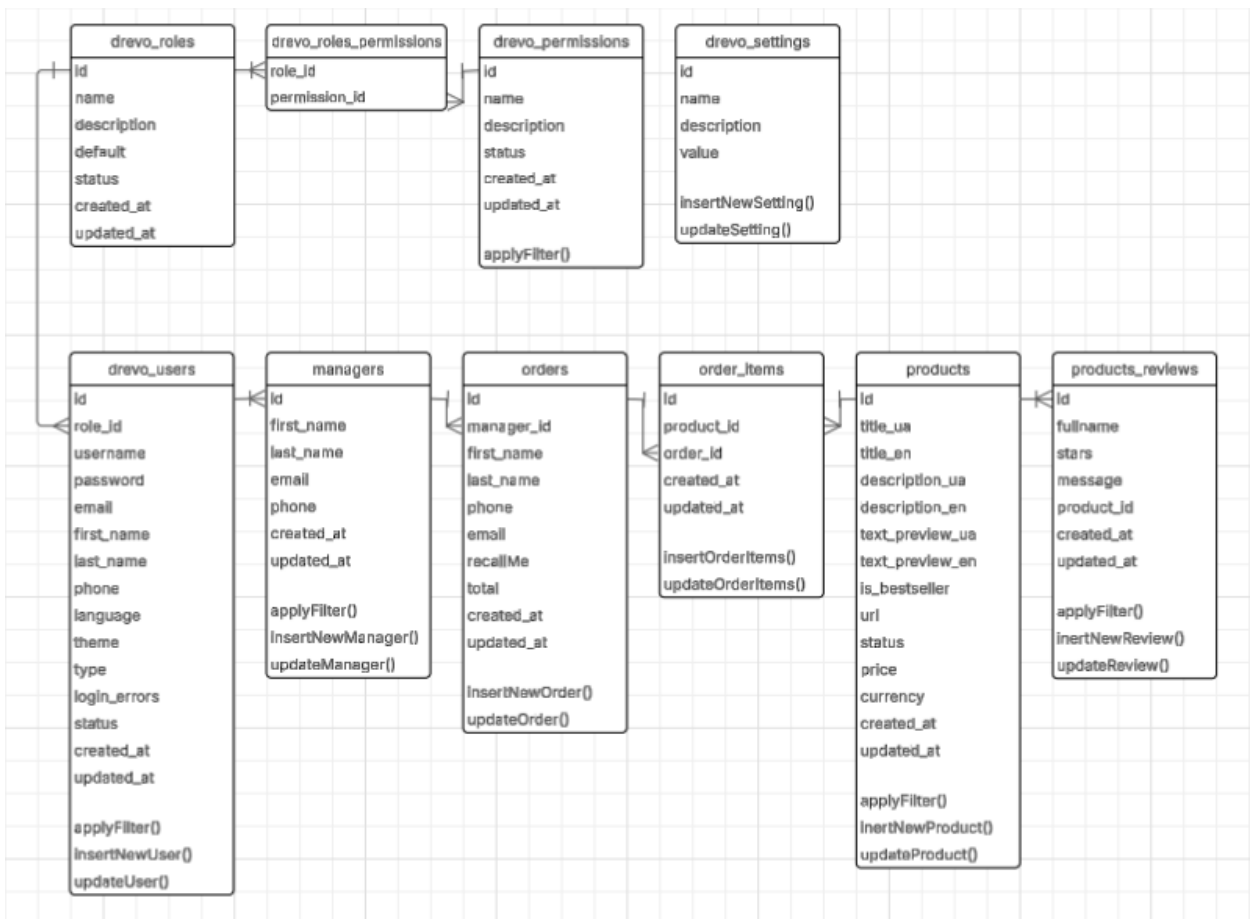


Рисунок 2.4 – Діаграма класів панелі управління

Джерело: розроблено автором

### 2.3 Опис архітектури продукту

Архітектура платформи інтернет-магазину побудована за принципом багаторівневої клієнт-серверної моделі з використанням MVC-підходу (Model-View-Controller), що забезпечує чітке розділення логіки даних, інтерфейсу та обробки запитів.

На клієнтському рівні реалізовано інтерфейс користувача за допомогою:

- Vue.js — для реактивних компонентів (форма фільтрації, швидке додавання товарів, повідомлення тощо);
- Bootstrap — для адаптивного та сучасного дизайну;
- jQuery — для простих взаємодій з DOM, подій, AJAX-запитів.

Цей рівень відповідає за зручну взаємодію користувача із системою, забезпечуючи високу швидкість відгуку та адаптивність на різних пристроях.

На серверній частині використовується CodeIgniter 4 — легкий та продуктивний PHP-фреймворк з підтримкою REST API. Його завдання:

- обробка HTTP-запитів;
- аутентифікація та авторизація користувачів;
- обробка бізнес-логіки (замовлення, товари, ролі тощо);
- робота з базою даних через ORM (Model).

Контролери взаємодіють із моделями для обробки даних, після чого повертають результати або в форматі JSON (для Vue) або у вигляді шаблонів (для класичних сторінок).

Зберігання даних здійснюється в реляційній базі даних MySQL. Структура побудована згідно з діаграмою “Сутність-зв’язок” і включає основні таблиці:

- users, roles, permissions — для керування доступом;
- products, categories, orders, invoices — для управління торговим процесом;
- reviews, settings — для розширеної функціональності.

Передбачено розширення функціональності через REST API, що дозволяє зручно працювати з мобільними додатками, сторонніми сервісами або інтеграцією з іншими системами.

## **Висновок до розділу 2**

Проектування структури та процесів для інтернет-магазину є важливим етапом у створенні ефективної та зручної платформи для користувачів і адміністраторів. Визначення чіткої структури клієнтської частини і панелі управління дозволяє забезпечити безперебійне функціонування інтерфейсу для кінцевих користувачів та ефективне адміністрування для менеджерів та адміністраторів.

Моделювання даних на основі ER-діаграм дає змогу логічно структурувати базу даних, забезпечуючи ефективне зберігання та обробку даних, таких як інформація про користувачів, товари, замовлення та відгуки.

Це є основою для побудови реляційної бази даних, яка підтримує стабільну роботу системи.

Таким чином, створення чіткої архітектури і налаштування процесів дозволяють досягти високої ефективності роботи як клієнтської частини інтернет-магазину, так і адміністративної панелі управління.

## РОЗДІЛ 3

# РЕАЛІЗАЦІЯ ВЕБСИСТЕМИ “DREVO” ДЛЯ КЕРУВАННЯ ІНТЕРНЕТ-МАГАЗИНОМ ОДЯГУ

### 3.1 Особливості реалізації системи

Для створення сучасної платформи інтернет-магазину з функціональною панеллю управління були обрані перевірені та ефективні засоби розробки, що дозволяють забезпечити високу якість програмного забезпечення, гнучкість у розробці та простоту підтримки. Основні технології, які використовувалися у проекті, наведено нижче.

CodeIgniter 4 — це потужний PHP-фреймворк, що реалізує принципи MVC (Model-View-Controller). Він забезпечує чітке розділення логіки застосунку, представлення та роботи з даними, що позитивно впливає на масштабованість і підтримку проекту. Основні переваги CI4:

- висока продуктивність: завдяки мінімалістичному ядру, проекти на CI4 працюють швидко навіть при великій кількості запитів;
- проста структура проекту: дозволяє швидко орієнтуватися у коді навіть новому розробнику;
- гнучка система маршрутизації: надає повний контроль над url-адресами;
- захист безпеки: вбудовані інструменти захисту від XSS, CSRF, SQL-ін'єкцій;
- підтримка міграцій: спрощує керування структурою бази даних на різних середовищах;
- модульність: можливість створення багаторазових модулів, які легко повторно використовувати в інших проектах.

Bootstrap — це найпопулярніший CSS-фреймворк для розробки адаптивного веб-інтерфейсу. У даному проекті він використовується для створення інтерфейсів як для фронтенду магазину, так і для панелі адміністратора. Переваги Bootstrap: Готові компоненти: кнопки, форми,

таблиці, модальні вікна, алерти та інше — все вже стилізовано і готово до використання;

- система сітки (grid): дозволяє швидко верстати адаптивний дизайн під різні розширення екранів;
- кросбраузерність: усі компоненти однаково виглядають у сучасних браузерах;
- гнучка кастомізація: можливість налаштувати змінні теми під власний стиль проєкту.

Vue.js — це прогресивний фреймворк JavaScript, який використовується для побудови інтерфейсів користувача з високим рівнем інтерактивності. У проєкті застосовується для створення динамічних компонентів панелі управління. Основні переваги:

- реактивність: миттєве оновлення dom при зміні даних;
- компонентний підхід: логічна організація коду, полегшує тестування та повторне використання;
- легка інтеграція: vue можна поступово додавати до існуючих сторінок без повної переробки;
- швидкість роботи: ефективний рендеринг навіть при великому обсязі даних;
- підтримка екосистеми: зручна інтеграція з Vuex (для управління станом), Vue Router (для маршрутизації).

jQuery — це легка JavaScript-бібліотека, яка дозволяє спростити роботу з HTML-документом, обробку подій, анімацію та AJAX-запити. У рамках даного проєкту jQuery використовується для реалізації деяких функціональних елементів, які не потребують повноцінного фреймворку.

Переваги:

- скорочення коду: завдяки лаконічному синтаксису можна досягти бажаного результату меншим обсягом коду;
- широка підтримка: безліч готових плагінів та рішень у спільноті;
- зворотна сумісність: підтримка старих браузерів;

- зручна робота з AJAX: реалізація динамічного завантаження контенту без перезавантаження сторінки.

Використання комбінації CodeIgniter 4, Bootstrap, Vue.js та jQuery забезпечує потужну основу для побудови сучасного веб-додатку. Цей стек має низку суттєвих переваг:

- масштабованість: архітектура на основі MVC (CodeIgniter 4) та компонентного підходу (Vue.js) дозволяє без особливих зусиль додавати нові модулі, ролі користувачів, функціональні блоки та розширення. Це робить платформу зручною для подальшого розвитку;

- швидкість розробки: завдяки наявності готових рішень і шаблонів у Bootstrap, Vue та CI4, розробник може суттєво зекономити час на створенні типових елементів інтерфейсу й логіки. Бібліотека jQuery також допомагає швидко реалізовувати прості інтерактивні сценарії;

- гарна підтримка користувача: візуальна частина системи є адаптивною та оптимізованою під мобільні пристрої, що забезпечує комфортну взаємодію незалежно від роздільної здатності екрана. Швидке реагування інтерфейсу досягається завдяки Vue.js та AJAX-запитам без перезавантаження сторінки;

- гнучкість: завдяки чіткому розділенню клієнтської та серверної частини можлива ефективна інтеграція традиційного серверного середовища PHP із сучасними фронтенд-інструментами. Це дає змогу реалізовувати як прості сторінки, так і складні односторінкові додатки (SPA).

Сховище даних побудовано на основі реляційної бази даних MySQL, яка зберігає структуровану інформацію про користувачів, ролі, замовлення, товари та інші сутності, необхідні для роботи системи.

На діаграмі (рис 3.1) представлено модель “сутність-зв’язок” (ER-діаграма), яка відображає логіку взаємозв’язків між основними таблицями:

- drevo\_users — містить інформацію про користувачів системи та пов’язана з таблицею drevo\_roles;

- drevo\_roles, drevo\_permissions, drevo\_roles\_permissions — реалізують систему ролей і прав доступу;
- orders, order\_items — реалізують систему замовлень із деталізацією по кожному товару;
- products, products\_reviews — таблиці для товарів та їх відгуків;
- managers — окрема таблиця для менеджерів, які можуть бути відповідальними за замовлення;
- drevo\_settings — зберігає глобальні параметри конфігурації.

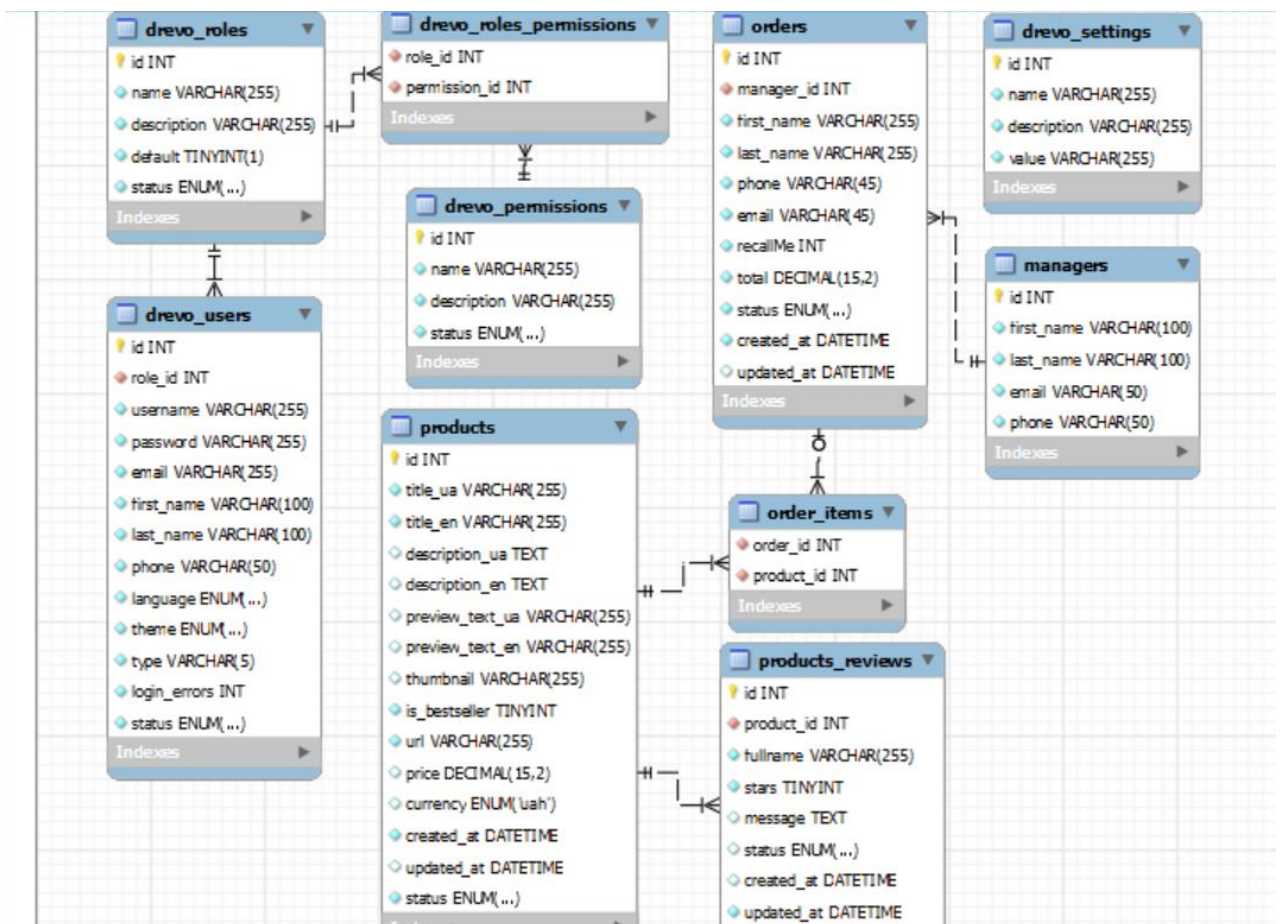


Рисунок 3.1 – Схема бази даних

Джерело: розроблено автором

Для взаємодії з базою даних використовується ORM (Object-Relational Mapping), яку надає CodeIgniter 4 через компоненти:

- model — для кожної таблиці створено окремий клас-модель, що інкапсулює доступ до даних;
- entity — для роботи з окремими об'єктами сутностей;
- query builder — для зручного складання запитів без написання SQL вручну.

ORM забезпечує:

- інкапсуляцію логіки доступу до даних;
- автоматичне оновлення і валідацію записів;
- захист від SQL-ін'єкцій;
- зручність розширення моделі при зміні структури бази даних.

«Сутність-зв'язок» діаграма (Entity-Relationship diagram) (рис 3.2) - ця модель дозволяє візуалізувати основні сутності, їх атрибути та взаємозв'язки між ними, що є основою для побудови реляційної бази даних.

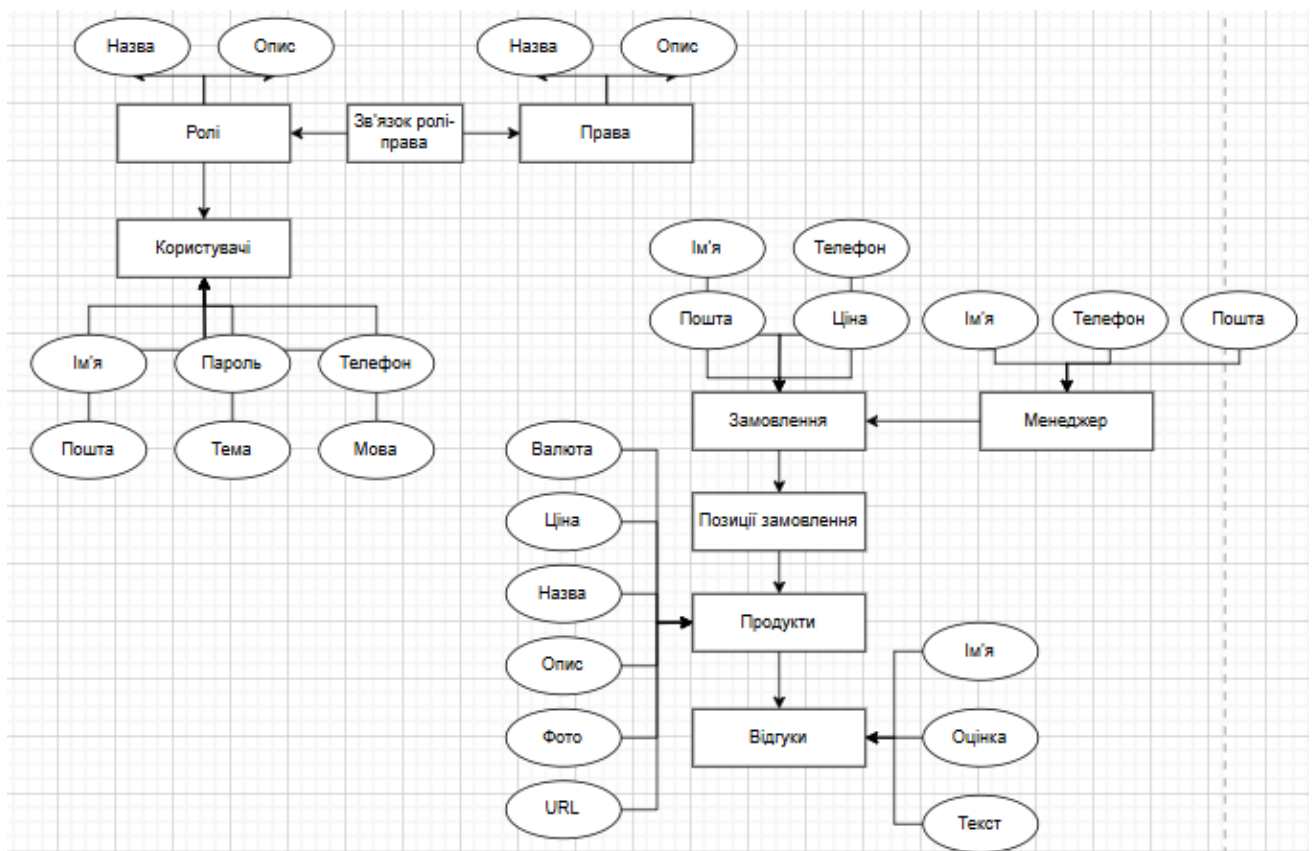


Рисунок 3.2 – Entity-Relationship діаграма

Джерело: розроблено автором

Основні сутності системи:

1. *Користувачі (drevo\_users)*. Ця таблиця містить інформацію про всіх користувачів системи - як клієнтів, так і адміністраторів. Основні поля: `username`, `email`, `password`, `role_id`, `language`, `status`, `created_at`. Кожен користувач має роль, яка визначає рівень доступу.
2. *Ролі (drevo\_roles)* Сутність містить назви ролей, наприклад: "Адміністратор", "Менеджер", "Користувач". Використовується для контролю доступу до функцій системи.
3. *Дозволи (drevo\_permissions)* Це список дозволів (`permissions`), які можуть бути призначені ролям. Зв'язок `drevo_roles_permissions` реалізує відношення "багато до багатьох" між ролями та дозволами.
4. *Товари (products)* Містить дані про товари, що продаються в інтернет-магазині. Основні атрибути: `title_ua`, `description_en`, `price`, `currency`, `status`, `is_bestseller`.
5. *Відгуки (products\_reviews)* Містять думки користувачів щодо товарів. Мають зв'язок з товарами через поле `product_id`.
6. *Замовлення (orders)* Зберігає інформацію про замовлення користувачів. Кожне замовлення пов'язане з менеджером, який його обробляє (`manager_id`).
7. *Елементи замовлення (order\_items)* Допоміжна таблиця, що реалізує зв'язок "багато до багатьох" між замовленнями та товарами.
8. *Менеджери (managers)* Окрема таблиця для зберігання інформації про менеджерів, що відповідають за обробку замовлень.
9. *Системні налаштування (drevo\_settings)* Містить ключові конфігураційні параметри системи, які можна змінювати через систему управління.

Конструювання системи є критично важливою фазою, у якій визначається архітектура усієї системи. На рис. 3.3 та рис. 3.4 показано діаграму

компонентів нашого застосунку. Він складається з 3 основних частин: клієнтська, серверна та база даних.

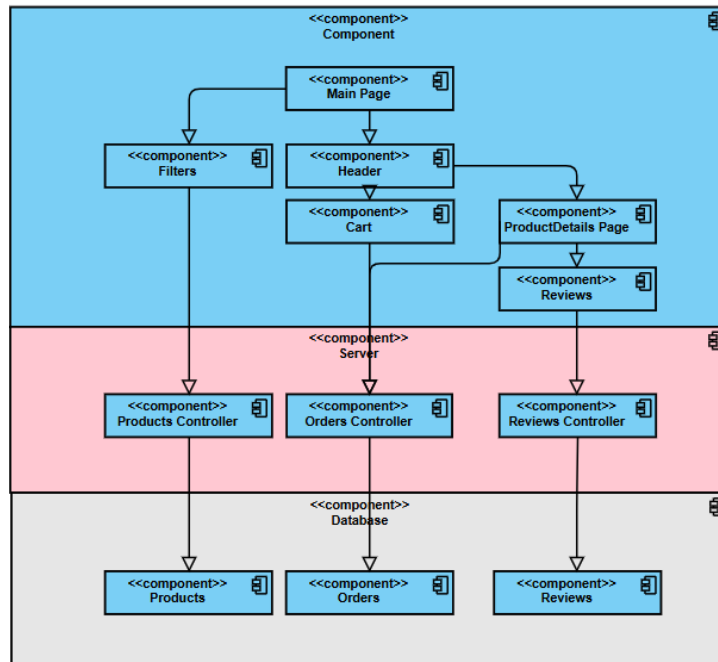


Рисунок 3.3 – Component Diagram. Магазин

Джерело: Автор.

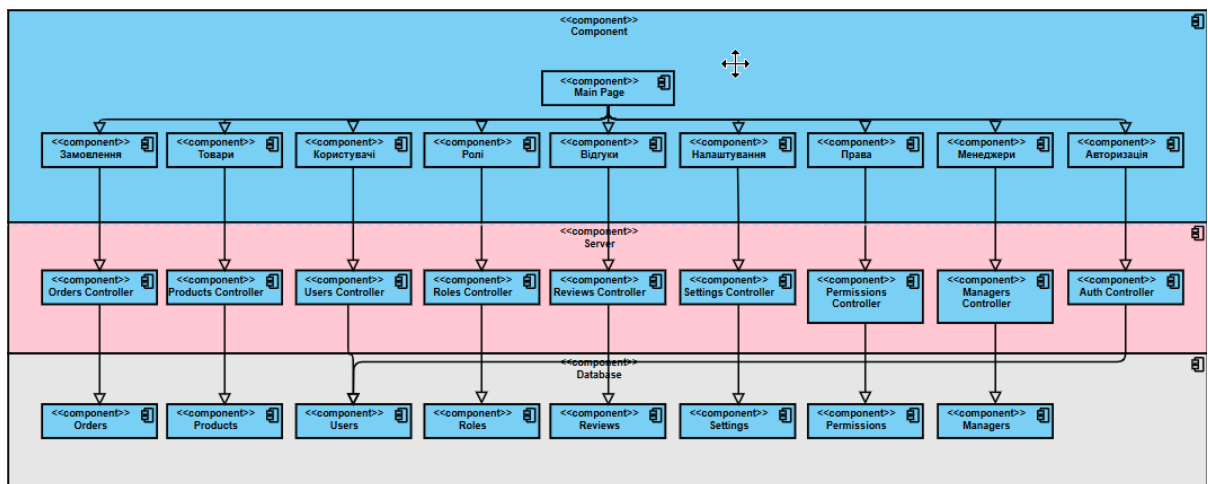


Рисунок 3.4 – Component Diagram. Веб-системи

Джерело: Автор.

Клієнтська частина (frontend) — представлена вгорі діаграми (синій блок). Ця частина включає основні сторінки застосунку, такі як:

- головна сторінка;

- модулі для роботи з товарами, замовленнями, користувачами;
- блоки налаштувань, авторизації, фільтрації, перегляду відгуків тощо.

Компоненти реалізовані з використанням таких технологій, як Vue.js, Bootstrap і jQuery. Вони відповідають за взаємодію з користувачем, обробку подій і відображення інтерфейсу.

Серверна частина (backend) — середній (рожевий) блок діаграми. Вона реалізована на основі фреймворку CodeIgniter 4 і складається з контролерів, кожен із яких обробляє запити від відповідного фронтенд-компонента:

OrdersController, ProductsController, UsersController тощо. Ці контролери відповідають за логіку обробки запитів, перевірку прав доступу та взаємодію з базою даних.

OrdersController, ProductsController, UsersController тощо. Ці контролери відповідають за логіку обробки запитів, перевірку прав доступу та взаємодію з базою даних.

База даних — нижній (сірий) блок діаграми. Вона містить основні таблиці, з якими працює система:

Orders, Products, Users, Roles, Permissions, Managers, Settings, Reviews тощо. База даних зберігає всю необхідну інформацію про користувачів, товари, замовлення та інші сутності.

Завдяки чіткому поділу на частини та використанню сучасних інструментів, система є модульною, масштабованою та легкою у супроводі. Кожен компонент має окрему відповідальність, що значно спрощує як розробку, так і тестування.

### **3.2 Тестування програмного продукту**

Тестування системи є обов'язковим етапом життєвого циклу розробки програмного забезпечення, який дозволяє виявити помилки, перевірити правильність роботи окремих компонентів, а також оцінити стабільність і продуктивність усього застосунку.

Основні види тестування, які були застосовані:

Модульне тестування (Unit testing). Застосовувалося для перевірки окремих контролерів серверної частини, таких як OrdersController, ProductsController, UsersController тощо. Було протестовано правильність:

- обробки запитів;
- виконання бізнес-логіки;
- валідації даних;
- взаємодії з моделями бази даних.

Використовувалися можливості вбудованого тестування у CodeIgniter 4 (PHPUnit) та бібліотека PHPUnit.

Інтеграційне тестування (Integration testing). Перевіряло взаємодію між клієнтською частиною, сервером і базою даних. Тестувалися основні сценарії, наприклад:

- додавання нового товару;
- оформлення замовлення;
- реєстрація та авторизація користувача;
- перегляд і додавання відгуків;
- зміна налаштувань.

Функціональне тестування. Перевіряло відповідність реалізованих функцій вимогам технічного завдання. Кожна функція проходила перевірку на коректність виконання в різних умовах (успішний сценарій, помилкові дані, відсутність підключення до сервера тощо).

Тестування інтерфейсу (UI testing). Проводилося вручну шляхом взаємодії з елементами веб-інтерфейсу. Перевірялась зручність, логіка переходів між сторінками, доступність основних кнопок, відповідність дизайну. Частково використовувалися автоматизовані інструменти, як-от Selenium.

Тестування безпеки. Основна увага була приділена:

- захисту від SQL-ін'єкцій;

- перевірки прав доступу (різні рівні доступу користувачів: адміністратори, менеджери, звичайні користувачі);
- безпеці обробки даних у формі (використання CSRF-захисту, валідації на стороні клієнта і сервера).

Тестування навантаження (Load testing). Було проведено базове тестування із симуляцією одночасного доступу кількох користувачів до основних функцій (перегляд товарів, оформлення замовлень) для оцінки швидкодії застосунку.

### 3.3 Використання програмного продукту

Панель управління (рис 3.5). Співробітники компанії мають окрему форму дозволу, де необхідно запровадити вхід та пароль. Після успішного входу вони потрапляють на головну сторінку із загальною інформацією про свою компанію.

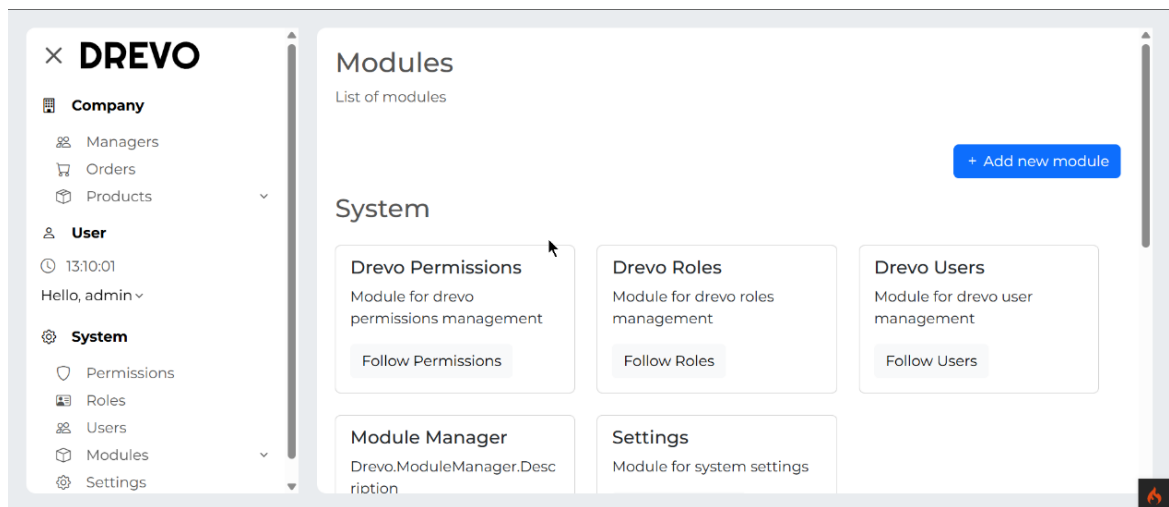


Рисунок 3.5 – Головна сторінка. Панель управління

Джерело: розроблено автором

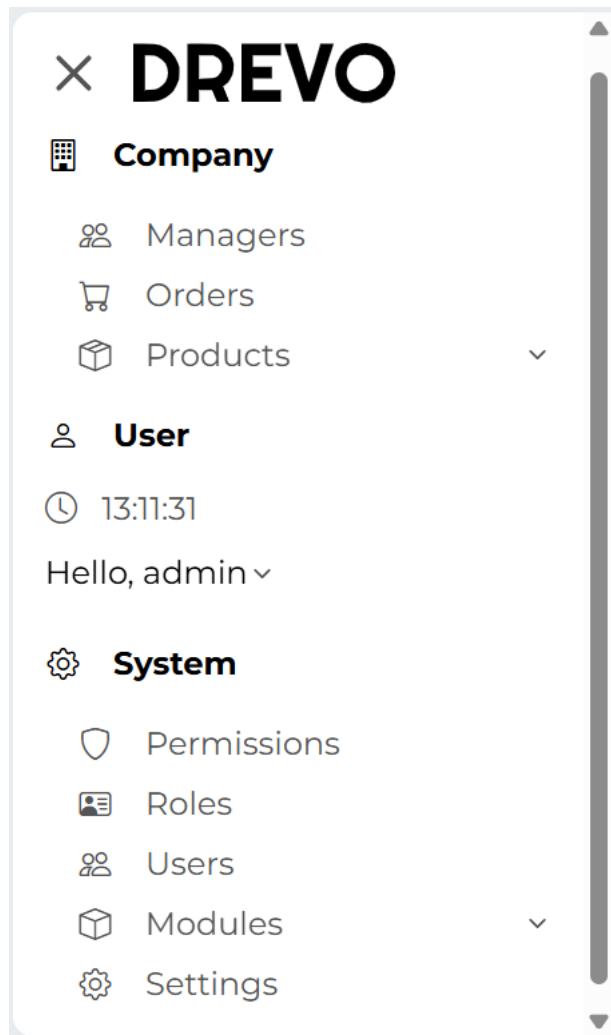
Адміністративна панель розділена на три групи модулів (рис 3.6):

1. Модулі Компанії - управління товарами, відгуками, замовленнями та менеджерами;

2. Модулі користувача - редагування особистої інформації працівника та вихід із облікового запису;

3. Системні модулі - управління ролями, правами доступу та загальною інформацією про систему.

4.



*Рисунок 3.6 – Групи модулів*

*Джерело: розроблено автором*

Наявність модулів залежить від ролі користувача (рис 3.7):

- менеджер доступний для модулів компанії та користувача;
- усі три групи модулів доступні адміністратору.

## Drevo Roles

Module for drevo roles management

[+ Add new role](#)











#	Name	Description	Default	Status	Actions
1	User		1	active	 
2	Manager		0	active	 
3	Admin		0	active	 
4	Developer		0	active	 
8	TEST	TEST	0	disabled	 

Рисунок 3.7 – Ролі

Джерело: розроблено автором

Кожен модуль містить таблицю з основною інформацією (рис 3.8). Під час перетину конкретного елемента здійснюється перехід на сторінку її редагування, де відображаються всі деталі та доступні дії: редагування, додавання та видалення.

### Editing the order

Recall me

Status: Processed

Client: Носунько

First name:

Last name:

Phone:

Email:

Manager:

T-shirt Market HOCKEY BEAR T-SHIRT white  
2070.00

Socks (Set) Dickies Valley Grove Socks Grey  
1199.00

[+ Add new item](#)

Total: 3269

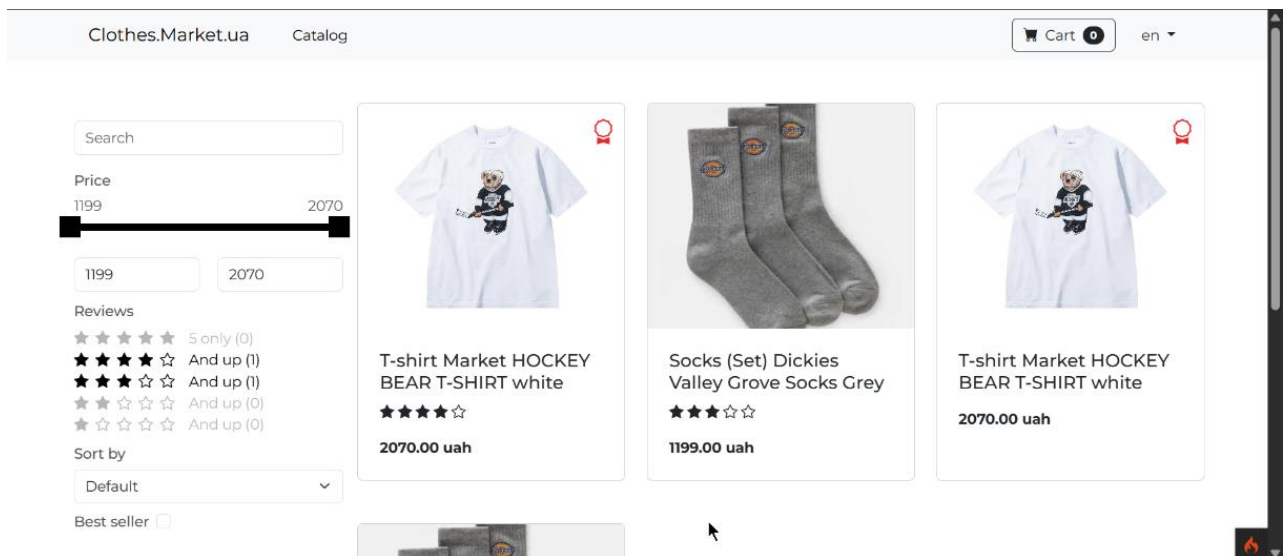
Drevo.Admin v1.0.0 created by Drevo

Рисунок 3.8 – Редагування замовлення

Джерело: розроблено автором

Архітектура системи побудована з можливістю масштабування: розробники можуть легко додавати нові модулі в міру зростання проекту. Інтерфейс інтуїтивно зрозумілий і дозволяє користувачам ефективно працювати з доступними функціями.

Інтернет-магазин. На першому переході на головну сторінку користувач бачить каталог товарів із фільтрами, які дозволяють фільтрувати товари за різними параметрами для більш зручного пошуку потрібного продукту.



*Рисунок 3.9 – Головна сторінка сайту*

*Джерело: розроблено автором*

Кожна карта продукту має посилання, за допомогою якого ви можете перейти на сторінку з детальною інформацією про продукт (рис 3.10): опис, характеристики, умови доставки та повернення. Нижче також перелік відгуків, де кожен користувач може залишити свій коментар через спеціальну форму.

Натиснувши кнопку «Купівля», товари додаються до кошика (рис 3.11). Переходячи до модального вікна "Кошик", користувач бачить усі додаткові товари і може почати розміщувати замовлення.

Щоб розмістити замовлення, користувач заповнює форму персональними даними (рис 3.12): ім'я, прізвище, електронна пошта, номер телефону, а також може вказувати, що менеджер повертається, щоб уточнити деталі.

## Футболка Market HOCKEY BEAR T-SHIRT white



2070.00 uah

- 1 +

Купити

## Опис

100% Cotton

## Доставка

- Доставка **безкоштовна для замовлень від 5000 грн.**
- Доставка здійснюється виключно **Новою Поштою.**
- **Замовлення відправляємо щодня**, в той самий день, коли воно було розміщене та оплачене, за однієї умови: до 17:00. Більш пізні замовлення відправимо вже наступного робочого дня.
- Також, онлайн-замовлення можна забрати самостійно з нашого фізичного магазину в **Києві, на вул. Саксаганського, 18**, розраховувались готівкою або картою на місці.

## Гарантія

- Ми продаємо виключно оригінальний товар і співпрацюємо напроям з брендами-виробниками та офіційними дилерами, тому на всі товари діє гарантія виробника.
- Якщо в процесі експлуатації виявився дефект - [зв'яжіться з нами](#) будь-яким зручним способом, ми організуємо заміну, гарантійний ремонт або повернемо кошти.
- Якщо куплений у нас товар з якихось причин не підійшов, його можна обміняти або повернути за стандартних умов: протягом 14 днів з моменту покупки, зі збереженням товарного вигляду та за наявності товарного чеку.
- Зробити це можна як [у нашому магазині на Саксаганського 18](#), так і Новою Поштою. Послуги доставки сплачує покупець, у гарантійних випадках - SHIFT.

## Новий відгук або коментар

Ім'я

Повідомлення

Оцінка ☆☆☆☆☆

Відправити

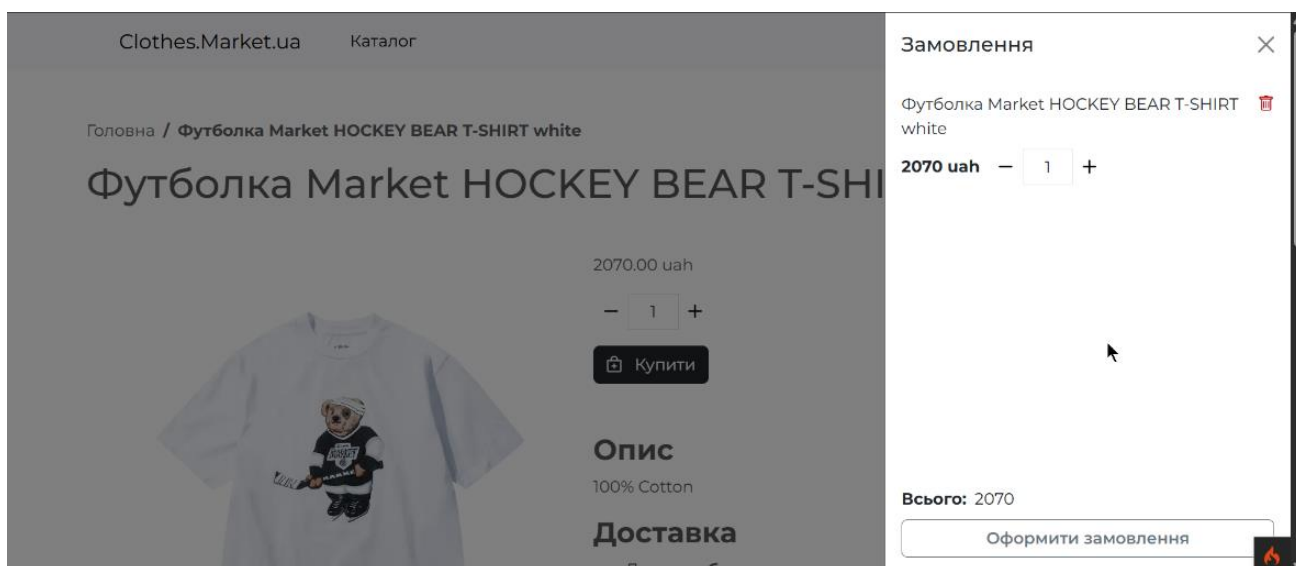
Носулько Даниїл Вікторович

★★★★☆

ТЕКСТ

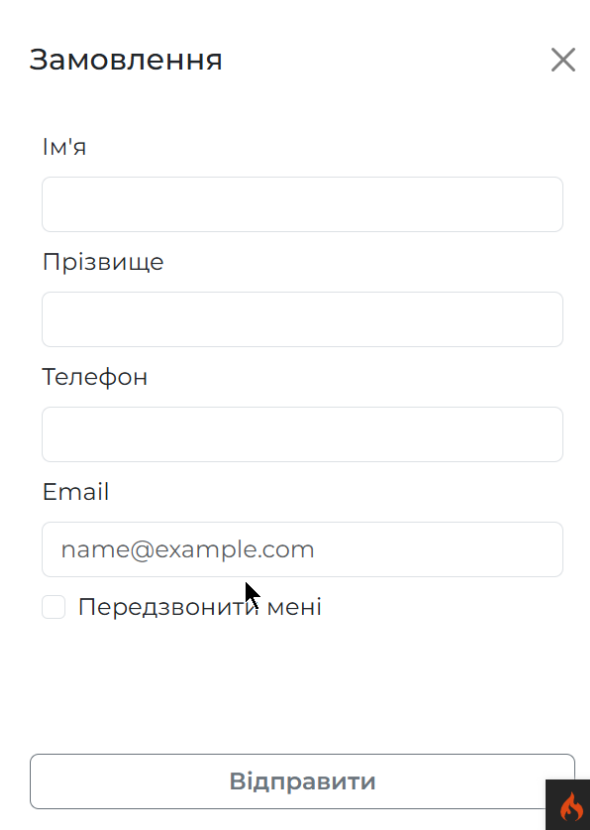
*Рисунок 3.10 – Деталі товару*

*Джерело: розроблено автором*



*Рисунок 3.11 – Кошик*

*Джерело: розроблено автором*



Замовлення X

Ім'я

Прізвище

Телефон

Email

name@example.com

Передзвонити мені

Відправити

*Рисунок 3.12 – Форма для заповнення інформації про клієнта*

*Джерело: розроблено автором*

Після натискання кнопки "Надіслати" замовлення передається на адміністративну панель, де менеджери бачать його у списку нових замовлень і можуть призначити себе відповідальним за його обробку.

### **Висновок до розділу 3**

У даному розділі було розглянуто процес конструювання системи інтернет-магазину з панеллю управління. Було обґрунтовано вибір засобів розробки: фреймворку CodeIgniter 4 як основи серверної частини, бібліотеки Bootstrap для швидкої та адаптивної верстки, Vue.js для реалізації інтерактивних елементів користувацького інтерфейсу та jQuery для спрощення роботи з DOM та Ajax.

Розроблена архітектура базується на класичному розділенні на клієнтську, серверну частину та базу даних, що забезпечує гнучкість і

масштабованість. Було також побудовано діаграму компонентів, що відображає взаємозв'язки між основними модулями системи.

Особливу увагу приділено моделюванню даних за допомогою діаграми «Сутність-зв'язок» (ERD), що дозволило сформувати логічну структуру бази даних для подальшої реалізації з використанням ORM-технологій, зокрема вбудованої в CI4 моделі Active Record.

Таким чином, результати даного розділу створюють ґрунт для практичної реалізації системи, забезпечуючи узгодженість між логікою застосунку, збереженням даних та інтерфейсною частиною.

## ВИСНОВОК

В результаті роботи над кваліфікаційною роботою було розроблено Вебсайт інтернет магазину з продажу офісного обладнання, з використанням багатокритеріальних фільтрів.

У першому розділі проаналізовано актуальність проблеми та розглянуто наявні аналоги виявлено їх переваги та недоліки. Визначено вхідні та вихідні дані, поставлено задачу та кроки для її реалізації.

У другому розділі було проведено проектування застосунку за допомогою діаграм UML. Сформовано функціональні та нефункціональні вимоги, діаграму прецедентів, для візуалізації використання системи. Створено діаграму класів та діаграму “сутність-зв’язок” для отримання структур для зберігання даних. Створено діаграму активності, яка затвердила потік використання системи з точки зору звичайного користувача.

У третьому розділі було описано особливості реалізації системи. Для цього було описано використання різних технологій для створення сучасного продукту, який може забезпечити зручність та ефективність для веб-сайту. Описано архітектуру програмного забезпечення, методи та схеми зберігання даних. Також було описано конструкцію та взаємодію різних частин системи за допомогою діаграми компонентів. Проведено тестування, та виправлено наявні в системі помилки. Наведено інструкцію користувачів по використанню веб-сайту інтернет-магазину з продажу офісного обладнання, з використанням багатокритеріальних фільтрів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Базан А.В. Проектування інформаційних систем: навч. посіб. — Харків: ХНУ імені В.Н. Каразіна, 2018. — 256 с.
2. ISO/IEC/IEEE 42010:2011. Systems and software engineering — Architecture description.
3. Офіційна документація CodeIgniter 4. URL: [https://codeigniter.com/user\\_guide](https://codeigniter.com/user_guide)
4. Сторінка компанії Odoo. URL: <https://www.odoo.com/> (Дата звернення: 10.03.2025).
5. Сторінка компанії Tilda. URL: <https://tilda.cc/> (Дата звернення: 29.04.2025).
6. Офіційна документація Vue.js 2. URL: <https://v2.vuejs.org/>
7. Bootstrap Documentation. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
8. jQuery API Documentation. URL: <https://api.jquery.com/>
9. Шрайбер Ю., Бредлі К. Веб-програмування з використанням PHP і MySQL. — Київ: Діалектика, 2020. — 448 с.
10. Freeman E., Freeman E. Head First Design Patterns. — O'Reilly Media, 2020. — 694 p.
11. MDN Web Docs — Mozilla Developer Network. URL: <https://developer.mozilla.org/>
12. Stack Overflow. URL: <https://stackoverflow.com/>
13. Sommerville I. Software Engineering (10th Edition). — Pearson Education Limited, 2015. — 792 p.
14. Філаретова Л.М. Інформаційні системи та технології в економіці: навч. посібник. — Київ: Центр учбової літератури, 2019. — 312 с.
15. Офіційна документація MySQL. URL: <https://dev.mysql.com/doc/>
16. Офіційна документація Git. URL: <https://git-scm.com/doc>

17. Nuseibeh B., Easterbrook S. Requirements Engineering: A Roadmap. — Proceedings of the Conference on the Future of Software Engineering, 2000.
18. Макаров С.В. Проектування баз даних: навч. посіб. — Харків: НТУ "ХПІ", 2021. — 248 с.
19. Vue Currency Input Documentation. URL: <https://vue-currency-input.netlify.app/>
20. Axios Documentation. URL: <https://axios-http.com/docs/intro>
21. RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. URL: <https://datatracker.ietf.org/doc/html/rfc7231>
22. Шевченко А.М. Безпека інформаційних технологій: навч. посіб. — Львів: Видавництво Львівської політехніки, 2020. — 280 с.

## ЛІСТИНГ КОДУ

```

<?php
namespace Modules\ModuleConstructor\Controllers;

use CodeIgniter\Database\Config;
use Modules\Drevo\Controllers\Drevo;
use Twig\Environment;
use Twig\Loader\FilesystemLoader;

class Ajax extends Drevo
{
    public FilesystemLoader $loader;
    public Environment $environment;

    public function __construct()
    {
        $this->loader = new FilesystemLoader(APPPATH . 'Modules/ModuleConstructor/Templates');
        $this->environment = new Environment($this->loader);
    }

    public function create()
    {
        $post = $this->request->getJSON(true);

        $this->createTable($post);
        $this->generateFilesByTemplates($post);

        $this->updateGlobalRoutes($post);
        $this->generateCommonModuleInfo($post);

        return $this->respond([
            'status' => true,
        ], 200);
    }

    private function generateFilesByTemplates($post) {
        $tableName = $post['table']['name'];
        $fields = $post['fields'];

        $modulePath = str_replace(DIRECTORY_SEPARATOR, '/', APPPATH . trim($post['module']['path'], '/'));
        $className = $post['module']['className'];
        $templatePath = APPPATH . 'Modules/ModuleConstructor/Templates';

        $folders = [
            'controllers' => [
                'Ajax.twig' => 'Ajax.php',
                'View.twig' => "{$className}.php"
            ],
            'views' => [
                'editForm.twig' => 'forms/editForm.php',
                'newForm.twig' => 'forms/newForm.php',
                'list.twig' => 'lists/list.php',
                'menu.twig' => 'menu.php',
            ]
        ];

        foreach ($folders as $folder => $files) {
            $targetDir = "{$modulePath}/$folder";

            if (!is_dir($targetDir)) {
                mkdir($targetDir, 0777, true);
            }

            foreach ($files as $template => $target) {
                $templateFile = "{$templatePath}/$folder/$template";

                if (file_exists($templateFile)) {
                    $content = $this->environment->render("{$folder}/{$template}", [
                        'module' => $post['module'],
                        'fields' => array_values($post['fields']),
                        'table' => $post['table']
                    ]);

                    $targetPath = "{$targetDir}/$target";

                    if (!is_dir(dirname($targetPath))) {
                        mkdir(dirname($targetPath), 0777, true);
                    }

                    file_put_contents($targetPath, $content);
                }
            }
        }

        $modelTemplate = "{$templatePath}/Model.twig";
        if (file_exists($modelTemplate)) {
            $modelContent = $this->environment->render('Model.twig', [
                'module' => $className,
                'table' => $tableName,
                'fields' => array_keys($fields),
            ]);

            $modelDir = "{$modulePath}/Models";
            if (!is_dir($modelDir)) mkdir($modelDir, 0777, true);

            file_put_contents("{$modelDir}/{$className}Model.php", $modelContent);
        }
    }

    private function getFieldFormat($field) {
        switch ($field['type']) {
            case 'INT(11)':
            case 'FLOAT':
                return 'number';

            case 'VARCHAR(255)':
            case 'TEXT':
                return 'string';

            case 'DATE':
            case 'DATETIME':
                return 'date';
        }
    }
}

```

Рисунок А.1 – Генерація шаблонів контролерів, представлень і моделей у

CodeIgniter

Джерело: розроблено автором

```

<?php

namespace Modules\Drevo\Controllers;

use CodeIgniter\API\ResponseTrait;
use CodeIgniter\Email\Email;
use CodeIgniter\HTTP\RedirectResponse;
use Config\Database;
use Config\Drevo as DrevoConfig;
use Config\Services;
use Modules\Drevo\Controllers\Drevo as DrevoController;

class Ajax extends DrevoController
{
    use ResponseTrait;

    public DrevoConfig $drevoConfig;
    public Email $email;

    protected $folder_directory = "Modules\\Drevo\\Views\\";

    public function __construct()
    {
        parent::__construct();

        $this->drevoConfig = config('Drevo');
        $this->email = Services::email();
    }

    public function login(): RedirectResponse
    {
        $validation = Services::validation();
        $db = Database::connect();

        if ($this->auth->user()) {
            return redirect()->to('/drevo/dashboard');
        }

        $data['header']['title'] = 'Login';
        $post = $this->request->getPost();

        $validation->setRules([
            'username' => 'required',
            'password' => 'required',
        ]);

        if (!$validation->run($post)) {
            return redirect()->to('/drevo/login');
        }

        $foundUser = $this->auth->auth($post['username'], $post['password']);

        if ($foundUser) {
            if ($this->auth->checkPermission('Drevo.Login', $foundUser['role_id'])) {
                session()->set(array('id' => $foundUser['id'], 'username' => $foundUser['username']));

                $userDetails = [
                    'id' => $foundUser['id'],
                    'username' => $foundUser['username'],
                    'role_id' => $foundUser['role_id'],
                    'language' => $foundUser['language'],
                ];
                $savedUser = $this->auth->user($userDetails);

                return redirect()->to('/drevo/dashboard');
            }

            return redirect()->to('/drevo/login');
        }

        return redirect()->to('/drevo/login');
    }
}

```

*Рисунок А.2 – Реалізація функції автентифікації користувача у CodeIgniter із перевіркою дозволів доступу  
Джерело: розроблено автором*