

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**
Факультет інформаційних технологій

ПОГОДЖЕНО
Декан факультету

**ДОПУСКАЄТЬСЯ ДО
ЗАХИСТУ**
Завідувач кафедри комп'ютерних
наук

_____ **Ігор БОЛБОТ**
(підпис)

_____ **Белла ГОЛУБ**
(підпис)

« ____ » _____ 2026 р.

« ____ » _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**на тему: «Інформаційна система обліку замовлень товарів в інтернет-
магазині»**

Спеціальність «122 Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Гарант освітньої програми
д.е.н, професор

_____ **Роман РУДЕНСЬКИЙ**
(підпис)

**Керівник бакалаврської
кваліфікаційної роботи**
старший викладач

_____ **Юрій МІЛОВІДОВ**
(підпис)

Виконав

_____ **Віталій КОНОНЧУК**
(підпис)

Київ-2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій**

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук

к.т.н., доцент _____ Белла ГОЛУБ
(підпис)

«___» _____ 2026 р.

**ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ**

Конончуку Віталію Сергійовичу

(прізвище, ім'я, по батькові)

Спеціальність «122 Комп'ютерні науки»
Освітньо-професійна програма «Комп'ютерні науки»
Тема бакалаврської кваліфікаційної роботи

«Інформаційна система обліку замовлень товарів в інтернет-магазині»

затверджена наказом від «6» __ січня __ 2025 р. № 46 «С»

Термін подання завершеної роботи на кафедру «___» _____ 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи: веб-технології HTML, CSS, JavaScript; PHP; MySQL; Telegram Bot API; XAMPP.

Перелік питань, що підлягають дослідженню: аналіз існуючих систем обліку замовлень; проєктування структури бази даних; розробка веб-застосунку та API; реалізація Telegram-сповіщень; тестування програмного забезпечення.

Дата видачі завдання «___» _____ 2025 р.

Керівник бакалаврської _____ **Юрій МІЛОВІДОВ**
кваліфікаційної роботи (підпис)

Завдання взяв до виконання _____ **Віталій КОНОНЧУК**
(підпис)

Зміст

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	5
ВСТУП.....	7
1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Опис предметної області	9
1.2 Огляд інформаційних джерел та існуючих рішень.....	10
1.3 Аналіз вимог до програмної системи	11
1.4 Постановка задачі.....	13
1.5 Моделювання предметної області	14
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	18
2.1 Логічна модель даних (ER-діаграма).....	18
2.2 Діаграма класів	19
2.3 Діаграма компонентів	20
2.4 Архітектура системи	21
3 РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	23
3.1 Система управління базою даних	23
3.2 Розробка структури бази даних.....	28
3.3 Архітектура програмного забезпечення.....	33
3.4 Вибір інструментальних засобів розробки	37
3.5 Реалізація програмних модулів.....	39
4 РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ СИСТЕМИ.....	45

4.1 Тестування системи.....	45
4.2 Вимоги до апаратного та програмного забезпечення	51
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТКИ.....	58
Додаток А	58
Додаток Б.....	59
Додаток В	59
Додаток Г.....	60
Додаток Д.....	61
Додаток Е	61
Додаток Ж	62
Додаток З.....	62
Додаток И.....	63
Додаток К	64
Додаток Л.....	65

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – Application Programming Interface – інтерфейс прикладного програмування, що забезпечує взаємодію між клієнтською та серверною частинами системи.

DB – Database – база даних, що використовується для зберігання інформації про користувачів, товари та замовлення.

SQL – Structured Query Language – мова структурованих запитів для роботи з базами даних.

CRUD – Create, Read, Update, Delete – основні операції над даними в інформаційній системі.

UI – User Interface – інтерфейс користувача, що забезпечує взаємодію користувача з системою.

UX – User Experience – досвід користувача під час взаємодії з веб-системою.

HTTP – HyperText Transfer Protocol – протокол передачі даних між клієнтом і сервером.

JSON – JavaScript Object Notation – формат обміну даними між клієнтською та серверною частинами системи.

REST – Representational State Transfer – архітектурний стиль взаємодії веб-сервісів.

URL – Uniform Resource Locator – адреса ресурсу в мережі Інтернет.

ID – Identifier – унікальний ідентифікатор запису в базі даних.

MVC – Model-View-Controller – архітектурний шаблон побудови програмного забезпечення.

PHP – Hypertext Preprocessor – серверна мова програмування, що використовується для реалізації логіки системи.

JS – JavaScript – мова програмування, що використовується для реалізації клієнтської частини системи.

HTML – HyperText Markup Language – мова розмітки веб-сторінок.

CSS – Cascading Style Sheets – мова стилів для оформлення веб-інтерфейсу.

AJAX – Asynchronous JavaScript and XML – технологія асинхронного обміну даними між клієнтом і сервером без перезавантаження сторінки.

Auth – Authentication – процес перевірки користувача в системі.

Admin – Administrator – користувач із розширеними правами доступу до системи.

Order – замовлення користувача в інтернет-магазині.

Cart – кошик покупця для формування замовлення.

ВСТУП

Об'єктом дослідження виступає процес автоматизації обліку замовлень товарів в інтернет-магазині.

Предметом дослідження є методи та засоби проектування і реалізації інформаційної системи обліку замовлень товарів з використанням веб-технологій [3].

Метою кваліфікаційної роботи є розробка інформаційної системи обліку замовлень товарів в інтернет-магазині, яка забезпечує автоматизацію процесів управління товарами, оформлення замовлень та адміністрування системи.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

- виконати аналіз предметної області та дослідити особливості функціонування сучасних інтернет-магазинів;
- провести аналіз існуючих програмних рішень та визначити їх переваги й недоліки;
- сформулювати функціональні та нефункціональні вимоги до інформаційної системи;
- розробити UML-моделі та логічну структуру бази даних;
- спроектувати архітектуру програмного забезпечення;
- реалізувати клієнтську та серверну частини системи;
- розробити модулі обробки товарів, кошика та замовлень;
- реалізувати адміністративну панель для управління системою;
- провести тестування працездатності програмного забезпечення.

Особливістю розробленої системи є використання Telegram-сповіщень, модульної структури та централізованого управління замовленнями через веб-інтерфейс.

Практичне значення роботи полягає у створенні програмного забезпечення, яке може бути використане для автоматизації діяльності інтернет-магазину, спрощення процесу оформлення замовлень,

централізованого управління товарами та підвищення ефективності взаємодії з користувачами.

Структура роботи складається з чотирьох розділів, висновків, списку використаних джерел та додатків. У першому розділі проведено системний аналіз предметної області, досліджено сучасні рішення та сформовано вимоги до системи. У другому розділі виконано проєктування інформаційного та програмного забезпечення, побудовано UML-діаграми та структуру бази даних. У третьому розділі описано процес розробки програмного забезпечення, архітектуру системи та реалізацію основних функціональних модулів. Четвертий розділ присвячено тестуванню системи, вимогам до програмного та апаратного забезпечення, а також рекомендаціям щодо впровадження системи.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Предметна область обліку замовлень товарів в інтернет-магазині охоплює сукупність процесів, пов'язаних із управлінням каталогом товарів, обробкою інформації про користувачів, формуванням кошика покупця, оформленням замовлень та контролем їх виконання. Важливим елементом таких систем є забезпечення швидкого доступу до інформації про товари, можливість автоматичного оновлення даних та централізоване зберігання інформації у базі даних.

Система забезпечує управління товарами, замовленнями, кошиком покупця, адміністративними функціями та статистикою продажів.



Рисунок 1.1 – Взаємодія користувача

На рисунку 1.1 представлено загальну схему взаємодії користувача з інформаційною системою інтернет-магазину.

Тому виникає потреба у створенні зручної системи для автоматизації оформлення та обробки замовлень, яка забезпечуватиме зручний інтерфейс, автоматизацію процесів обліку замовлень та централізоване управління даними.

1.2 Огляд інформаційних джерел та існуючих рішень

Для обґрунтування необхідності розробки власної інформаційної системи обліку замовлень товарів доцільно провести аналіз сучасних програмних рішень, які використовуються у сфері електронної комерції. Існуючі платформи інтернет-магазинів відрізняються функціональними можливостями, архітектурою, способами адміністрування та рівнем інтеграції із зовнішніми сервісами. Порівняння існуючих платформ дозволило визначити функції, які доцільно реалізувати у власній системі.

Shopify є популярною платформою електронної комерції, яка забезпечує створення інтернет-магазинів із підтримкою товарів, замовлень та платіжних систем [21].

WooCommerce є модулем для WordPress та дозволяє реалізувати функціонал інтернет-магазину з підтримкою товарів і замовлень [23].

OpenCart характеризується простотою структури та відносно невеликими системними вимогами. Проте система має обмежені можливості масштабування та потребує додаткових модулів для реалізації розширеного функціоналу [22].

Magento використовується для великих інтернет-магазинів та підтримує масштабовану архітектуру [24].

Для більш наочного аналізу існуючих рішень доцільно провести їх порівняння за основними функціональними характеристиками. У таблиці 1.1 представлено узагальнене порівняння найбільш популярних платформ електронної комерції.

Таблиця 1.1 – Порівняльна характеристика систем електронної комерції

№ з/п	Назва системи	Переваги	Недоліки
1	Shopify	Простота використання, хмарна інфраструктура	Обмежена кастомізація, платні сервіси
2	WooCommerce	Гнучкість, велика кількість плагінів	Залежність від модулів, складність оптимізації
3	OpenCart	Простота структури, невеликі системні вимоги	Обмежене масштабування
4	Magento	Масштабованість, розширений функціонал	Складність адміністрування, високі вимоги до сервера
5	Розроблювана система	Простота використання, модульна архітектура, адаптація під потреби користувача	Обмежений функціонал порівняно з корпоративними платформами

Проведений аналіз існуючих рішень показав необхідність створення інформаційної системи з підтримкою управління товарами, оформлення замовлень, Telegram-сповіщень та адміністративної панелі.

1.3 Аналіз вимог до програмної системи

Розробка інформаційної системи обліку замовлень товарів в інтернет-магазині потребує формування чітких функціональних та нефункціональних вимог до програмного забезпечення. Аналіз вимог дозволяє визначити основні можливості системи, особливості взаємодії користувачів із програмним забезпеченням, а також технічні характеристики, необхідні для забезпечення стабільної та ефективної роботи системи.

Функціональні вимоги описують основні задачі, які повинна виконувати інформаційна система. Вони визначають перелік операцій, доступних користувачам та адміністраторам системи, а також логіку обробки даних.

До основних функціональних вимог системи належать:

- інтеграція із Telegram Bot API [20];
- система адміністративних сповіщень;
- модуль статистики та аналітики;
- підтримка ролей адміністраторів;
- імпорт товарів із CSV-файлів.

Система повинна забезпечувати відображення товарів, оформлення замовлень, управління кошиком покупця та адміністративними функціями.

Адміністративна частина системи повинна забезпечувати управління товарами, замовленнями, статистикою та користувачами.

Крім основних функцій, система повинна забезпечувати стабільну та швидку роботу.

У системі також реалізовано інтеграцію з Telegram Bot API для автоматичного надсилання повідомлень користувачам та адміністраторам після оформлення замовлення.

Крім основного функціоналу, система підтримує формування статистики продажів, управління адміністраторами та імпорт товарів із CSV-файлів.

Для більш наочного представлення вимог до програмного забезпечення доцільно сформувані таблицю функціональних та нефункціональних вимог.

Таблиця 1.2 – Функціональні та нефункціональні вимоги системи

Функціональні вимоги	Нефункціональні вимоги
Відображення товарів	Висока швидкість роботи системи
Оформлення замовлень	Стабільність роботи
Управління кошиком	Зручний інтерфейс користувача
Адміністрування товарів	Масштабованість системи
Формування звітів	Захист даних користувачів
Робота API	Підтримка сучасних браузерів

1.4 Постановка задачі

На основі проведеного аналізу предметної області, існуючих програмних рішень та вимог до програмного забезпечення було визначено необхідність розробки інформаційної системи обліку замовлень товарів в інтернет-магазині. Основною задачею розробки є створення програмного забезпечення, яке забезпечуватиме автоматизацію процесів управління товарами, оформлення замовлень та адміністрування системи через веб-інтерфейс.

Розроблювана система повинна реалізовувати взаємодію між клієнтською та серверною частинами програмного забезпечення, забезпечувати централізоване зберігання інформації у базі даних та підтримувати можливість обробки запитів користувачів у режимі реального часу. Під час розробки основна увага приділялася швидкості оформлення замовлень та зручності роботи адміністратора.

Для досягнення поставленої мети необхідно реалізувати модулі управління товарами, кошиком покупця, оформлення замовлень, адміністративну панель та систему Telegram-сповіщень.

Система повинна забезпечувати автоматизацію обліку замовлень, збереження інформації у базі даних, формування статистики продажів та підтримку адміністративних функцій.

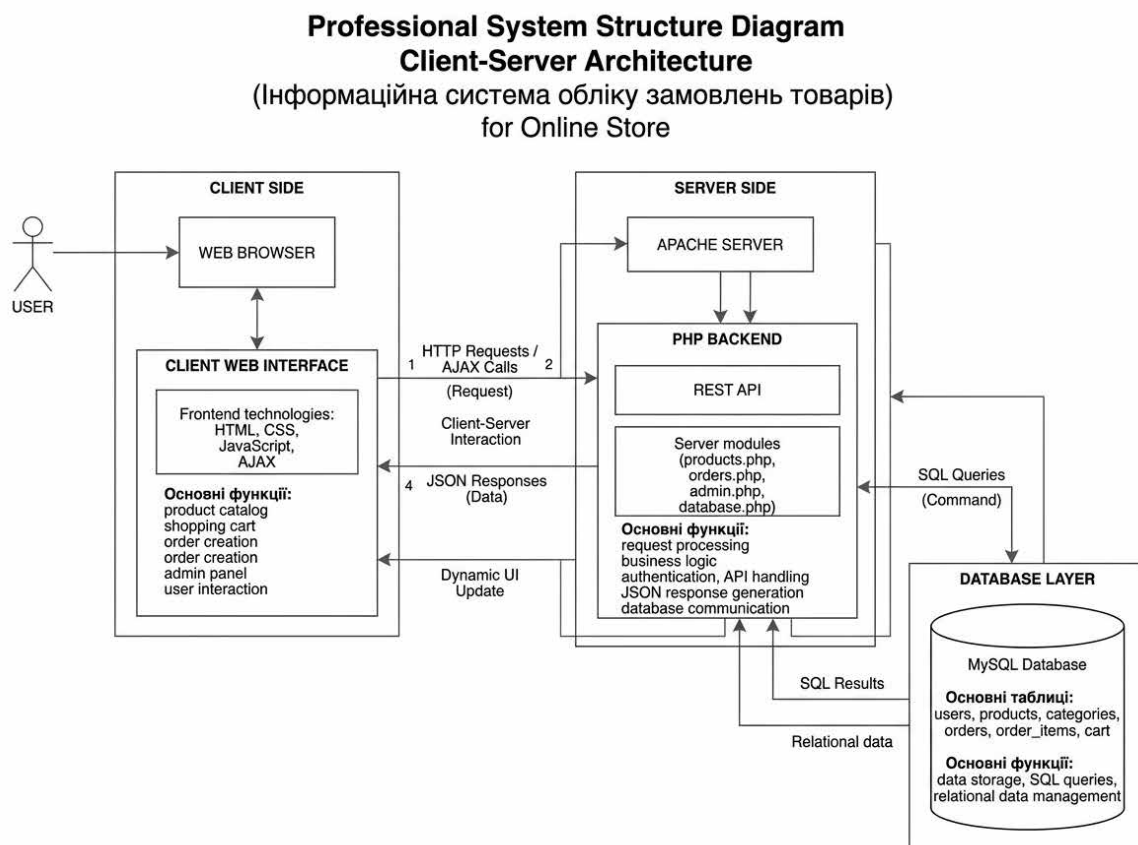


Рисунок 1.2 – Структура системи

На рисунку 1.2 представлено структурну схему інформаційної системи.

Результатом виконання поставленої задачі повинна стати інформаційна система, яка забезпечуватиме автоматизацію процесів обліку замовлень товарів, підтримуватиме ефективне управління каталогом продукції та забезпечуватиме зручну взаємодію між користувачем і адміністрацією інтернет-магазину.

1.5 Моделювання предметної області

Предметна область інформаційної системи обліку замовлень товарів включає взаємодію між користувачем, серверною частиною системи та базою даних. Основними об'єктами предметної області є товари, користувачі, замовлення, категорії товарів та адміністративна панель управління.

Діаграма варіантів використання дозволяє описати взаємодію користувачів із системою та визначити основні функції програмного забезпечення. Основними учасниками системи є користувач та адміністратор.

Користувач системи може переглядати товари, формувати кошик та оформлювати замовлення, а адміністратор — управляти товарами, замовленнями та статистикою системи.

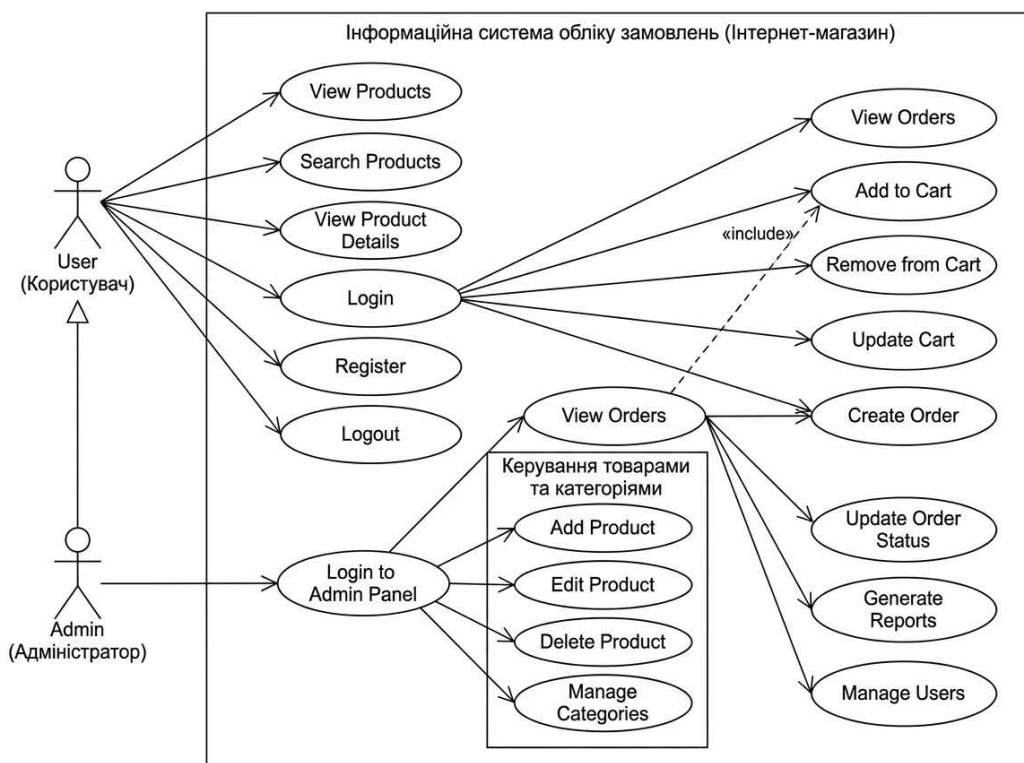


Рисунок 1.3 – Use-case діаграма

На рисунку 1.3 представлено діаграму варіантів використання системи.

Для опису послідовності виконання процесу оформлення замовлення доцільно використовувати діаграму діяльності. Дана модель дозволяє відобразити логіку взаємодії користувача із системою під час вибору товарів, формування кошика та підтвердження замовлення.

Order Placement Process in an E-commerce Order Management System

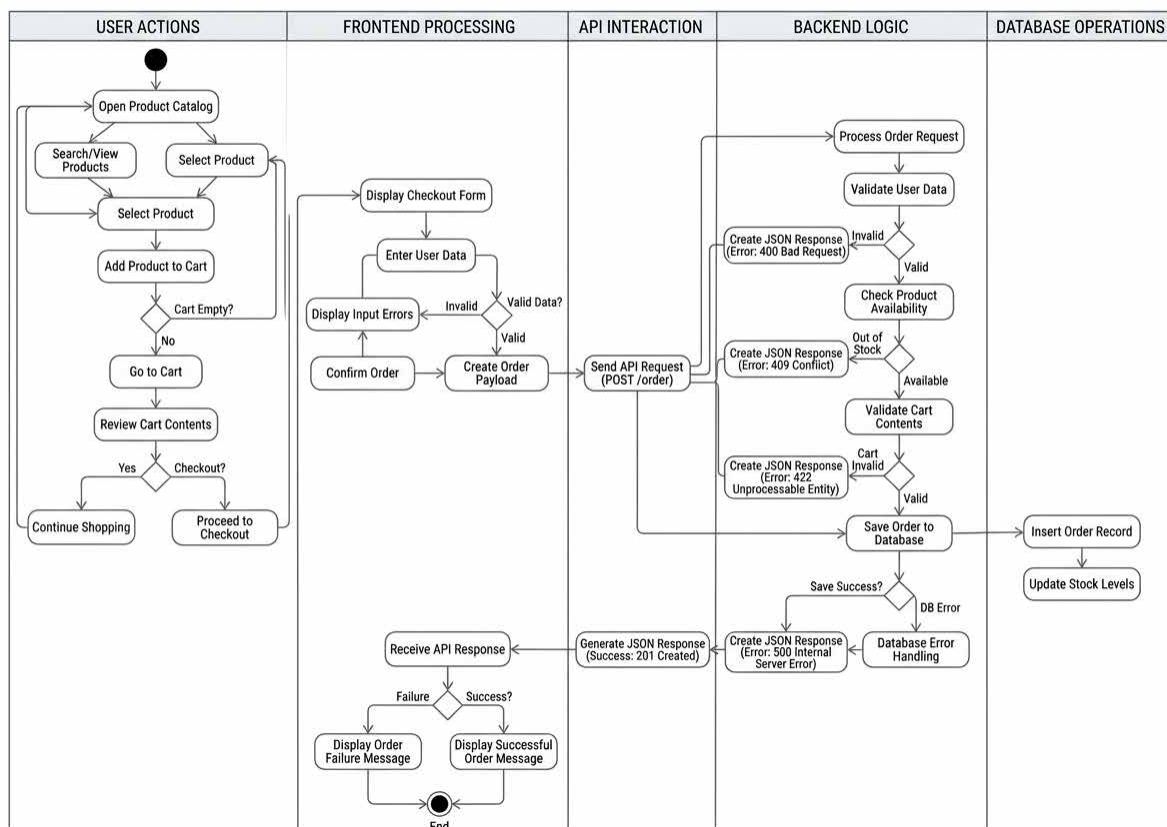


Рисунок 1.4 – Оформлення замовлення

На рисунку 1.4 представлено процес оформлення замовлення.

Для опису структури даних інформаційної системи використовується діаграма класів. Основними сутностями системи є користувачі, товари, замовлення та адміністративний модуль.

Для відображення структури програмного забезпечення та взаємодії між його компонентами доцільно використовувати діаграму компонентів. Дана модель дозволяє представити взаємодію між клієнтською частиною системи, серверною логікою, API та базою даних.

Архітектура системи включає такі основні компоненти:

- клієнтський веб-інтерфейс;
- серверну частину програмного забезпечення;
- API для обробки запитів;
- систему управління базою даних;

– адміністративну панель.

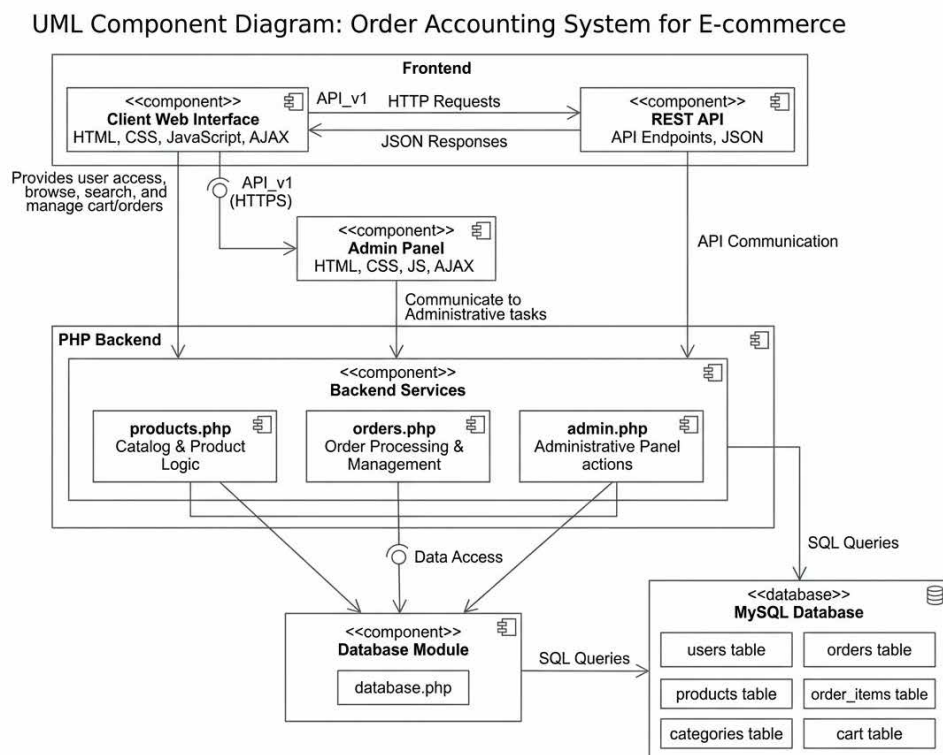


Рисунок 1.5 – Компоненти системи

На рисунку 1.5 представлено діаграму компонентів інформаційної системи.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель даних (ER-діаграма)

Одним із ключових етапів проєктування інформаційної системи є розробка логічної моделі даних. Логічна модель дозволяє формалізувати структуру бази даних, визначити основні сутності системи, їх атрибути та взаємозв'язки між ними. Побудова ER-діаграми (Entity Relationship Diagram) забезпечує наочне представлення структури даних та спрощує подальшу реалізацію системи управління базою даних.

**ER-діаграма системи обліку замовлень інтернет-магазину
(Логічна модель)**

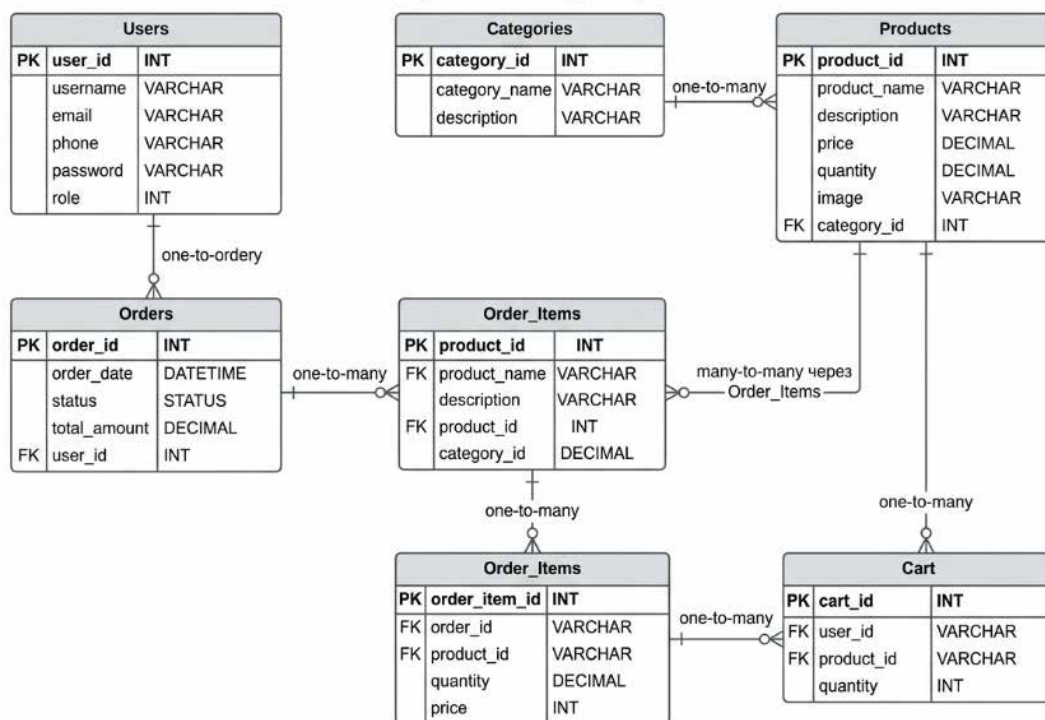


Рисунок 2.1 – ER-діаграма

На рисунку 2.1 представлено ER-діаграму бази даних.

2.2 Діаграма класів

Діаграма класів є однією з основних UML-моделей, що використовується для опису структури програмного забезпечення [12]. Вона дозволяє представити класи системи, їх атрибути, методи та взаємозв'язки між об'єктами. Побудова діаграми класів забезпечує формалізацію структури програмного забезпечення та спрощує процес реалізації функціональних компонентів системи.

У межах розробки інформаційної системи обліку замовлень товарів діаграма класів використовується для представлення основних сутностей програмного забезпечення та логіки їх взаємодії. Основними класами системи є User, Admin, Product, Category, Cart та Order.

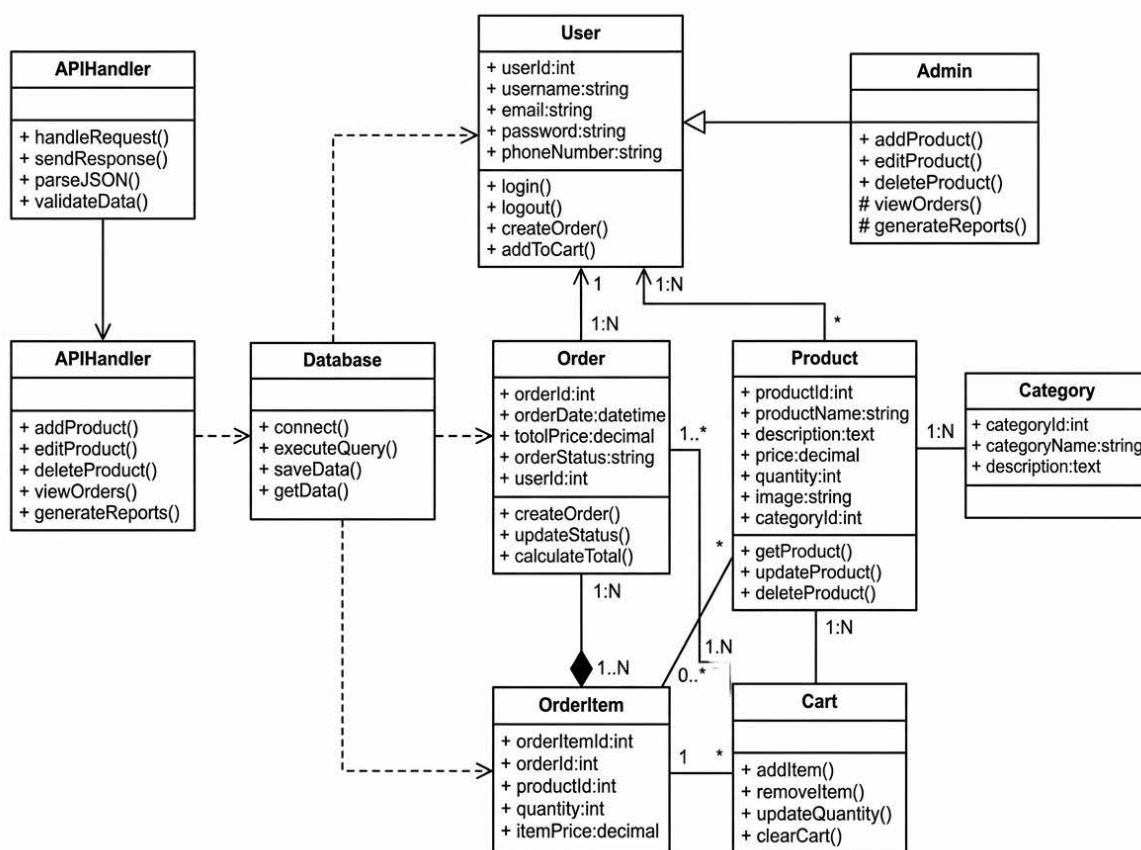


Рисунок 2.2 – Діаграма класів

На рисунку 2.2 представлено діаграму класів системи.

2.3 Діаграма компонентів

Діаграма компонентів використовується для відображення загальної структури програмного забезпечення та взаємодії між його основними складовими. Вона дозволяє представити архітектуру інформаційної системи у вигляді окремих компонентів, що взаємодіють між собою через визначені інтерфейси та механізми обміну даними.

У межах розробки інформаційної системи обліку замовлень товарів діаграма компонентів використовується для моделювання взаємодії між клієнтською частиною системи, серверною логікою, API-модулями та базою даних. Використання компонентного підходу дозволяє забезпечити модульність програмного забезпечення, спростити процес підтримки системи та забезпечити можливість її подальшого масштабування.

Архітектура розроблюваної системи базується на клієнт-серверному принципі взаємодії [11]. Користувач здійснює взаємодію із системою через веб-інтерфейс, який надсилає запити до серверної частини. Сервер виконує обробку даних, взаємодіє із базою даних та повертає результат клієнтській частині у форматі JSON.

Основними компонентами інформаційної системи є:

- клієнтський веб-інтерфейс;
- серверна частина програмного забезпечення;
- API-модулі;
- модуль адміністрування;
- система управління базою даних.

Система управління базою даних забезпечує централізоване зберігання інформації про користувачів, товари, категорії та замовлення. База даних використовується серверною частиною системи для виконання SQL-запитів та збереження результатів роботи програмного забезпечення.

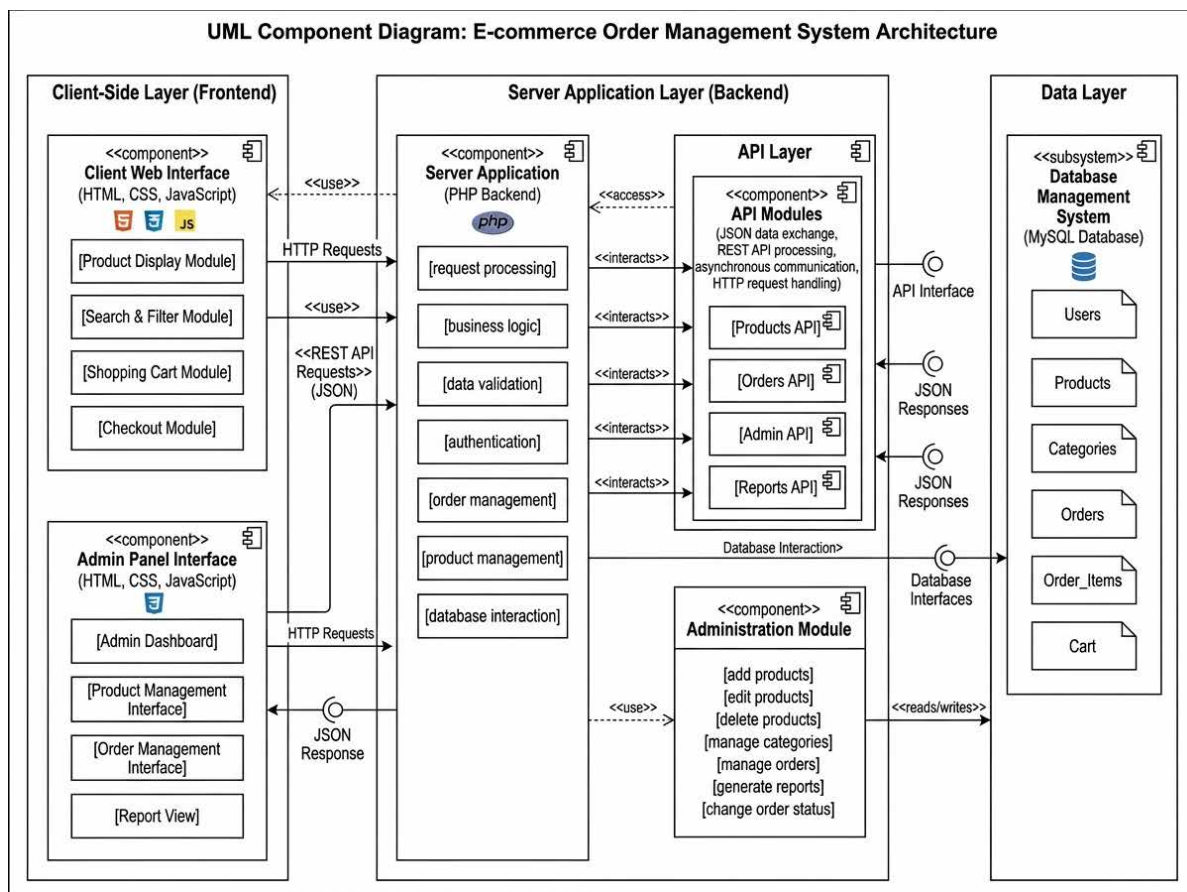


Рисунок 2.3 – Компоненти системи

На рисунку 2.3 представлено діаграму компонентів системи.

2.4 Архітектура системи

Інформаційна система реалізована за клієнт-серверною архітектурою. Клієнтська частина забезпечує взаємодію користувача із системою через веб-інтерфейс, а серверна частина відповідає за обробку запитів, бізнес-логіку та взаємодію з базою даних.

Основними компонентами системи є веб-інтерфейс користувача, серверні PHP-модулі, база даних MySQL та Telegram Bot API.

Для взаємодії між компонентами системи використовуються HTTP-запити та формат JSON [6].

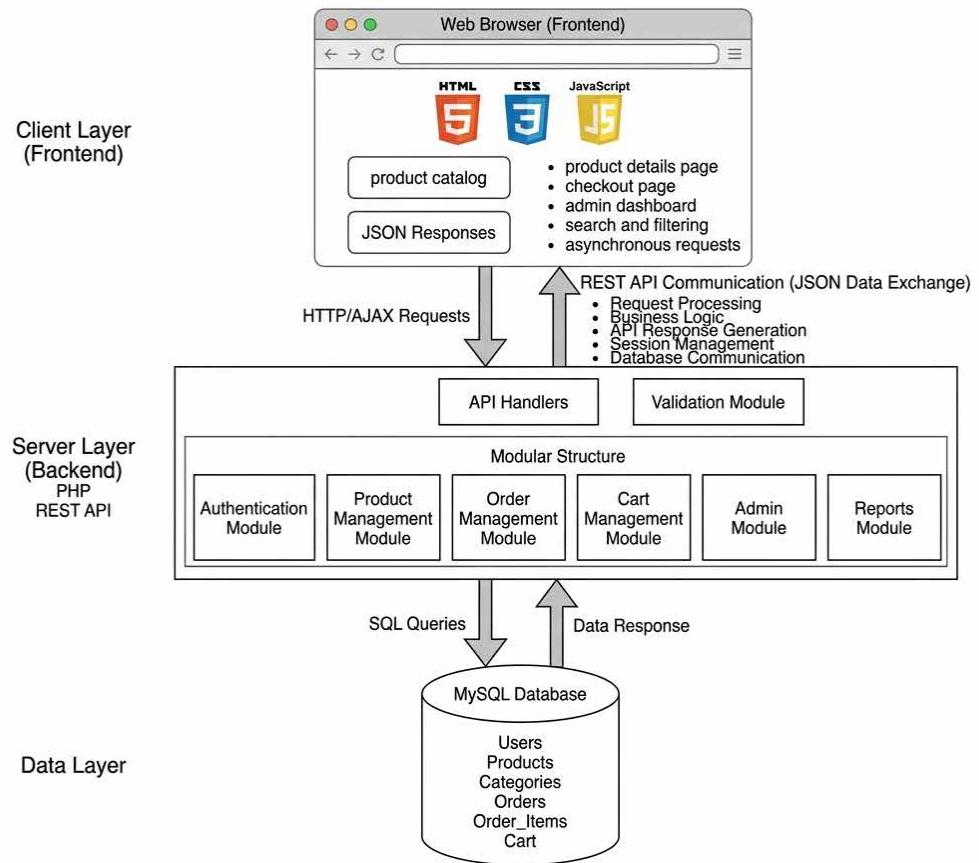


Рисунок 2.4 – Архітектура системи

На рисунку 2.4 представлено архітектуру інформаційної системи.

3 РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Система управління базою даних

Для забезпечення роботи інформаційної системи обліку замовлень товарів було використано систему управління базами даних MySQL [5]. Вибір MySQL обумовлений високою продуктивністю, підтримкою багатокористувацького режиму роботи, простотою інтеграції з PHP та широким використанням у веб-застосунках.

Система управління базою даних використовується для централізованого зберігання інформації про користувачів, товари, категорії, замовлення та інші компоненти інформаційної системи. Взаємодія серверної частини із базою даних здійснюється через SQL-запити [10].

Таблиця users призначена для зберігання інформації про користувачів системи.

Таблиця 3.1 – Структура таблиці users

По ле	Тип даних	Опис
id	INT	Унікальний ідентифікатор користувача
us ername	VARCHA R(255)	Ім'я користувача
em ail	VARCHA R(255)	Електронна пошта
pa ssword	VARCHA R(255)	Пароль користувача
rol	VARCHA	Роль користувача

Поле	Тип даних	Опис
е	R(50)	

Для створення таблиці users використовується SQL-запит:

```
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    role VARCHAR(50) DEFAULT 'user'
);
```

Таблиця products використовується для зберігання інформації про товари інтернет-магазину.

Таблиця 3.2 – Структура таблиці products

Поле	Тип даних	Опис
id	INT	Ідентифікатор товару
name	VARCHAR(255)	Назва товару
description	TEXT	Опис товару
price	DECIMAL(10,2)	Ціна товару
quantity	INT	Кількість товару
image	VARCHAR(255)	Зображення товару

Пол е	Тип даних	Опис
category_id	INT	Ідентифікатор категорії

SQL-запит створення таблиці products:

```
CREATE TABLE products (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    description TEXT,
    price DECIMAL(10,2),
    quantity INT,
    image VARCHAR(255),
    category_id INT
);
```

Для групування товарів за категоріями використовується таблиця categories.

Таблиця 3.3 – Структура таблиці categories

Пол е	Тип даних	Опис
id	INT	Ідентифікатор категорії
name	VARCHAR(255)	Назва категорії
description	TEXT	Опис категорії

Таблиця orders використовується для зберігання інформації про оформлені замовлення користувачів.

Таблиця 3.4 – Структура таблиці orders

По ле	Тип даних	Опис
id	INT	Ідентифікатор замовлення
user _id	INT	Ідентифікатор користувача
ord er_date	DATETIME	Дата створення замовлення
stat us	VARCHAR R(50)	Статус замовлення
tota l_price	DECIMAL L(10,2)	Загальна сума замовлення

Для зберігання інформації про товари у складі замовлення використовується таблиця order_items.

Таблиця 3.5 – Структура таблиці order_items

По ле	Тип даних	Опис
id	INT	Ідентифікатор запису
orde r_id	INT	Ідентифікатор замовлення
pro duct_id	INT	Ідентифікатор товару
qua ntity	INT	Кількість товару
pric	DECIMAL	Ціна товару

По ле	Тип даних	Опис
е	L(10,2)	

Таблиця cart використовується для реалізації функціоналу кошика покупця.

Таблиця 3.6 – Структура таблиці cart

По ле	Тип даних	Опис
id	INT	Ідентифікатор запису
user_id	INT	Ідентифікатор користувача
product_id	INT	Ідентифікатор товару
quantity	INT	Кількість товару

Між таблицями бази даних реалізовано логічні зв'язки. Один користувач може оформити декілька замовлень, тому між таблицями users та orders існує зв'язок типу «один до багатьох». Аналогічний зв'язок існує між таблицями categories та products.

Зв'язок між товарами та замовленнями реалізовано через проміжну таблицю order_items, що дозволяє реалізувати зв'язок типу «багато до багатьох».

Наприклад, для отримання списку товарів використовується запит:

```
SELECT * FROM products;
```

Для додавання нового замовлення використовується запит:

```
INSERT INTO orders(user_id, order_date, status, total_price)
VALUES (1, NOW(), 'new', 1500.00);
```

Це дозволяє забезпечити стабільну роботу програмного забезпечення та зменшити ризик пошкодження даних.

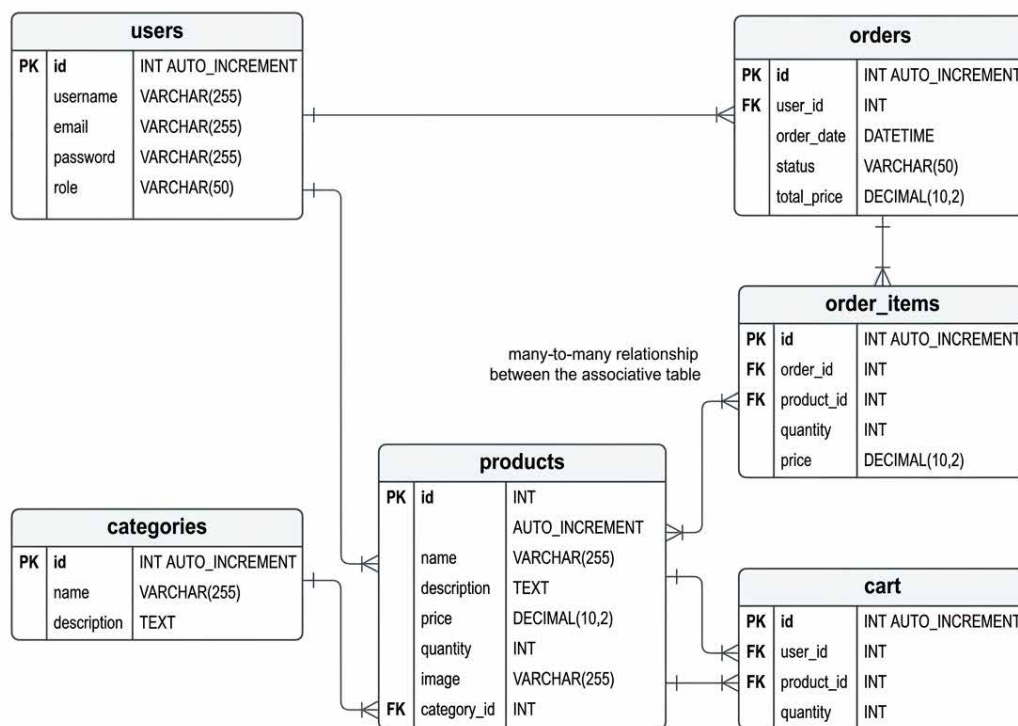


Рисунок 3.1 – Структура БД

На рисунку 3.1 представлено структуру бази даних.

Таким чином, використання системи управління базами даних MySQL дозволило реалізувати централізоване зберігання інформації, підтримку взаємозв'язків між сутностями предметної області та ефективну обробку даних у межах інформаційної системи обліку замовлень товарів.

3.2 Розробка структури бази даних

Після визначення основних функціональних вимог до інформаційної системи було виконано проектування структури бази даних. На цьому етапі

було створено структуру таблиць бази даних, яка забезпечує ефективне зберігання інформації, підтримку взаємозв'язків між сутностями та швидке виконання SQL-запитів.

Структура бази даних розроблена з урахуванням особливостей функціонування інтернет-магазину та включає таблиці для роботи з користувачами, товарами, категоріями, замовленнями та кошиком покупця. Під час проєктування бази даних використовувався реляційний підхід, що дозволяє підтримувати логічні зв'язки між таблицями та забезпечувати цілісність інформації.

У процесі розробки структури бази даних були визначені такі основні сутності:

- користувачі;
- товари;
- категорії;
- замовлення;
- елементи замовлення;
- кошик покупця.

Для кожної таблиці було визначено первинні ключі, зовнішні ключі та типи даних полів.

Таблиця 3.7 – Основні характеристики таблиці users

По ле	Тип даних	Призначен ня
id	INT	Первинний ключ
us ername	VARCHA R(255)	Ім'я користувача
em ail	VARCHA R(255)	Електронна пошта

password	VARCHAR R(255)	Пароль
role	VARCHAR R(50)	Роль користувача

Таблиця 3.8 – Основні характеристики таблиці products

Поле	Тип даних	Призначення
id	INT	Первинний ключ
name	VARCHAR R(255)	Назва товару
description	TEXT	Опис товару
price	DECIMAL L(10,2)	Ціна товару
quantity	INT	Кількість товару
image	VARCHAR R(255)	Зображення товару
category_id	INT	Зовнішній ключ категорії

Таблиця 3.9 – Основні характеристики таблиці categories

Поле	Тип даних	Призначення
id	INT	Первинний ключ
name	VARCHAR R(255)	Назва категорії

desc ription	TEXT	Опис категорії
-----------------	------	-------------------

Таблиця 3.10 – Основні характеристики таблиці orders

По ле	Тип даних	Призначення
id	INT	Первинний ключ
user _id	INT	Ідентифікатор користувача
ord er_date	DATETIME	Дата створення
stat us	VARCHAR R(50)	Статус замовлення
tota l_price	DECIMAL L(10,2)	Загальна сума

Таблиця 3.11 – Основні характеристики таблиці order_items

По ле	Тип даних	Призначення
id	INT	Первинний ключ
orde r_id	INT	Ідентифікатор замовлення
pro duct_id	INT	Ідентифікатор товару
qua ntity	INT	Кількість товару
pric e	DECIMAL L(10,2)	Ціна товару

Таблиця 3.12 – Основні характеристики таблиці cart

Поле	Тип даних	Призначення
id	INT	Первинний ключ
user_id	INT	Ідентифікатор користувача
product_id	INT	Ідентифікатор товару
quantity	INT	Кількість товару

Основними таблицями системи є users, products, orders, categories та cart.

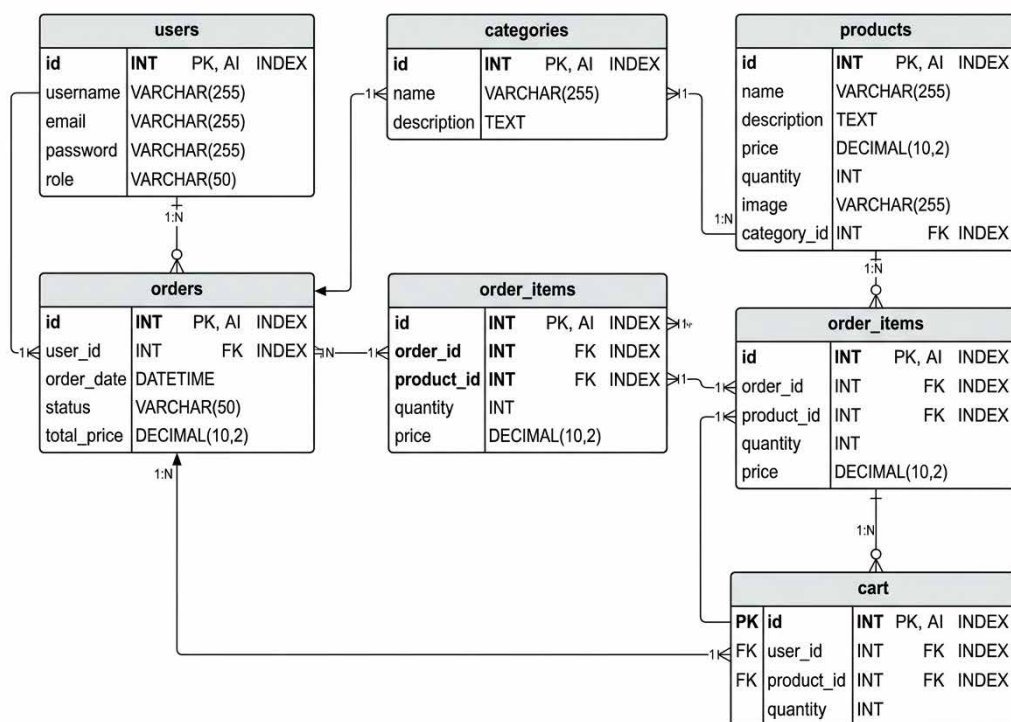


Рисунок 3.2 – Логічна структура БД

На рисунку 3.2 представлено логічну структуру бази даних.

3.3 Архітектура програмного забезпечення

Архітектура програмного забезпечення визначає структуру інформаційної системи, взаємодію між її компонентами та принципи організації обробки даних. Під час розробки інформаційної системи обліку замовлень товарів було використано клієнт-серверну архітектуру, що забезпечує розподіл функціоналу між клієнтською та серверною частинами системи.

Загальна структура програмного забезпечення складається з таких основних компонентів:

- Telegram Bot API [20];
- клієнтська частина;
- серверна частина;
- API-модулі;
- система управління базою даних.

Клієнтська частина реалізована з використанням HTML, CSS та JavaScript і відповідає за взаємодію користувача із системою [6–9]. Основною задачею клієнтської частини є відображення інформації, обробка дій користувача та надсилання HTTP-запитів до серверної частини системи.

До основних модулів клієнтської частини належать:

- модуль роботи з товарами;
- модуль кошика;
- модуль оформлення замовлення;
- модуль адміністрування;
- модуль генерації звітів.

Для реалізації клієнтської логіки у системі використовуються JavaScript-файли:

Таблиця 3.13 – Основні JavaScript-модулі системи

Мо дуль	Призначення
------------	-------------

Мо дуль	Призначення
products.js	Відображення та обробка товарів
cart.js	Реалізація функціоналу кошика
orders.js	Оформлення та обробка замовлень
admin.js	Адміністративна панель
reports.js	Формування звітів

Модуль products.js відповідає за отримання списку товарів із сервера, їх відображення у веб-інтерфейсі та взаємодію користувача із каталогом продукції.

Модуль cart.js реалізує логіку додавання товарів до кошика, зміну кількості товарів та видалення товарів із кошика.

Модуль orders.js використовується для оформлення замовлень та передачі інформації про замовлення на сервер.

Модуль admin.js забезпечує реалізацію адміністративної панелі, через яку адміністратор може управляти товарами та переглядати список замовлень.

Модуль reports.js використовується для відображення статистики продажів та інформації про замовлення.

Серверна частина програмного забезпечення реалізована мовою програмування PHP [4]. Вона відповідає за обробку HTTP-запитів, взаємодію з базою даних та виконання бізнес-логіки системи.

У серверній частині використовуються окремі API-обробники:

Таблиця 3.14 – Основні PHP-модулі серверної частини

Модуль	Призначення
products.php	Обробка запитів до товарів
orders.php	Обробка замовлень
admin.php	Адміністративні функції
database.php	Підключення до бази даних

Файл database.php забезпечує підключення до бази даних MySQL та використовується всіма серверними модулями системи.

Приклад підключення до бази даних:

```
<?php
$host = "localhost";
$user = "root";
$password = "";
$dbase = "online_store";

$conn = mysqli_connect($host, $user, $password, $dbase);
?>
```

Обмін даними між клієнтською та серверною частинами здійснюється через API із використанням HTTP-запитів та формату JSON.

Приклад отримання списку товарів із сервера:

```
fetch('api/products.php')
    .then(response => response.json())
    .then(data => {
        console.log(data);
    });
```

Після отримання запиту серверна частина виконує SQL-запит до бази даних та повертає результат у форматі JSON.

Приклад SQL-запиту:

```
$sql = "SELECT * FROM products";
```

```
$result = mysqli_query($conn, $sql);
```

Взаємодію між компонентами системи можна представити у вигляді такої схеми:

Користувач → Веб-інтерфейс → PHP API → MySQL



Telegram Bot API

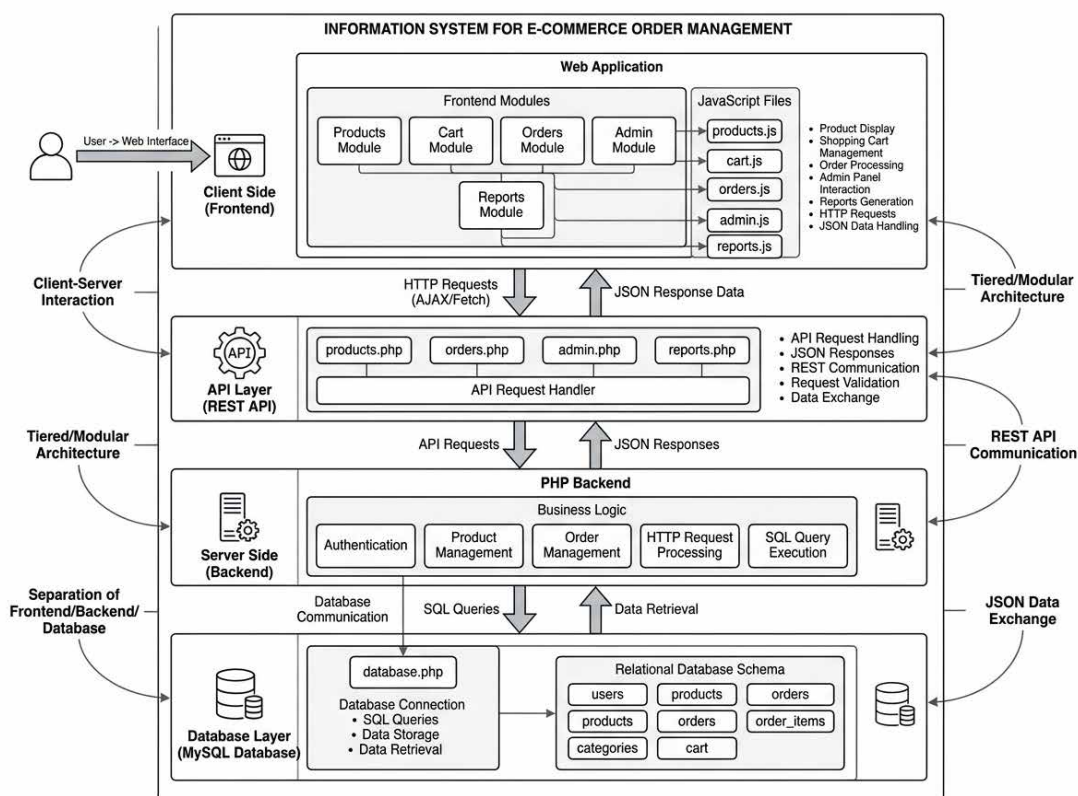


Рисунок 3.3 – Архітектура ПЗ

На рисунку 3.3 представлено архітектуру програмного забезпечення.

3.4 Вибір інструментальних засобів розробки

Під час розробки інформаційної системи обліку замовлень товарів було обрано сучасні програмні засоби та технології, які забезпечують ефективну реалізацію веб-застосунку, підтримку клієнт-серверної взаємодії та стабільну роботу системи. Вибір інструментальних засобів здійснювався з урахуванням функціональних вимог системи, простоти інтеграції компонентів та можливості подальшого масштабування програмного забезпечення.

Для створення користувацького інтерфейсу використовувалися HTML, CSS та JavaScript. Серверна частина системи реалізована мовою PHP, а для зберігання інформації використовується система управління базами даних MySQL.

Під час розробки програмного забезпечення використовувалося середовище Visual Studio Code та локальне серверне середовище XAMPP [17] [19].

Для автоматичного надсилання повідомлень користувачам та адміністраторам використовується Telegram Bot API [20].

Таблиця 3.15 – Інструментальні засоби розробки

Інструмент	Призначення
HTML	Структура веб-сторінок
CSS	Стилізація інтерфейсу
JavaScript	Клієнтська логіка
PHP	Серверна частина системи
MySQL	Система управління базою даних
AJAX	Асинхронний обмін даними



Використання зазначених інструментальних засобів дозволило реалізувати повноцінну інформаційну систему обліку замовлень товарів із підтримкою клієнт-серверної архітектури, API-взаємодії та централізованого зберігання інформації.

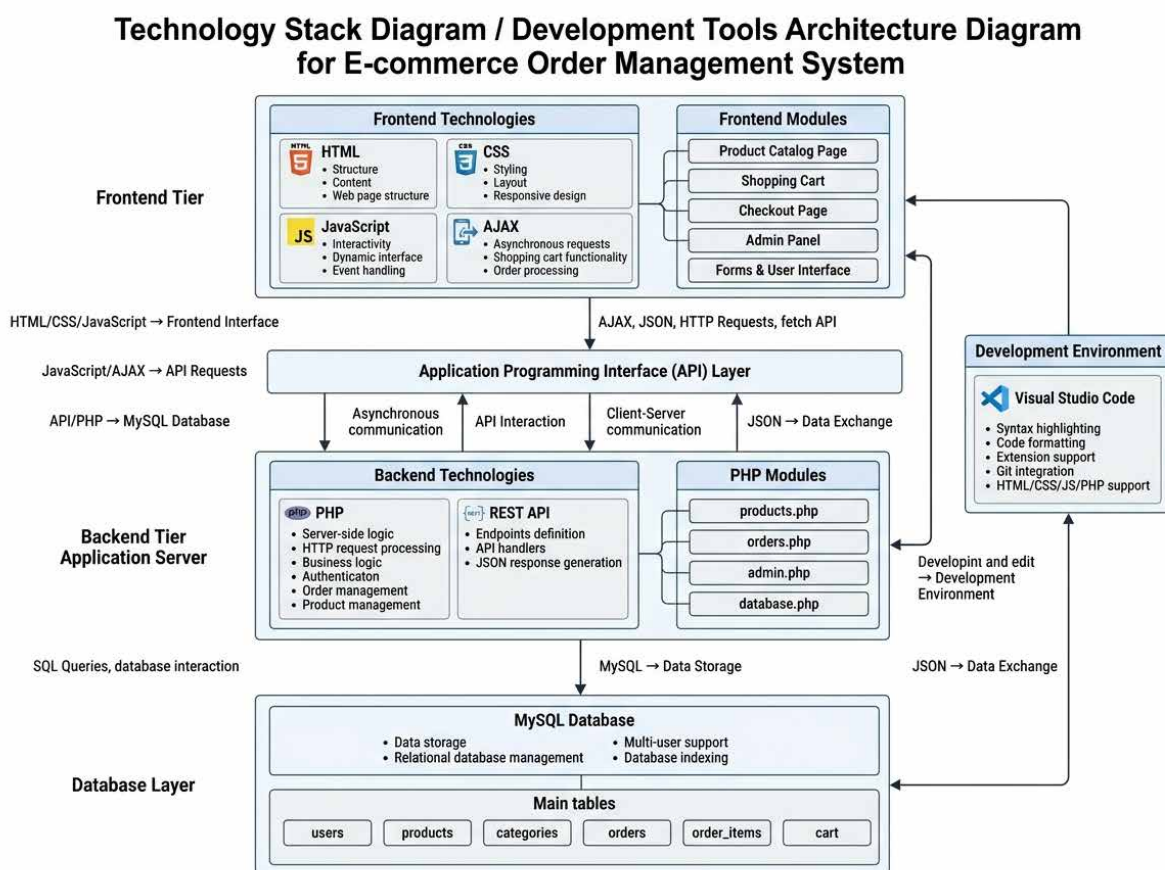


Рисунок 3.4 – Взаємодія технологій

На рисунку 3.4 представлено взаємодію використаних технологій.

3.5 Реалізація програмних модулів

Після проєктування архітектури системи та структури бази даних реалізовано основні програмні модулі інформаційної системи обліку замовлень товарів. Реалізація програмних модулів здійснювалася відповідно до функціональних вимог системи та передбачала створення окремих компонентів для роботи з товарами, кошиком, замовленнями та адміністративною панеллю.

Таблиця 3.16 – Основні програмні модулі системи

Модуль	Призначення
products.js products.php	Робота з товарами
cart.js	Реалізація кошика
orders.js orders.php	Оформлення замовлень
admin.js admin.php	Адміністративна панель
reports.js	Формування звітів
database.php	Підключення до бази даних

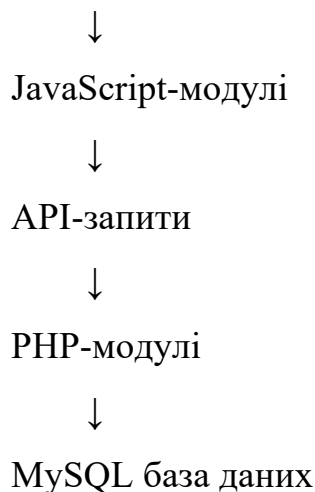
Серверна частина модуля виконує SQL-запит до бази даних та повертає результат у форматі JSON.

Після отримання запиту серверна частина виконує додавання замовлення до таблиці orders.

Модуль адміністрування реалізований за допомогою файлів admin.js та admin.php. Основною задачею модуля є забезпечення можливості управління товарами та замовленнями через адміністративну панель.

Структуру взаємодії програмних модулів можна представити у вигляді схеми:

Клієнтський інтерфейс



Для оперативного отримання інформації про нові замовлення у системі було реалізовано інтеграцію з Telegram Bot API. Після оформлення замовлення інформація про покупця та перелік товарів автоматично надсилається адміністратору у Telegram.

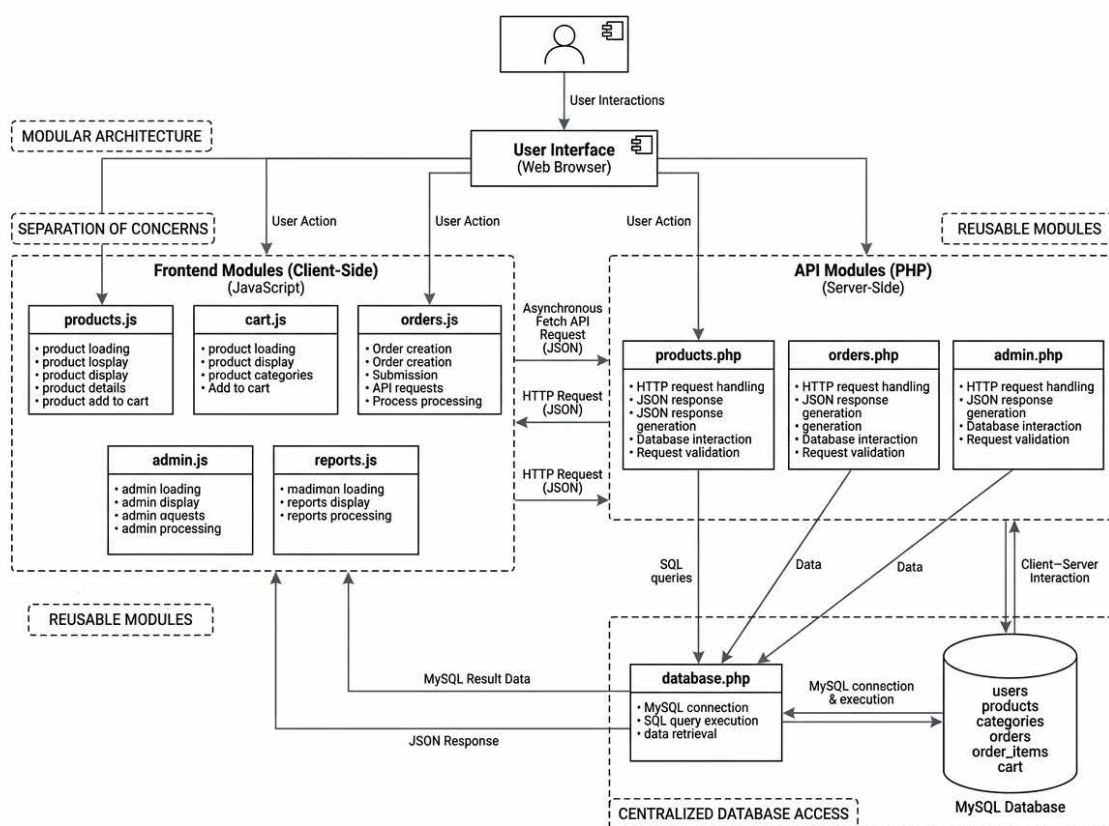


Рисунок 3.5 – Взаємодія модулів

На рисунку 3.5 представлено взаємодію модулів системи.

Для розширення функціональних можливостей інформаційної системи було реалізовано інтеграцію з Telegram Bot API. Модуль використовується для автоматичного надсилення повідомлень користувачам та адміністраторам після оформлення замовлення.

Після оформлення замовлення адміністратор отримує повідомлення у Telegram із переліком товарів та загальною сумою. Сформоване повідомлення надсилається через Telegram-бота із використанням HTTP-запиту до Telegram API.

Основними функціями Telegram-модуля є:

- надсилення повідомлення користувачу;
- надсилення повідомлення адміністратору;
- автоматичне формування тексту повідомлення;
- передача інформації про доставку;
- сповіщення про нові замовлення.

Після оформлення замовлення користувач отримує повідомлення із деталями покупки, а адміністратор — сповіщення про нове замовлення та можливість оперативної обробки заявки.

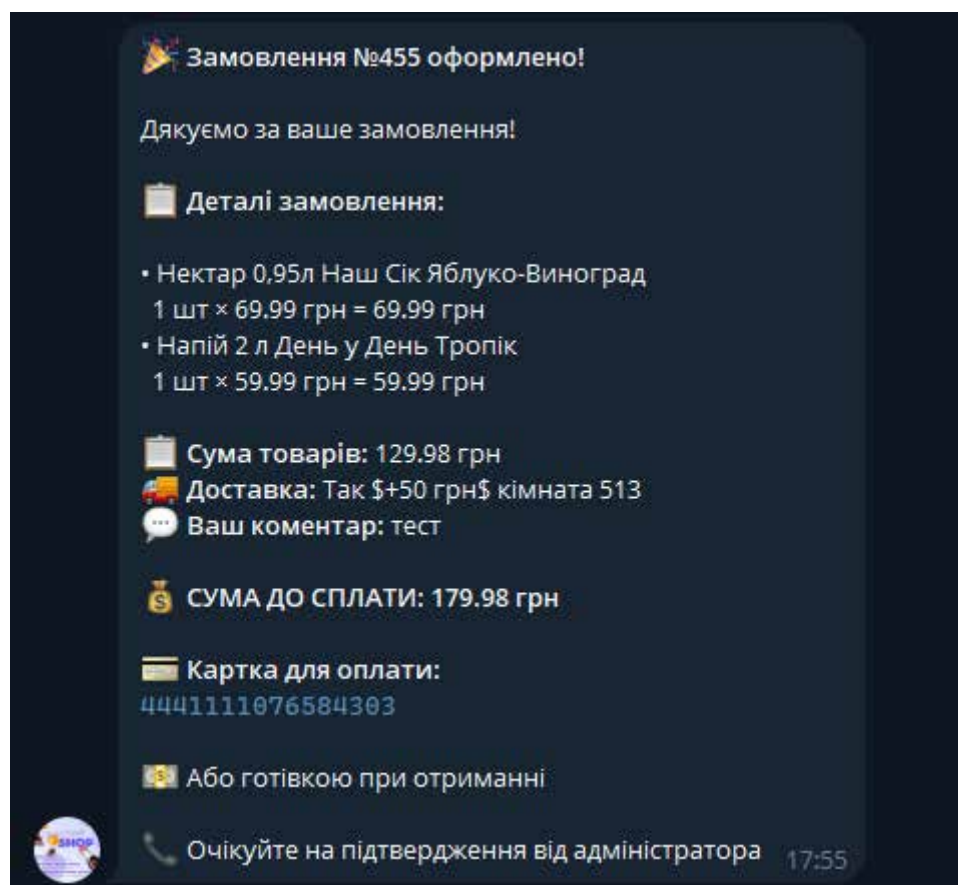


Рисунок 3.6 – Telegram-сповіщення

На рисунку 3.6 представлено приклад Telegram-сповіщення.

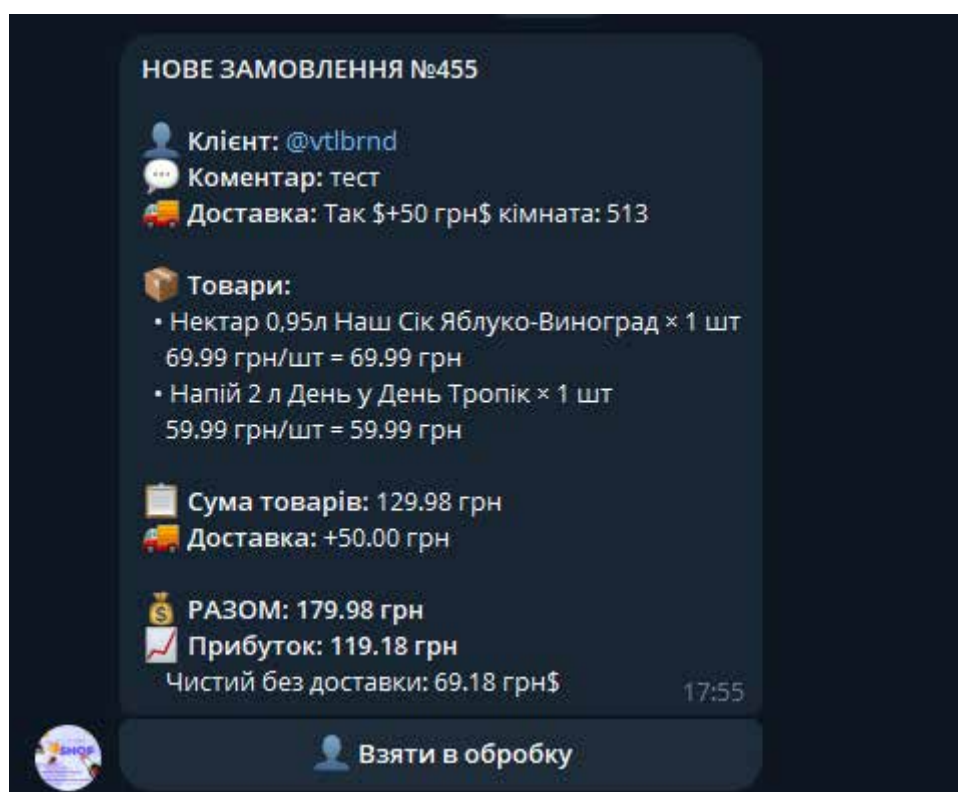


Рисунок 3.7 – Повідомлення адміну

На рисунку 3.7 представлено повідомлення для адміністратора.

Під час тестування Telegram-сповіщень перевірялася швидкість надсилання повідомлень після оформлення замовлення.

Крім інтеграції з Telegram Bot API у системі реалізовано модуль статистики та аналітики. Даний модуль дозволяє отримувати інформацію про:

- кількість замовлень;
- загальний прибуток;
- собівартість товарів;
- виручку;
- статистику адміністраторів;
- популярність товарів.

Для формування статистики використовуються SQL-запити до бази даних та автоматичний розрахунок фінансових показників.

У системі також реалізовано модуль керування адміністраторами. Модуль дозволяє:

- створювати нових адміністраторів;
- змінювати ролі;
- активувати та деактивувати облікові записи;
- виконувати порівняння статистики адміністраторів.

Для спрощення додавання великої кількості товарів реалізовано модуль імпорту CSV-файлів. Даний модуль дозволяє автоматично завантажувати інформацію про товари до бази даних без ручного введення даних.

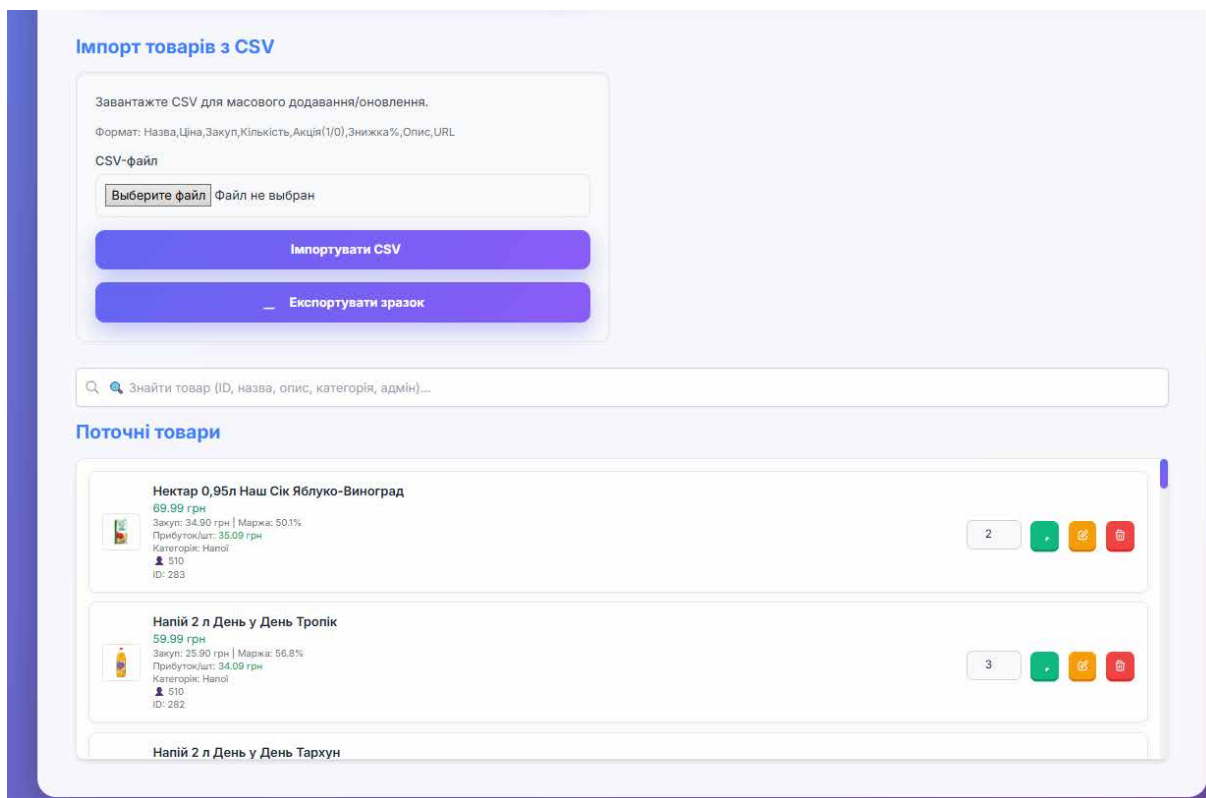


Рисунок 3.8 – Імпорт CSV

На рисунку 3.8 представлено імпорт товарів із CSV-файлу.

Інтерфейс системи адаптовано для роботи як на комп'ютерах, так і на мобільних пристроях.

4 РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЇ СИСТЕМИ

4.1 Тестування системи

Після завершення реалізації програмних модулів інформаційної системи було проведено тестування працездатності програмного забезпечення.

Для перевірки роботи системи було сформовано набір тестових сценаріїв.

Таблиця 4.1 – Результати тестування основних функцій системи

	Тестовий сценарій	Очікуваний результат	Результат
	Відкриття каталогу товарів	Відображення списку товарів	Виконано успішно
	Додавання товару до кошика	Товар додається до кошика	Виконано успішно
	Видалення товару з кошика	Товар видаляється з кошика	Виконано успішно
	Оформлення замовлення	Дані замовлення зберігаються у БД	Виконано успішно
	Авторизація адміністратора	Вхід до адмін-панелі	Виконано успішно
	Додавання нового товару	Новий товар з'являється у каталозі	Виконано успішно
	Редагування товару	Дані товару оновлюються	Виконано успішно
	Отримання даних через API	Сервер повертає JSON-відповідь	Виконано успішно

Виявлені незначні помилки інтерфейсу та обробки даних були усунені на етапі налагодження програмного забезпечення.

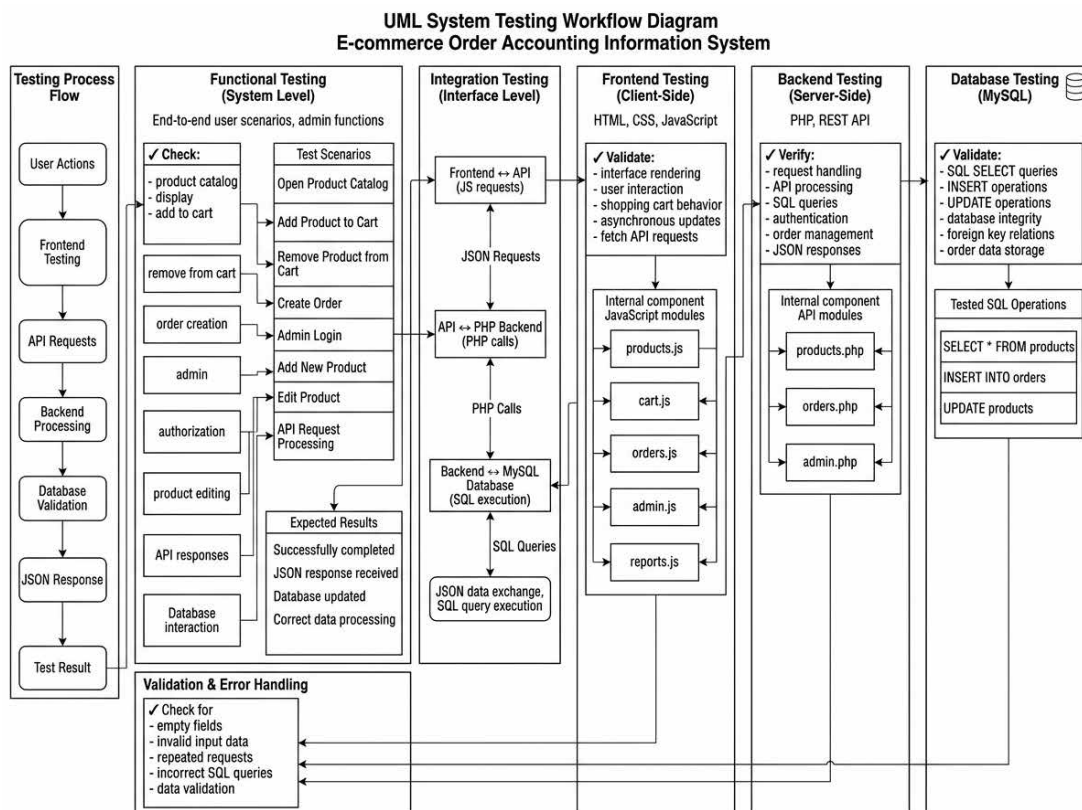


Рисунок 4.1 – Тестування замовлення

На рисунку 4.1 показано процес тестування оформлення замовлення.

Під час тестування також перевірялася швидкість надсилання Telegram-сповіщень після оформлення замовлення. У результаті перевірки встановлено, що повідомлення надходять адміністратору практично миттєво, що дозволяє оперативно обробляти нові замовлення та контролювати роботу системи.

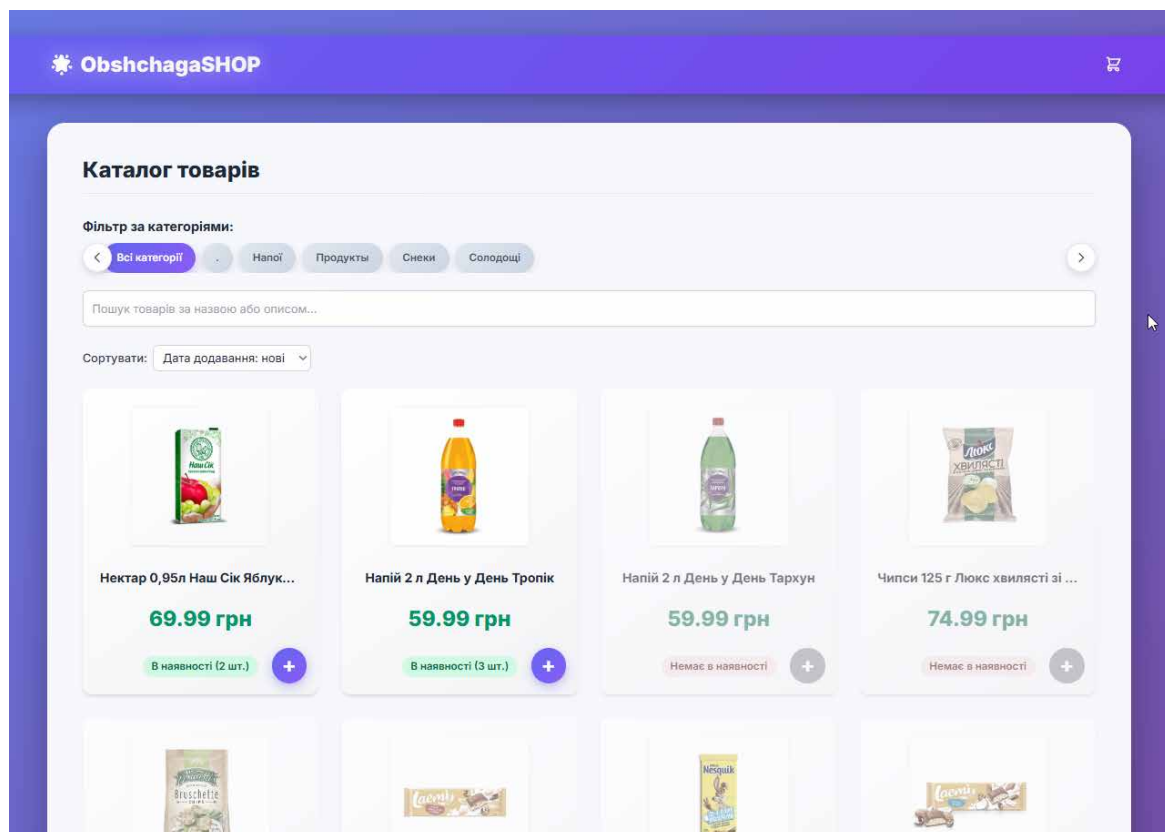


Рисунок 4.2 – Головна сторінка

На рисунку 4.2 представлено головну сторінку системи.

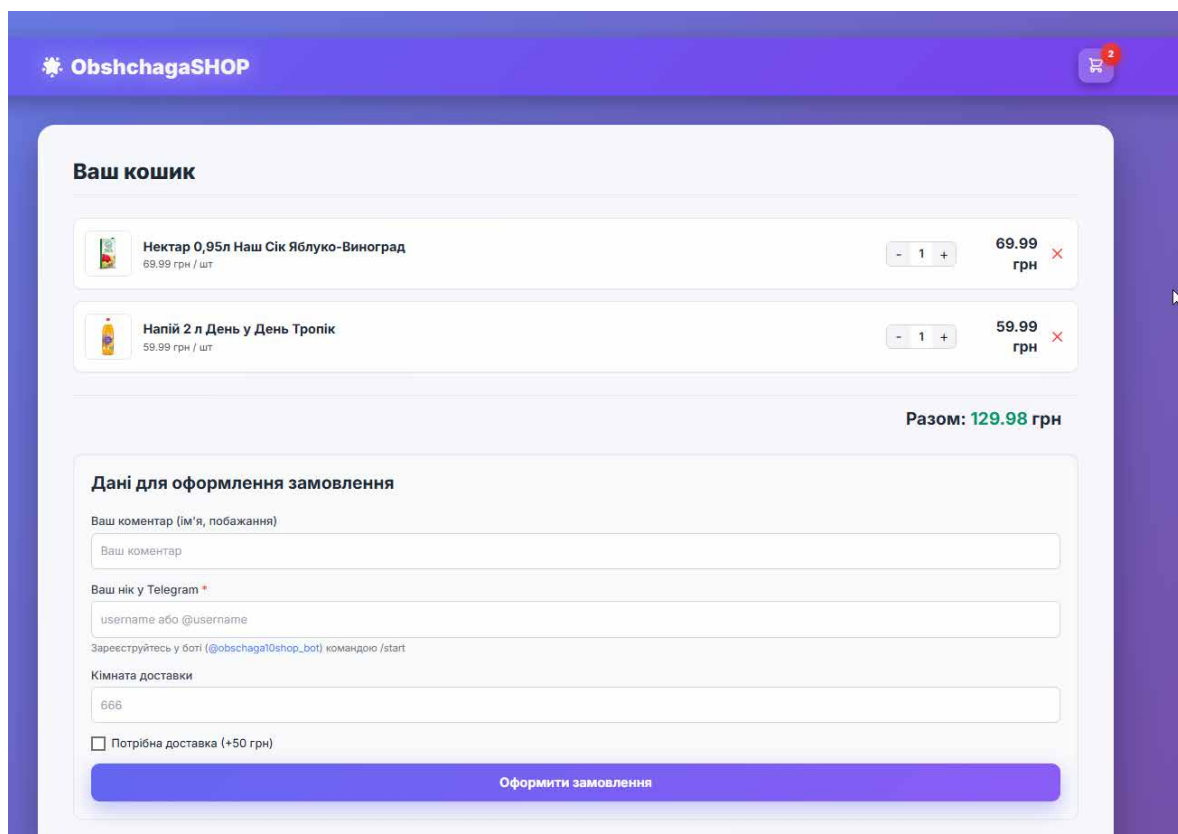


Рисунок 4.3 – Оформлення замовлення

На рисунку 4.3 представлено форму оформлення замовлення.

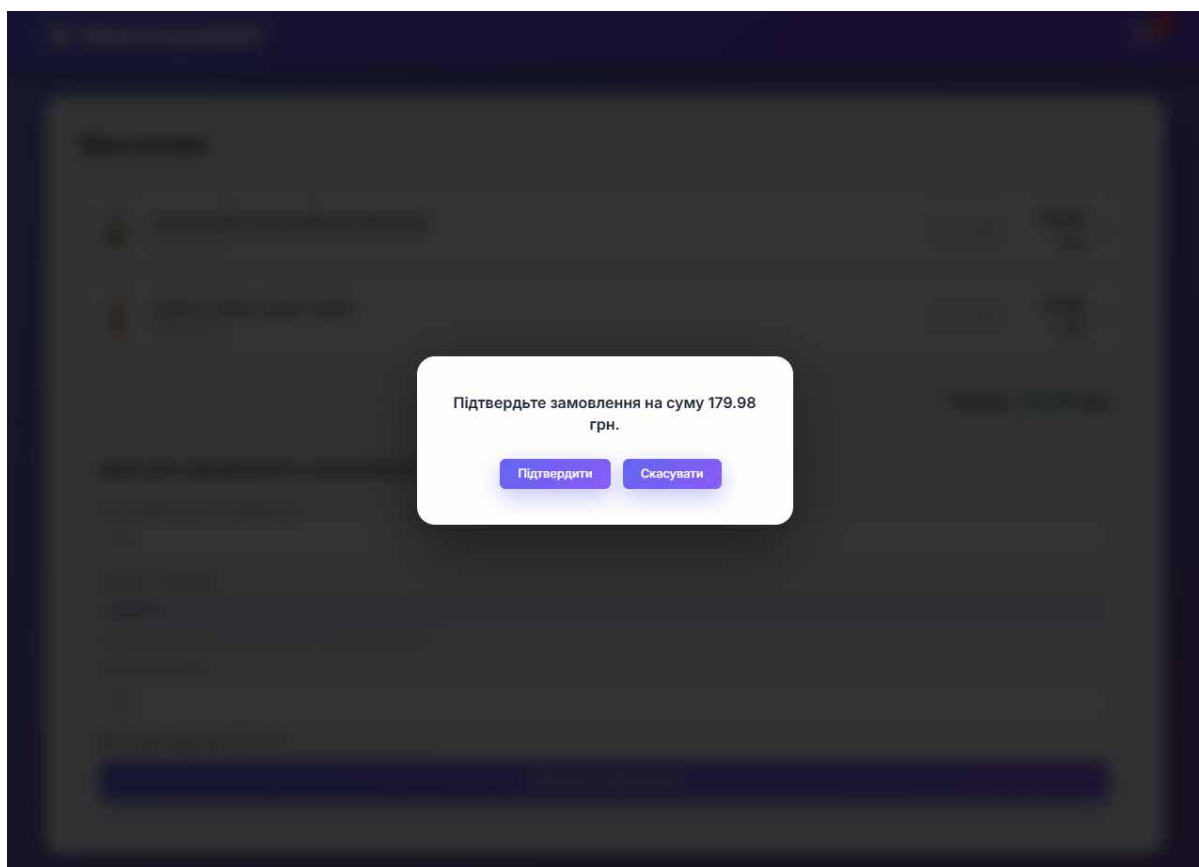


Рисунок 4.4 – Підтвердження замовлення

На рисунку 4.4 представлено підтвердження замовлення.

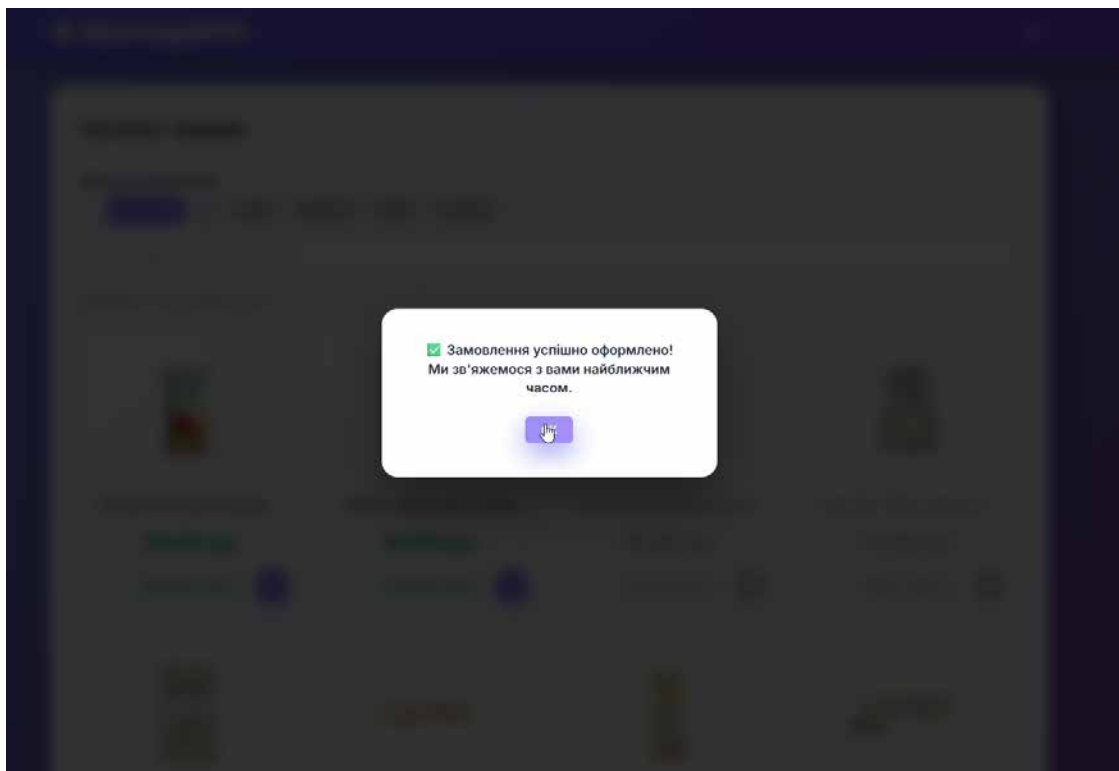


Рисунок 4.5 – Повідомлення

На рисунку 4.5 представлено повідомлення про оформлення замовлення.

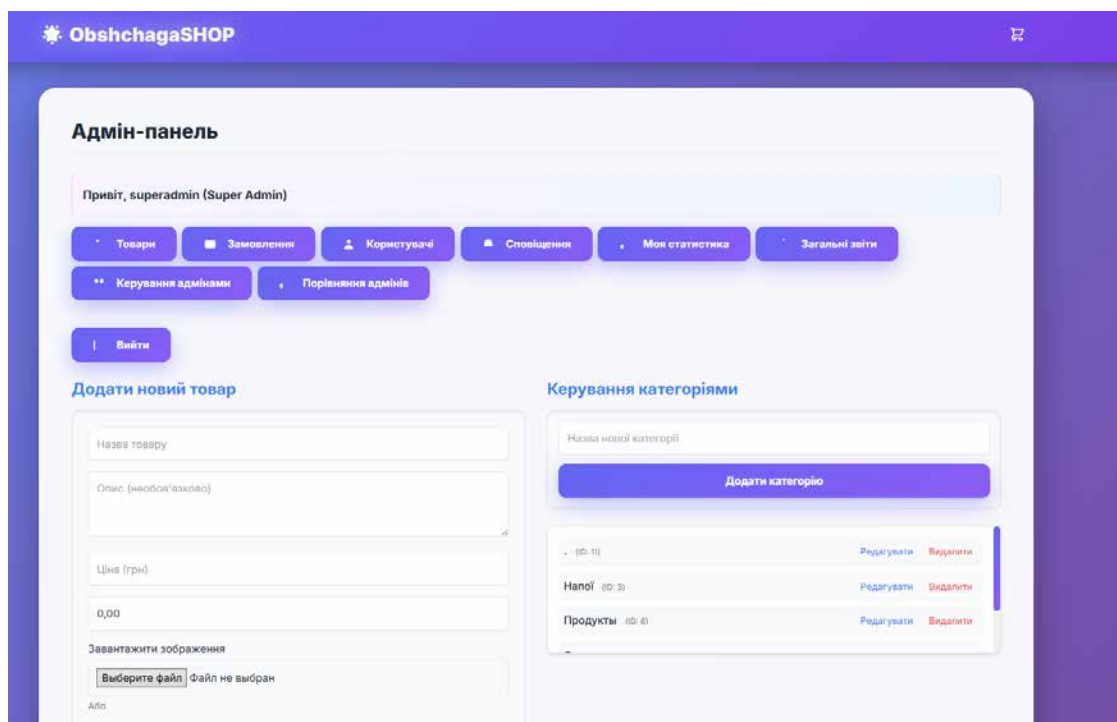


Рисунок 4.6 – Адмін-панель

На рисунку 4.6 представлено адміністративну панель системи.

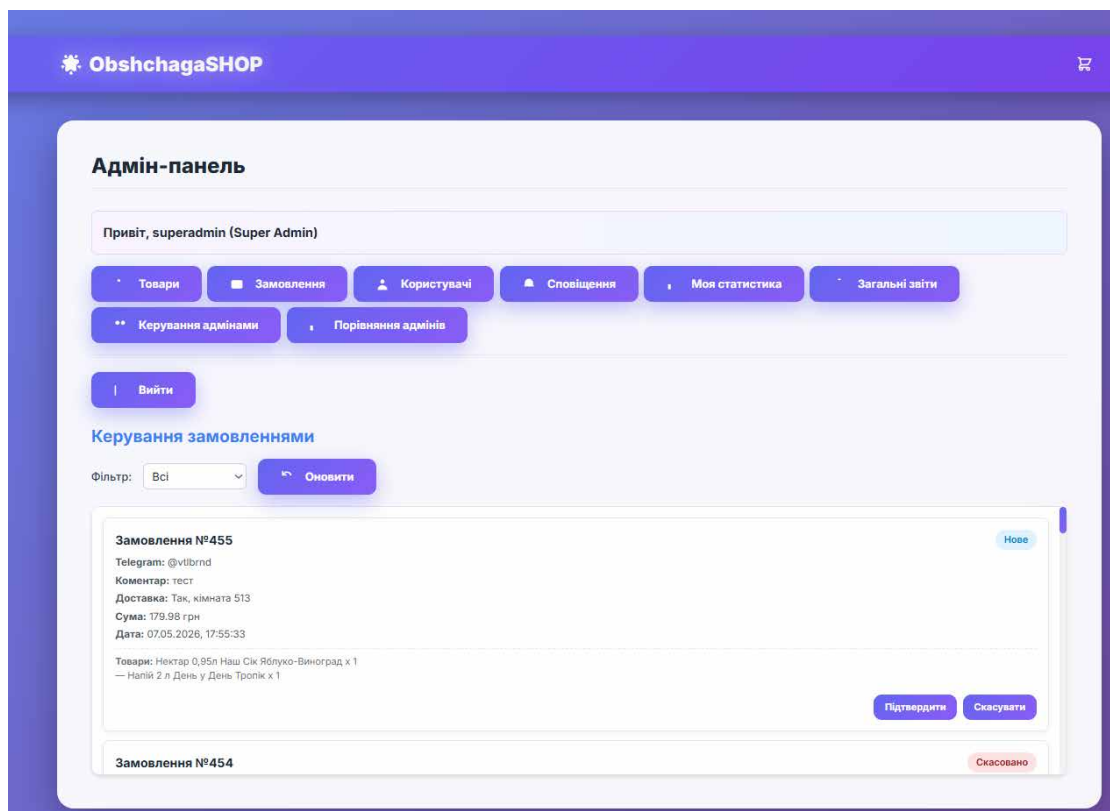


Рисунок 4.7 – Замовлення

На рисунку 4.7 представлено модуль управління замовленнями.

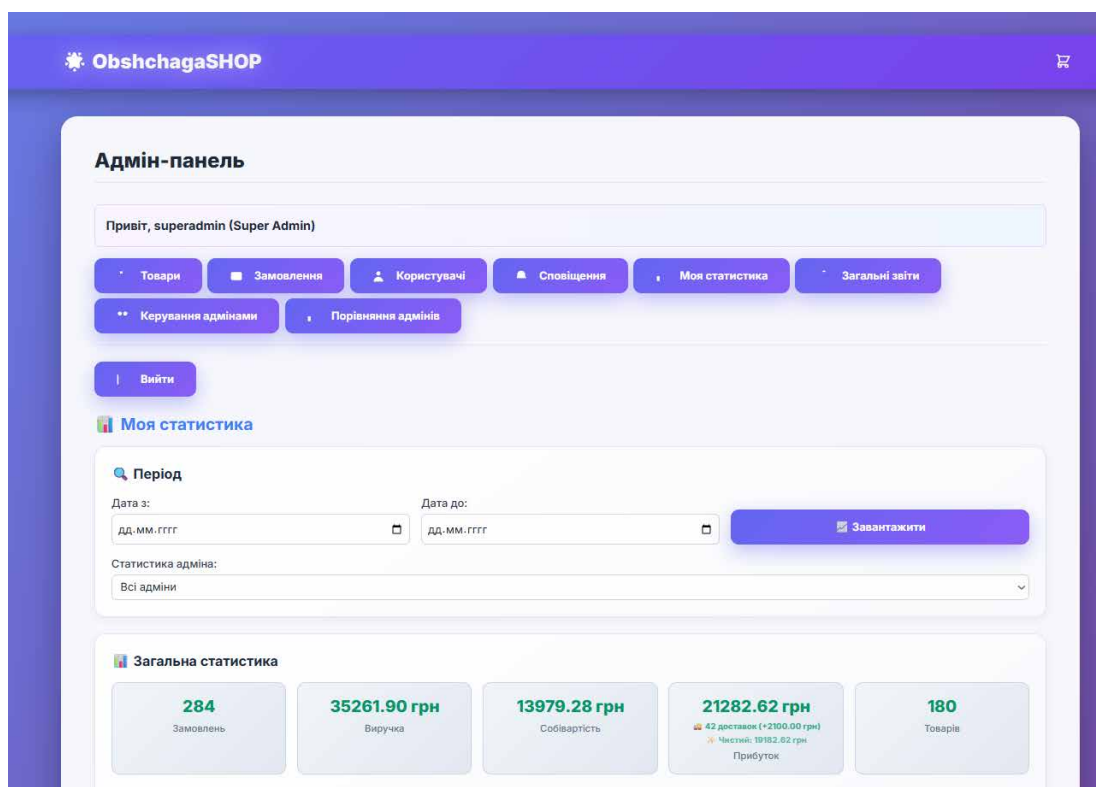


Рисунок 4.8 – Статистика

На рисунку 4.8 представлено модуль статистики.

4.2 Вимоги до апаратного та програмного забезпечення

Для забезпечення стабільної роботи інформаційної системи обліку замовлень товарів необхідно дотримуватися визначених вимог до апаратного та програмного забезпечення.

Рекомендовані апаратні характеристики серверного обладнання наведено у таблиці 4.2.

Таблиця 4.2 – Рекомендовані апаратні характеристики серверної частини

Компонент	Мінімальні вимоги	Рекомендовані вимоги
Процесор	2 ядра, 2.0 GHz	4 ядра, 3.0 GHz
Оперативна пам'ять	4 GB RAM	8 GB RAM
Накопичувач	10 GB HDD	SSD 50 GB
Мережа	100 Mbps	1 Gbps

Для функціонування серверної частини програмного забезпечення використовуються такі програмні компоненти:

Таблиця 4.3 – Програмне забезпечення системи

Програмний компонент	Призначення
Apache	Веб-сервер
PHP	Серверна мова програмування
MySQL	Система управління базою даних
Visual Studio Code	Середовище розробки

Google Chrome	Тестування веб-застосунку
---------------	---------------------------

Під час тестування інформаційної системи було підтверджено коректну роботу програмного забезпечення на персональному комп'ютері із такими характеристиками:

Таблиця 4.4 – Конфігурація тестового середовища

Компонент	Характеристика
Операційна система	Windows 10
Процесор	Intel Core i5
Оперативна пам'ять	8 GB RAM
Веб-сервер	Apache
PHP	Версія 8.x
MySQL	Версія 8.x
Браузер	Google Chrome

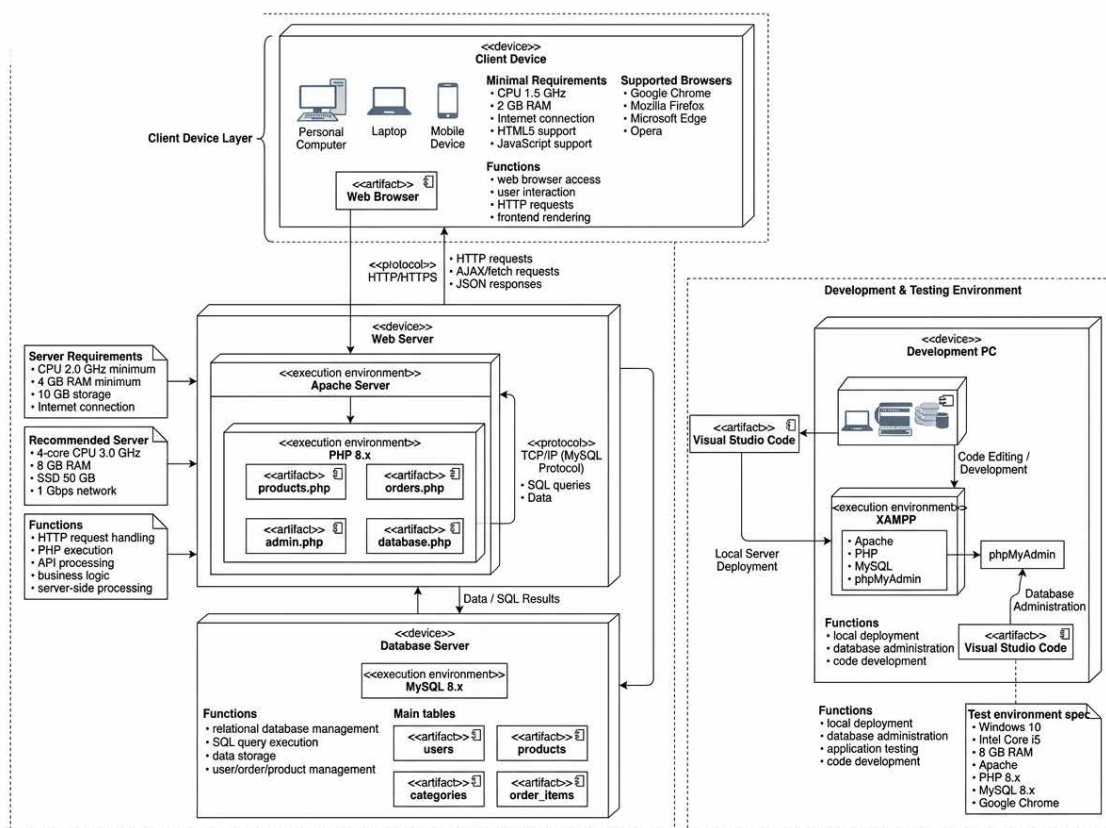


Рисунок 4.9 – Розгортання системи

На рисунку 4.9 представлено схему розгортання системи.

Під час тестування веб-застосунок коректно працював як у браузері Google Chrome, так і в Mozilla Firefox. Також було перевірено адаптивність інтерфейсу на мобільних пристроях у тому числі в інтерфейсі Telegram .

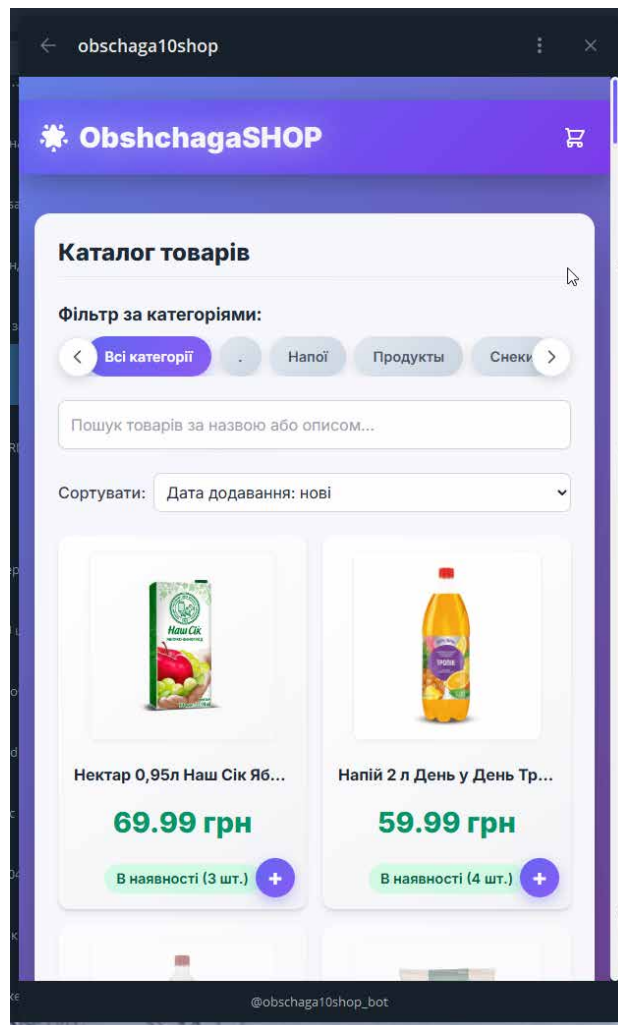


Рисунок 4.10 – Telegram WebApp

На рисунку 4.10 представлено інтерфейс Telegram WebApp.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було створено веб-застосунок для обліку замовлень товарів в інтернет-магазині. Реалізовано функції роботи з товарами, оформлення замовлень, адміністрування та Telegram-сповіщень.

У процесі розробки було спроектовано структуру бази даних, реалізовано клієнтську та серверну частини веб-застосунку, а також інтеграцію з Telegram Bot API для автоматичного надсилання повідомлень про нові замовлення.

Інформаційна система підтримує управління товарами, статистику продажів, модуль адміністрування та імпорт товарів із CSV-файлів. Для реалізації програмного забезпечення використовувалися HTML, CSS, JavaScript, PHP та MySQL [4–9].

Розроблена система може використовуватися для автоматизації оформлення замовлень в інтернет-магазині. У подальшому можливе розширення функціоналу, зокрема додавання онлайн-оплати та мобільної версії застосунку.

Система має перспективи подальшого розвитку. У майбутньому можливе впровадження онлайн-оплати, мобільної версії застосунку та розширення функціоналу статистики продажів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Басюк Т. М. Проєктування інформаційних систем : навчальний посібник. – Львів : Видавництво Львівської політехніки, 2021. – 348 с.
2. Гужва В. М. Інформаційні системи і технології на підприємствах : навчальний посібник. – Київ : КНЕУ, 2020. – 400 с.
3. Морзе Н. В., Барна О. В. Основи веб-програмування : навчальний посібник. – Київ : BHV, 2021. – 320 с.
4. PHP Documentation [Електронний ресурс]. – Режим доступу: <https://www.php.net/> (дата звернення: 15.05.2026).
5. MySQL Documentation [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/> (дата звернення: 15.05.2026).
6. Mozilla Developer Network Web Docs [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/> (дата звернення: 15.05.2026).
7. HTML Living Standard [Електронний ресурс]. – Режим доступу: <https://html.spec.whatwg.org/> (дата звернення: 15.05.2026).
8. CSS: Cascading Style Sheets [Електронний ресурс]. – Режим доступу: <https://www.w3.org/Style/CSS/> (дата звернення: 15.05.2026).
9. JavaScript Guide [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (дата звернення: 15.05.2026).
10. Date К. Дж. Введення в системи баз даних. – Київ : Діалектика, 2020. – 816 с.
11. Фаулер М. Архітектура корпоративних програмних застосунків. – Київ : Вільямс, 2019. – 544 с.
12. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. – Boston : Addison-Wesley, 1995. – 395 p.
13. Pressman R. Software Engineering: A Practitioner's Approach. – New York : McGraw-Hill Education, 2019. – 880 p.

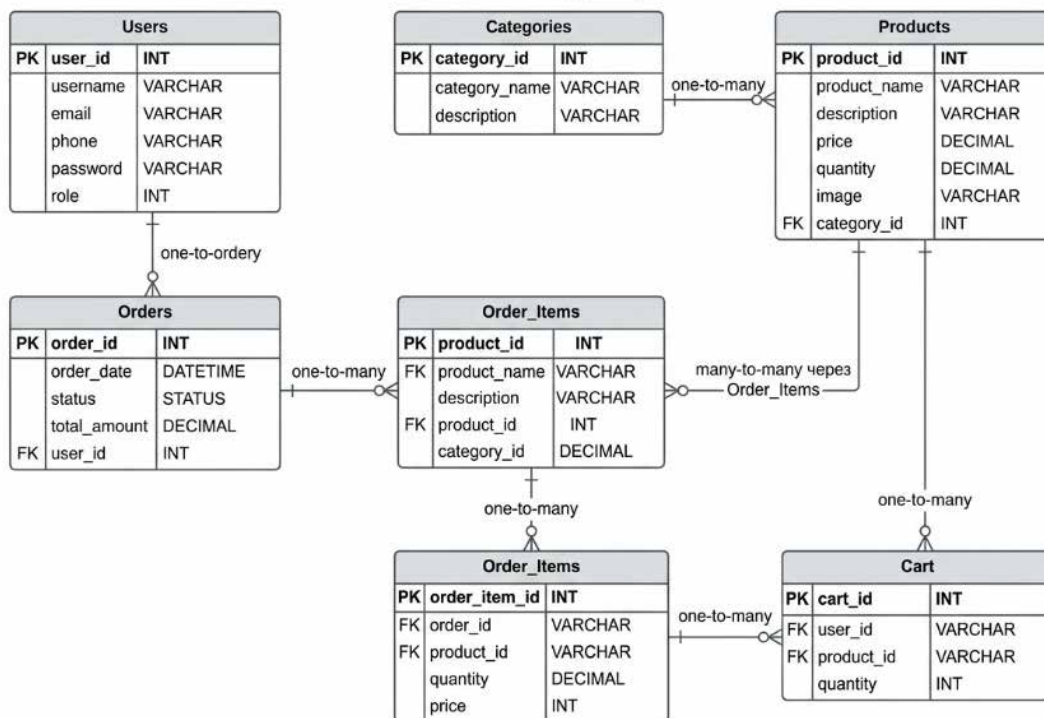
14. Sommerville I. Software Engineering. – Boston : Pearson Education, 2020. – 810 p.
15. Welling L., Thomson L. PHP and MySQL Web Development. – Boston : Addison-Wesley, 2021. – 768 p.
16. Silberschatz A., Korth H., Sudarshan S. Database System Concepts. – New York : McGraw-Hill, 2020. – 1376 p.
17. Visual Studio Code Documentation [Электронный ресурс]. – Режим доступа: <https://code.visualstudio.com/docs> (дата звернения: 15.05.2026).
18. Apache HTTP Server Documentation [Электронный ресурс]. – Режим доступа: <https://httpd.apache.org/docs/> (дата звернения: 15.05.2026).
19. XAMPP Documentation [Электронный ресурс]. – Режим доступа: <https://www.apachefriends.org/> (дата звернения: 15.05.2026).
20. Telegram Bot API [Электронный ресурс]. – Режим доступа: <https://core.telegram.org/bots/api> (дата звернения: 15.05.2026).
21. Shopify [Электронный ресурс]. – Режим доступа: <https://www.shopify.com/> (дата звернения: 15.05.2026).
22. OpenCart [Электронный ресурс]. – Режим доступа: <https://www.opencart.com/> (дата звернения: 15.05.2026).
23. WooCommerce [Электронный ресурс]. – Режим доступа: <https://woocommerce.com/> (дата звернения: 15.05.2026).
24. Magento Adobe Commerce [Электронный ресурс]. – Режим доступа: <https://business.adobe.com/products/magento/magento-commerce.html> (дата звернения: 15.05.2026).

ДОДАТКИ

Додаток А

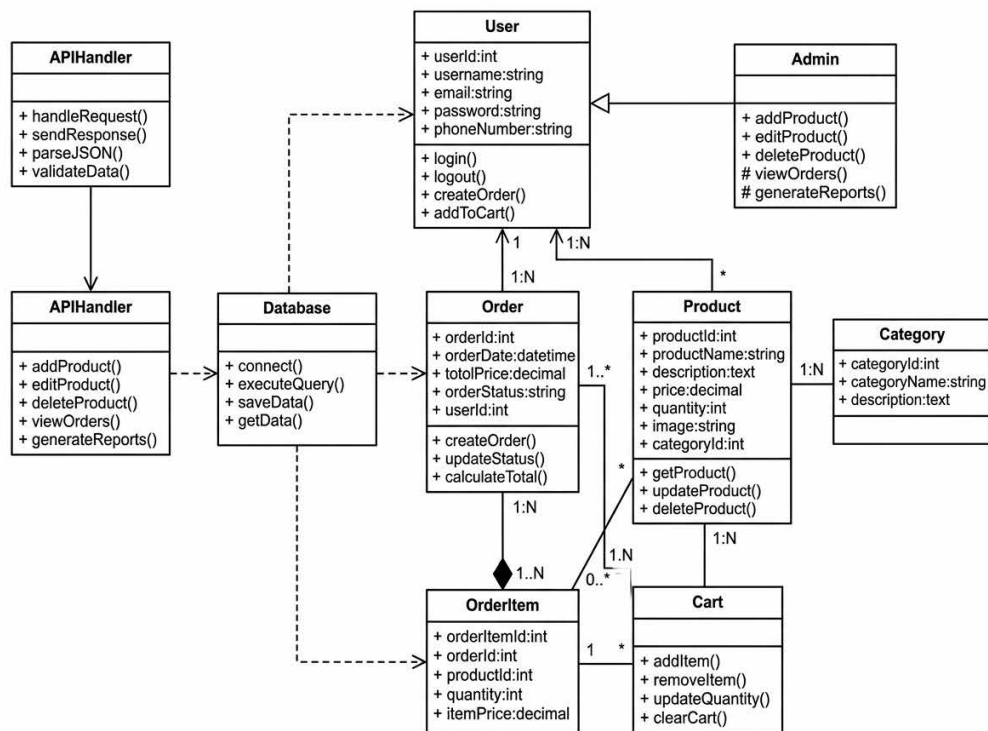
ER-діаграма бази даних інформаційної системи

**ER-діаграма системи обліку замовлень інтернет-магазину
(Логічна модель)**



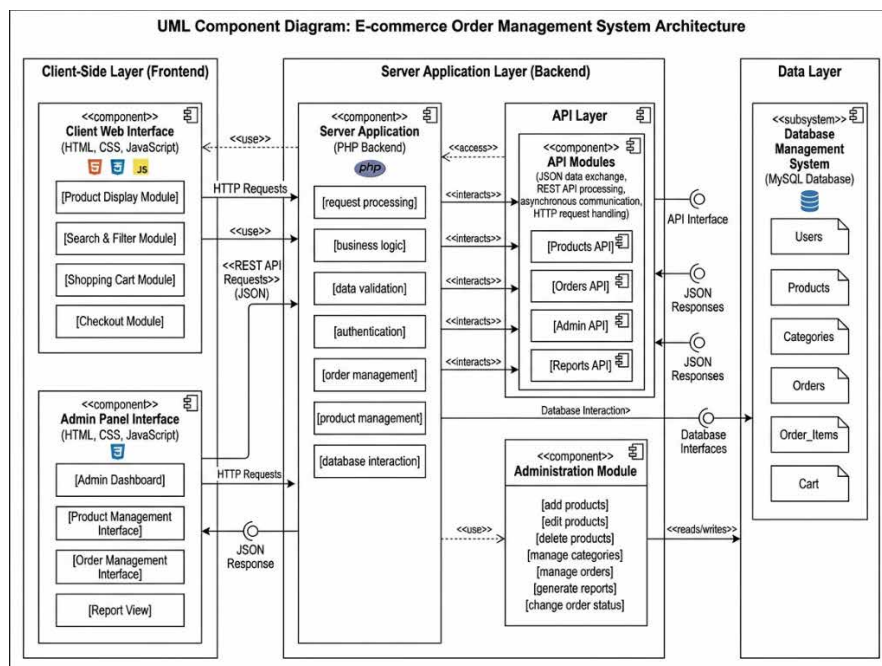
Додаток Б

Діаграма класів програмного забезпечення



Додаток В

Діаграма компонентів системи



Додаток Г

Структура бази даних MySQL

```

CREATE TABLE users (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(100),
  email VARCHAR(100),
  password VARCHAR(255),
  role VARCHAR(20)
);
  
```

```

CREATE TABLE products (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(255),
  description TEXT,
  price DECIMAL(10,2),
  quantity INT,
  );
  
```

```

        image VARCHAR(255),
        category_id INT
    );

```

Додаток Д

Фрагменти SQL-запитів

```
SELECT * FROM products;
```

```

INSERT INTO products(name, price, quantity)
VALUES ('Laptop', 25000, 5);

```

```

UPDATE products
SET quantity = 10
WHERE id = 1;

```

Додаток Е

Фрагменти програмного коду PHP

```

<?php
include 'database.php';

```

```

$sql = "SELECT * FROM products";
$result = mysqli_query($conn, $sql);

```

```
$data = [];
```

```
while($row = mysqli_fetch_assoc($result)) {
```

```

    $data[] = $row;
}

echo json_encode($data);
?>

```

Додаток Ж

Фрагменти програмного коду JavaScript

```

fetch('api/products.php')
    .then(response => response.json())
    .then(products => {
        console.log(products);
    });

```

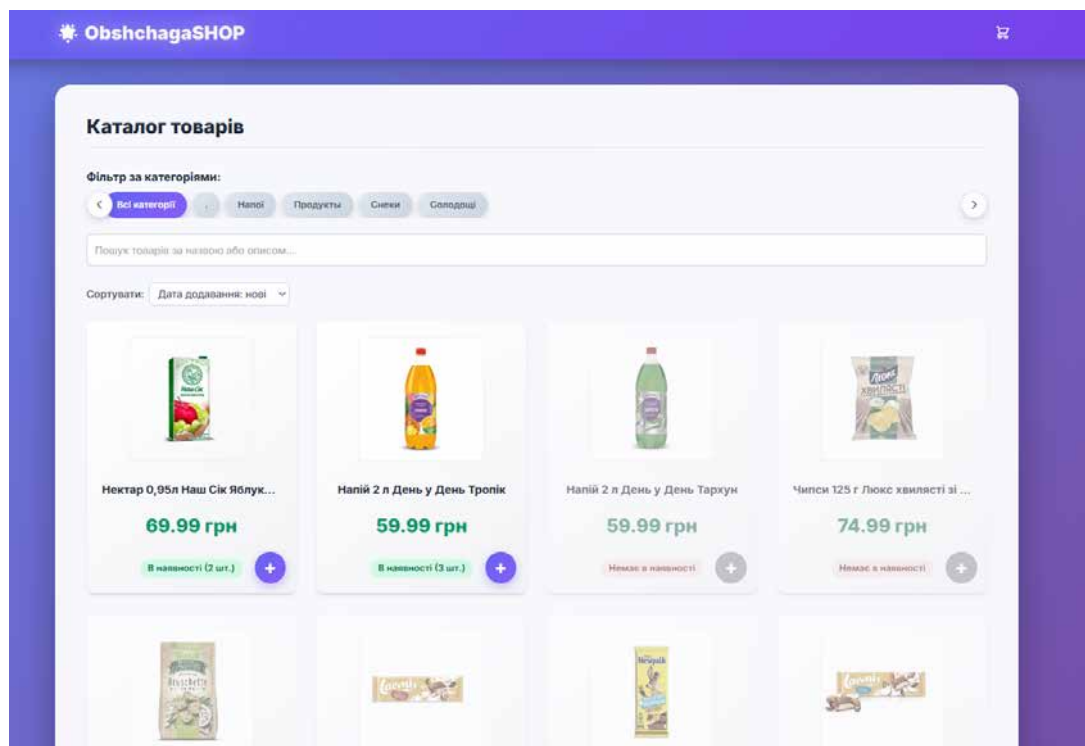
```

function addToCart(id) {
    console.log("Product added:", id);
}

```

Додаток З

Інтерфейс користувача інформаційної системи



Додаток И

Під час виконання бакалаврської кваліфікаційної роботи використовувалися інструменти штучного інтелекту ChatGPT та BlackBoxAI.

ChatGPT використовувався для:

- покращення стилістики тексту;
- генерації окремих фрагментів опису програмних модулів;
- перевірки граматики та структури тексту;
- формування прикладів SQL-запитів.

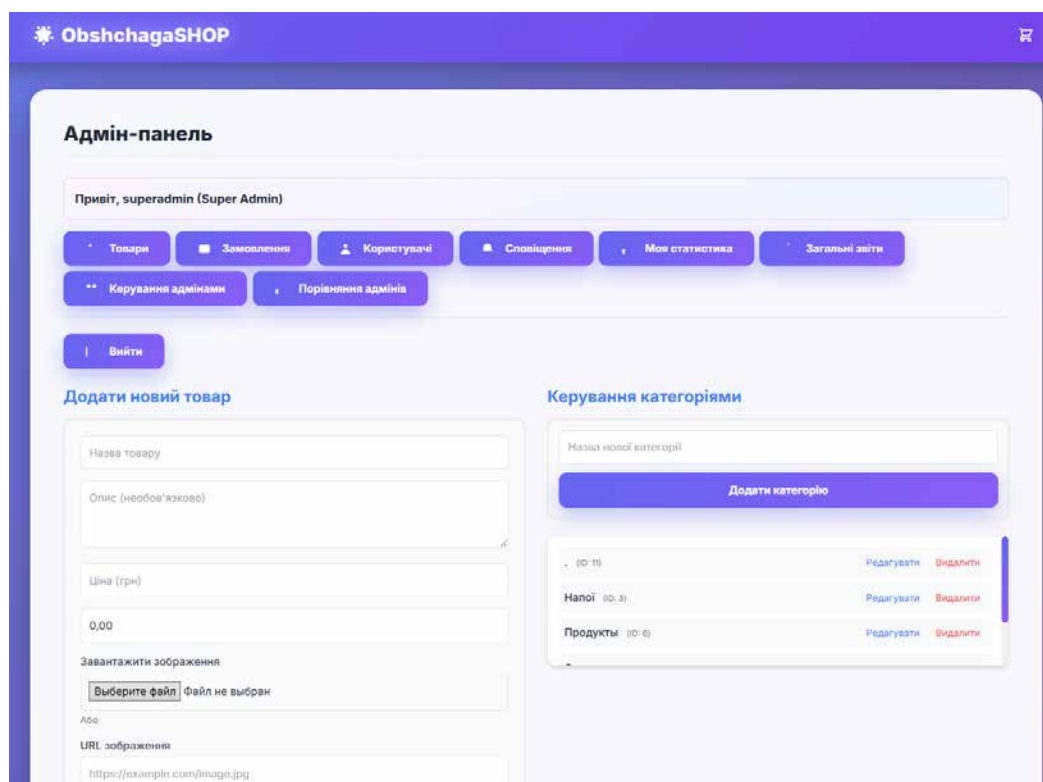
BlackBoxAI використовувався для:

- автоматичного доповнення програмного коду;
- генерації окремих фрагментів JavaScript та PHP-коду.

Усі результати, отримані за допомогою інструментів штучного інтелекту, були перевірені, відредаговані та адаптовані відповідно до тематики кваліфікаційної роботи.

Додаток К

Інтерфейс адміністративної панелі



Додаток Л

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ

І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

ДЕКЛАРАЦІЯ ЗДОБУВАЧА

Я, Конончук Віталій Сергійович, здобувач(ка) НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота (проєкт) на тему «Інформаційна система обліку замовлень товарів в інтернет-магазині» є результатом моєї особистої праці.
2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.
3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне):
 - ☒ інструменти генеративного ШІ не використовувалися.
 - ☒ інструменти ChatGPT, Gemini , BlackBoxAI були використані для:
 - ☐ перекладу іншомовних джерел;
 - ☒ стилістичного редагування та перевірки граматики;
 - ☒ допомоги у написанні/відлагодженні програмного коду;
 - ☒ інше: тестування безпеки розробленого продукту .

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

« ____ » _____
2026 р.

КОНОНЧУК
(підпис)

Віталій