МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»

Циклова комісія математичних дисциплін та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «АРХІТЕКТУРА ТА РОЗРОБКА ВЕБ-СИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ДЛЯ КОМП'ЮТЕРНИХ КОМПОНЕНТІВ»

Виконав: здобувач освіти 4 курсу, групи К121-22
Спеціальність 121 Інженерія
програмного забезпечення
(шифр і спеціальність)
Дмитро ХАЛАЧ
(ім'я та ПРІЗВИЩЕ)
Керівник Дмитро ДРАЄВСЬКИЙ
(ім'я та ПРІЗВИЩЕ)
Рецензент доцент кафедри програмної інженерії ЗНУ, к.т.н., доцент Віталій МУХІН
(посада, вчене звання, науковий ступінь, ім'я та ПРІЗВИЩЕ)

Запоріжжя
2026

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ВСП «Економіко-правничий фаховий коледж ЗНУ»

Освітньо–кваліфікаційний ступінь фаховий молодший бакалавр

Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Голова циклової комісії математичних дисциплін та інформаційних технологій

_____ Тетяна СМОЛЯНКОВА
(підпис)

“ 05 ” червня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ХАЛАЧУ Дмитру Олександровичу
(ПРІЗВИЩЕ, ім’я та по– батькові)

1. Тема роботи “Архітектура та розробка веб-системи електронної комерції для комп’ютерних компонентів”

Керівник роботи ДРАЄВСЬКИЙ Дмитро Сергійович
(науковий ступінь, вчене звання, прізвище, ім’я, по батькові)

Затверджені наказом ВСП ЕПФК ЗНУ від «01» грудня 2025 р. № 1728-с

2. Строк подання студентом роботи 05.06.2026

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

4. Зміст розрахунково– пояснювальної записки (перелік питань, які потрібно розробити)
1.Аналіз предметної області та вибір стеку технологій
2. Проєктування бази даних та архітектури системи
3. Програмна реалізація вебсистеми

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація до захисту

6. Консультанти розділів роботи

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

7. Дата видачі завдання 01.12.2025

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи:		
	постановка задачі	Грудень 2025	виконано
2.	Збір вихідних даних, обробка методичних		
	та теоретичних джерел	Січень 2026	виконано
3.	Розробка першого розділу:		
	огляд інформаційних систем та технологій для їхнього створення	Лютий 2026	виконано
4.	Розробка другого розділу:		
	розробка проєкту інформаційної системи та вибір технологій	Квітень 2026	виконано
5.	Розробка третього розділу:		
	розробка застосунку	Травень 2026	виконано
6.	Оформлення і нормоконтроль кваліфікаційної роботи та перевірка на плагіат	Червень 2026	виконано
7.	Захист кваліфікаційної роботи	16.06.2026	виконано

Здобувач освіти _____ Дмитро ХАЛАЧ
(підпис) (ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Дмитро ДРАЄВСЬКИЙ
(підпис) (ім'я ПРІЗВИЩЕ)

Нормоконтроль пройдено

Нормоконтролер _____ Юлія БОРИСОВСЬКА
(підпис) (ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Кваліфікаційна робота: 48 сторінок, 12 ілюстрацій, 5 таблиць, 1 додаток, 12 джерел.

Об’єкт дослідження: процес проєктування, розробки та використання вебсистем електронної комерції для продажу комп’ютерних компонентів.

Предмет дослідження: вебсистема електронної комерції для комп’ютерних комплектуючих з функціями перегляду каталогу, оформлення замовлень, обліку товарів та адміністрування.

Мета роботи: проєктування та програмна реалізація вебсистеми електронної комерції, яка забезпечує зручну взаємодію користувача з каталогом комп’ютерних компонентів, формування кошика, оформлення замовлень та керування товарами через адміністративну частину.

Методи дослідження: аналітичний метод, порівняльний аналіз, методи об’єктно-орієнтованого проєктування, методи проєктування баз даних, експериментальне тестування програмного забезпечення.

У кваліфікаційній роботі розглянуто процес створення веборієнтованої інформаційної системи для електронної комерції у сфері продажу комп’ютерних комплектуючих. Під час виконання роботи було проаналізовано предметну область, визначено основні вимоги до програмного продукту, спроєктовано структуру бази даних та розроблено основні функціональні модулі системи.

Для реалізації програмного продукту використано платформу Node.js, мову програмування TypeScript, вебфреймворк Express, шаблонізатор EJS та систему управління базами даних MySQL. Клієнтська частина системи побудована з використанням HTML5, CSS3 та Bootstrap 5.

У межах роботи реалізовано каталог товарів із можливістю сортування, механізм додавання товарів до кошика, оформлення замовлення, реєстрацію та авторизацію користувачів, особистий кабінет з історією замовлень, а також

приховану адміністративну панель для додавання, редагування та видалення товарів. Для підвищення надійності роботи системи процес оформлення замовлення реалізовано з використанням SQL-транзакцій, що дозволяє коректно створювати замовлення та оновлювати залишки товарів на складі.

Результати роботи можуть бути використані як основа для автоматизації діяльності інтернет-магазину комп’ютерних комплектуючих. Розроблена система спрощує процес обробки замовлень, забезпечує централізований облік товарів і надає адміністратору інструменти для керування асортиментом.

ВЕБСИСТЕМА, ЕЛЕКТРОННА КОМЕРЦІЯ, КОМП’ЮТЕРНІ КОМПЛЕКТУЮЧІ, NODE.JS, TYPESCRIPT, EXPRESS, EJS, MYSQL, БАЗА ДАНИХ

SUMMARY

Qualification Work: 48 pages, 12 figures, 5 tables, 1 appendix, 12 references.

Object of Study: The process of designing, developing, and using e-commerce web systems for selling computer components.

Subject of Study: An e-commerce web system for computer components with functions for catalog browsing, order processing, product accounting, and administration.

Purpose of the Work: To design and implement an e-commerce web system that provides convenient user interaction with a catalog of computer components, shopping cart formation, order processing, and product management through the administrative part of the system.

Research Methods: Analytical method, comparative analysis, object-oriented design methods, database design methods, and experimental software testing.

The qualification work examines the process of creating a web-oriented information system for e-commerce in the field of selling computer components. During the work, the subject area was analyzed, the main requirements for the software product were defined, the database structure was designed, and the main functional modules of the system were developed.

The software product was implemented using the Node.js platform, the TypeScript programming language, the Express web framework, the EJS template engine, and the MySQL database management system. The client side of the system was built using HTML5, CSS3, and Bootstrap 5.

As part of the work, a product catalog with sorting functionality, a mechanism for adding products to the shopping cart, order processing, user registration and authorization, a personal account with order history, and a hidden administrative panel for adding, editing, and deleting products were implemented. To improve the reliability of the system, the order processing procedure was implemented using

SQL transactions, which allows orders to be created correctly and product stock balances to be updated.

The results of the work can be used as a basis for automating the operation of an online store for computer components. The developed system simplifies the order processing procedure, provides centralized product accounting, and gives the administrator tools for managing the product range.

WEB SYSTEM, E-COMMERCE, COMPUTER COMPONENTS, NODE.JS, TYPESCRIPT, EXPRESS, EJS, MYSQL, DATABASE

ЗМІСТ

Реферат 4
Summary 6
Вступ 9
1 Аналіз предметної області та вибір технологій 12
1.1 Аналіз предметної області вебсистем електронної комерції 12
1.2 Аналіз існуючих аналогів інформаційних систем 14
1.3 Постановка задачі та формування вимог до системи 15
1.4 Обґрунтування вибору технологій розробки 18
2 Проєктування архітектури та бази даних системи 21
2.1 Проєктування функціональних можливостей системи 21
2.2 Проєктування структури бази даних 23
2.3 Проєктування архітектури вебдодатка 24
2.4 Проєктування взаємодії компонентів системи 25
3 Програмна реалізація вебсистеми електронної комерції 27
3.1 Налаштування серверної частини та маршрутизації 27
3.2 Реалізація роботи з каталогом товарів 29
3.3 Реалізація кошика та оформлення замовлення 31
3.4 Реалізація реєстрації, авторизації та особистого кабінету 35
3.5 Реалізація адміністративної панелі 37
3.6 Тестування роботи системи 39
Висновки 42
Перелік використаних джерел 44
Додаток А 45

ВСТУП

Електронна комерція є одним із найпоширеніших напрямів використання сучасних інформаційних технологій у сфері торгівлі. Інтернет-магазини дають змогу підприємствам швидше представляти товари покупцям, автоматизувати приймання замовлень, вести облік товарних залишків і зменшувати кількість ручних операцій. Для користувача така форма торгівлі є зручною, оскільки він може переглядати каталог, порівнювати товари, формувати кошик і оформлювати замовлення без відвідування фізичного магазину.

Особливо актуальною є розробка вебсистем електронної комерції для продажу комп’ютерних компонентів. Ця предметна область пов’язана з великою кількістю товарних позицій, різними характеристиками комплектуючих, постійною зміною цін і необхідністю контролю наявності товарів на складі. У разі використання ручного або частково автоматизованого обліку можуть виникати помилки під час оновлення залишків, обробки замовлень або керування асортиментом. Через це виникає потреба у створенні інформаційної системи, яка об’єднує каталог товарів, кошик, оформлення замовлень, облік користувачів і адміністративне керування товарами.

Актуальність теми полягає в необхідності автоматизації основних процесів роботи інтернет-магазину комп’ютерних комплектуючих. Розроблена вебсистема повинна забезпечувати зручну взаємодію користувача з каталогом товарів, давати можливість додавати товари до кошика, оформлювати замовлення, переглядати історію покупок, а також надавати адміністратору інструменти для керування асортиментом. Використання бази даних дозволяє централізовано зберігати інформацію про товари, користувачів і замовлення, а застосування транзакцій під час оформлення покупки підвищує надійність роботи системи.

Метою кваліфікаційної роботи є проєктування та програмна реалізація вебсистеми електронної комерції для продажу комп’ютерних компонентів, яка забезпечує перегляд каталогу товарів, роботу з кошиком, оформлення замовлень, авторизацію користувачів та адміністрування товарів.

Для досягнення поставленої мети необхідно виконати такі завдання:

– проаналізувати предметну область електронної комерції та особливості продажу комп’ютерних комплектуючих;

– розглянути існуючі підходи до створення інтернет-магазинів і визначити вимоги до розроблюваної системи;

– обґрунтувати вибір програмних засобів і технологій для реалізації вебдодатка;

– спроєктувати структуру бази даних для зберігання інформації про товари, користувачів, замовлення та складські залишки;

– розробити серверну частину системи з використанням Node.js, TypeScript та Express;

– реалізувати клієнтську частину вебдодатка з використанням EJS, HTML5, CSS3 та Bootstrap 5;

– створити функціонал каталогу товарів, кошика, оформлення замовлень, реєстрації та авторизації користувачів;

– реалізувати адміністративну панель для додавання, редагування та видалення товарів;

– перевірити працездатність основних функцій системи.

Об’єктом дослідження є процес проєктування та розробки вебсистем електронної комерції для автоматизації продажу товарів через інтернет.

Предметом дослідження є вебсистема електронної комерції для комп’ютерних комплектуючих, яка включає каталог товарів, кошик, модуль оформлення замовлень, авторизацію користувачів, особистий кабінет і адміністративну частину.

У роботі використано аналітичний метод для вивчення предметної області, порівняльний аналіз для оцінювання існуючих підходів до створення

вебсистем електронної комерції, методи проєктування баз даних для побудови структури зберігання інформації, а також експериментальний метод для перевірки роботи створеного програмного продукту.

Практичне значення роботи полягає в тому, що розроблена система може бути використана як основа для інтернет-магазину комп’ютерних комплектуючих. Вона дозволяє автоматизувати роботу з каталогом, спростити процес оформлення замовлень, забезпечити облік товарних залишків і надати адміністратору базові інструменти для керування товарами. Результати роботи можуть бути використані для подальшого розвитку системи, зокрема додавання пошуку, фільтрації за характеристиками, інтеграції з платіжними сервісами або службами доставки.

Результатом виконання кваліфікаційної роботи є вебдодаток, побудований за клієнт-серверною архітектурою. У системі реалізовано основні функції інтернет-магазину: перегляд товарів, сортування, додавання до кошика, оформлення замовлення, реєстрацію та вхід користувача, перегляд історії замовлень, а також керування товарами через адміністративну панель. Розробка виконана з використанням сучасних вебтехнологій, що забезпечує гнучкість структури проєкту та можливість його подальшого розширення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИБІР ТЕХНОЛОГІЙ

1.1 Аналіз предметної області вебсистем електронної комерції

Електронна комерція є одним із найважливіших напрямів розвитку сучасних інформаційних систем. Вона дозволяє підприємствам продавати товари через інтернет, автоматизувати приймання замовлень, вести облік клієнтів і контролювати залишки продукції на складі. Для покупця інтернет-магазин є зручним способом перегляду асортименту, вибору товару та оформлення замовлення без необхідності особистого відвідування торгової точки.

Предметна область продажу комп’ютерних комплектуючих має низку особливостей. До цієї категорії належать процесори, материнські плати, відеокарти, оперативна пам’ять, накопичувачі, блоки живлення, корпуси та інші компоненти. Такі товари можуть суттєво відрізнятися за характеристиками, ціною, наявністю та попитом серед покупців. Через це магазин комп’ютерних компонентів потребує зручної системи для зберігання та обробки інформації про товари.

У разі ручного або частково автоматизованого ведення обліку можуть виникати проблеми з актуальністю даних. Наприклад, працівник магазину може не встигнути вчасно оновити залишки товарів після продажу, через що користувач побачить у каталозі позицію, якої фактично немає на складі. Також ручне оформлення замовлень збільшує ризик помилок у даних клієнта, кількості товарів або сумі покупки.

Вебсистема електронної комерції дозволяє об’єднати основні процеси продажу в одному програмному середовищі. Користувач отримує доступ до каталогу товарів через вебінтерфейс, формує кошик і створює замовлення. Дані про товари, користувачів і замовлення зберігаються в базі даних, що забезпечує централізований облік інформації.

Порівняння ручного та автоматизованого підходу до організації продажу комп’ютерних комплектуючих наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння ручного та автоматизованого підходу до продажу комп’ютерних комплектуючих

Критерій	Ручний або частково автоматизований підхід	Вебсистема електронної комерції
Перегляд товарів	Покупець отримує інформацію від продавця або з окремих прайс-листів	Каталог товарів доступний через вебінтерфейс
Оформлення замовлення	Дані замовлення вносяться вручну	Замовлення формується через електронний кошик
Облік залишків	Потребує ручного оновлення після продажу	Залишки оновлюються після оформлення замовлення
Зберігання даних	Інформація може зберігатися в різних файлах або таблицях	Дані зберігаються централізовано в базі даних
Ризик помилок	Високий через людський фактор	Менший завдяки автоматизації основних операцій
Керування товарами	Потребує ручного редагування записів	Товари можна додавати, редагувати й видаляти через адміністративну панель

Дані таблиці 1.1 показують, що використання вебсистеми електронної комерції дозволяє зменшити кількість ручних операцій, спростити оформлення замовлень і централізувати зберігання інформації про товари, користувачів та покупки.

Для розроблюваної системи важливими є такі можливості: перегляд каталогу товарів, сортування позицій, додавання товарів до кошика, оформлення замовлення, реєстрація та авторизація користувача, перегляд історії покупок, а також керування товарами через адміністративну частину. Наявність цих функцій дозволяє розглядати систему як базову модель інтернет-магазину комп’ютерних комплектуючих.

1.2 Аналіз існуючих аналогів інформаційних систем

На ринку існує багато готових рішень для створення інтернет-магазинів. До них можна віднести OpenCart, WooCommerce, Shopify, Magento та інші системи керування електронною комерцією. Вони мають готові інструменти для створення каталогу, кошика, сторінок товарів і панелі адміністрування.

OpenCart є популярною системою для запуску інтернет-магазинів. Вона містить готову структуру каталогу, кошика, замовлень і адміністративної частини. Її перевагою є велика кількість модулів і шаблонів. Недоліком є те, що для зміни внутрішньої логіки часто потрібно адаптувати готові модулі або змінювати структуру системи.

WooCommerce є плагіном для WordPress і часто використовується для створення невеликих інтернет-магазинів. Він має зручну панель керування, багато тем оформлення та додаткових розширень. Водночас така система залежить від WordPress, сторонніх плагінів і тем, що може ускладнювати подальше доопрацювання або оптимізацію.

Shopify є хмарною платформою для електронної комерції. Вона дозволяє швидко створити магазин без самостійного налаштування серверної частини. Основним недоліком є обмежений контроль над внутрішньою логікою, сервером і базою даних. Крім того, використання платформи передбачає оплату тарифів і додаткових сервісів.

Magento орієнтована на великі проєкти з розвиненим каталогом і значною кількістю товарів. Вона має широкі можливості для налаштування, однак є складною для початкового освоєння та потребує більше серверних ресурсів.

Порівняння готових рішень для створення інтернет-магазину наведено в таблиці 1.2.

Таблиця 1.2 – Порівняння готових рішень для створення інтернет-магазину

Система	Переваги	Недоліки
OpenCart	Має готовий функціонал каталогу, кошика та замовлень	Для глибокої зміни логіки потрібне доопрацювання модулів
WooCommerce	Зручна панель керування, багато готових тем і плагінів	Залежить від WordPress і сторонніх розширень
Shopify	Швидкий запуск магазину, хмарна інфраструктура	Обмежений контроль над серверною частиною та базою даних
Magento	Потужні можливості для великих інтернет-магазинів	Складне налаштування та високі вимоги до ресурсів
Власна вебсистема	Повний контроль над структурою, логікою та базою даних	Потребує самостійного проєктування та розробки

На основі порівняння можна визначити, що готові платформи зручні для швидкого запуску магазину, однак власна розробка краще підходить для кваліфікаційної роботи. Вона дозволяє самостійно спроєктувати базу даних, реалізувати серверну логіку, налаштувати маршрутизацію, створити кошик, авторизацію користувачів та адміністративну частину. Такий підхід дає змогу продемонструвати практичні навички розробки вебдодатків і краще адаптувати систему до поставлених вимог.

1.3 Постановка задачі та формування вимог до системи

Основною задачею кваліфікаційної роботи є проєктування та розробка вебсистеми електронної комерції для продажу комп’ютерних компонентів. Система повинна забезпечувати базовий цикл роботи інтернет-магазину: перегляд товарів, додавання позицій до кошика, оформлення замовлення,

збереження даних у базі даних та керування товарами через адміністративну частину.

Розроблювана система орієнтована на дві основні групи користувачів: покупця та адміністратора. Покупець повинен мати можливість переглядати каталог комп'ютерних комплектуючих, ознайомлюватися з основною інформацією про товари, додавати потрібні позиції до кошика та оформлювати замовлення. Для зареєстрованого користувача передбачається можливість авторизації та перегляду особистого кабінету з історією попередніх замовлень.

Адміністратор повинен мати доступ до окремої частини системи, призначеної для керування асортиментом. Через адміністративну панель має виконуватися додавання товарів, редагування інформації про наявні позиції та видалення товарів з каталогу. Доступ до цієї частини має бути обмеженим, щоб звичайні користувачі не могли змінювати інформацію в каталозі.

Під час формування вимог до системи було визначено, що вебдодаток повинен забезпечувати відображення списку товарів у каталозі, сортування позицій за вибраним критерієм, додавання товарів до кошика, перегляд його вмісту та підрахунок загальної вартості замовлення. Також система має підтримувати оформлення замовлень, реєстрацію та авторизацію користувачів, доступ до особистого кабінету з історією покупок, а також надавати адміністратору можливість додавати, редагувати й видаляти товари. Уся інформація про товари, користувачів і замовлення повинна зберігатися в базі даних.

Крім функціональних вимог, для системи важливими є також нефункціональні вимоги. Інтерфейс вебдодатка повинен бути зрозумілим для користувача та зручним для виконання основних дій. Дані мають зберігатися централізовано в базі даних, а доступ до адміністративної частини повинен бути обмежений. Структура проєкту має дозволяти подальше розширення функціоналу.

Окрему увагу приділено процесу оформлення замовлення. Під час створення замовлення система повинна не лише зберегти інформацію про

покупку, а й оновити кількість товарів на складі. Для цього використовується транзакційний підхід, за якого створення замовлення, додавання товарів до складу замовлення та списання залишків виконуються як єдиний процес. Якщо на одному з етапів виникає помилка, зміни не повинні зберігатися частково, оскільки це може призвести до некоректних даних у базі.

Схему бізнес-процесу оформлення замовлення в системі наведено на рисунку 1.1.

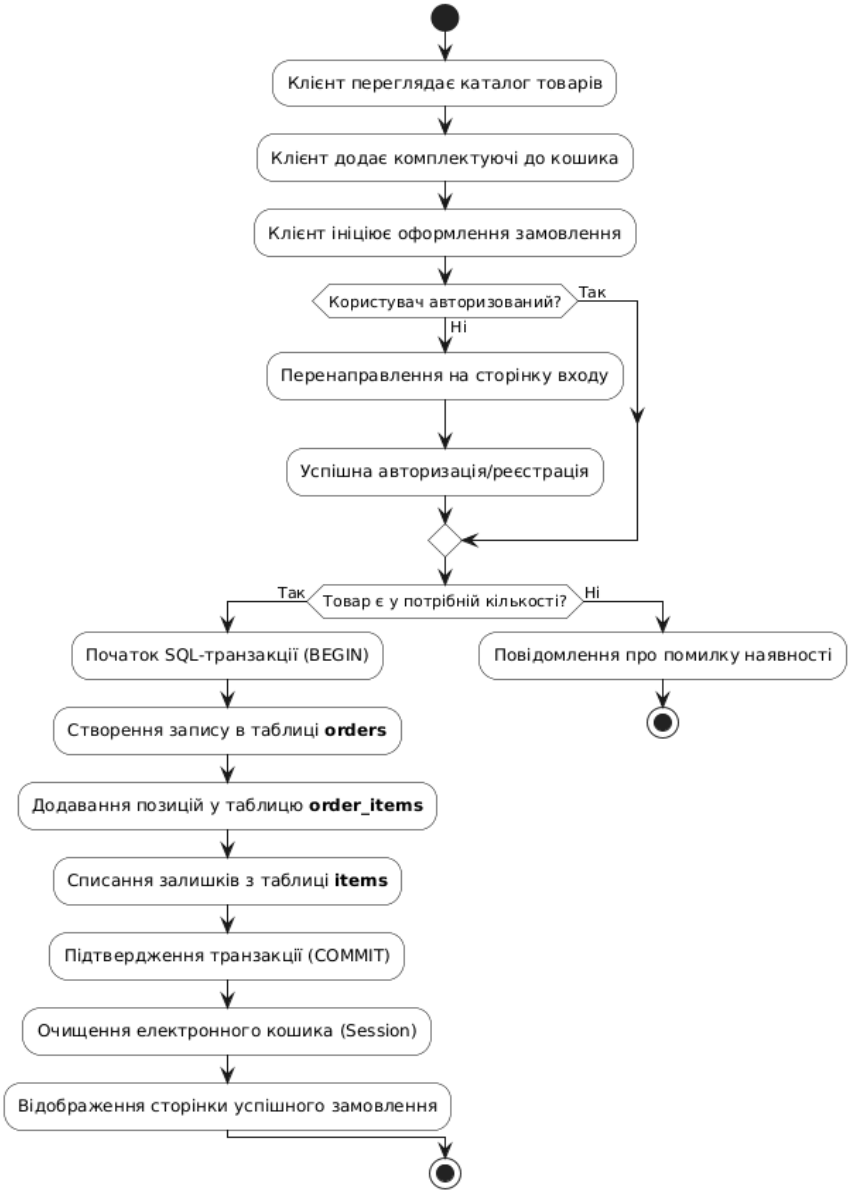


Рисунок 1.1 – Схема бізнес-процесу оформлення замовлення в системі

На рисунку 1.1 показано послідовність дій користувача під час оформлення замовлення. Спочатку клієнт переглядає каталог товарів і додає потрібні комплектуючі до кошика. Після початку оформлення система перевіряє, чи користувач авторизований. Якщо авторизація відсутня, користувач перенаправляється на сторінку входу або реєстрації.

Після успішної авторизації система перевіряє наявність товару в потрібній кількості. Якщо товар доступний, починається SQL-транзакція: створюється запис про замовлення, додаються позиції замовлення, оновлюються залишки товарів на складі, після чого транзакція підтверджується. Після успішного завершення процесу електронний кошик очищується, а користувачу відображається сторінка успішного оформлення замовлення. Якщо товару недостатньо, система повинна повідомити користувача про помилку наявності.

1.4 Обґрунтування вибору технологій розробки

Для реалізації вебсистеми електронної комерції було обрано технології, які дозволяють створити клієнт-серверний вебдодаток із підключенням до реляційної бази даних. Під час вибору технологічного стеку враховувалися зручність розробки, можливість реалізації маршрутизації, підтримка динамічного формування сторінок, робота з користувацькими сесіями, взаємодія з базою даних та можливість подальшого розширення системи.

Серверну частину вебдодатка реалізовано з використанням Node.js. Це середовище виконання JavaScript-коду на стороні сервера, яке дозволяє обробляти HTTP-запити, взаємодіяти з базою даних і формувати відповіді для клієнтської частини. Використання Node.js є доцільним для створення вебзастосунків, оскільки ця платформа має велику кількість готових модулів і добре підходить для розробки серверної логіки інтернет-магазину [1].

Для організації маршрутизації використано фреймворк Express. Він забезпечує створення маршрутів для основних сторінок системи: головної сторінки, кошика, сторінок реєстрації та входу, особистого кабінету, оформлення замовлення та адміністративної панелі. Express також спрощує обробку даних, які надходять із форм, і дозволяє розділити логіку системи за окремими маршрутами [2].

Серверна логіка системи написана мовою TypeScript. Використання TypeScript дозволяє застосовувати типізацію, що робить код більш зрозумілим і зменшує ймовірність помилок під час розробки. У межах проєкту TypeScript використовується для опису моделей системи, зокрема товарів, користувачів і замовлень, а також для реалізації логіки роботи з базою даних [3].

Для формування сторінок вебдодатка використано шаблонізатор EJS. Він дозволяє поєднувати HTML-розмітку з даними, які передаються із серверної частини. Завдяки цьому сторінки каталогу, кошика, профілю користувача та адміністративної панелі можуть динамічно відображати інформацію, отриману з бази даних [4].

Клієнтська частина системи побудована з використанням HTML5, CSS3 та Bootstrap 5. HTML5 використовується для побудови структури сторінок, CSS3 – для оформлення зовнішнього вигляду, а Bootstrap 5 – для створення адаптивного інтерфейсу. Це дозволяє зробити сторінки більш зручними для перегляду на різних пристроях і скоротити час розробки базових елементів інтерфейсу [5, 8, 9].

Для зберігання даних використано систему управління базами даних MySQL. У базі даних зберігається інформація про товари, користувачів, замовлення та позиції замовлень. Такий підхід дозволяє організувати централізоване зберігання інформації та забезпечити зв’язки між основними сутностями системи. Взаємодія серверної частини з базою даних виконується за допомогою SQL-запитів через модуль mysql2 [4, 6, 12].

Для збереження стану користувача використовується механізм сесій. Сесії дозволяють зберігати інформацію про кошик, стан авторизації

користувача та доступ до адміністративної частини. Це важливо для інтернет-магазину, оскільки користувач повинен мати можливість додавати товари до кошика та переходити між сторінками без втрати вибраних позицій [10].

Загальну структуру технологічного стеку вебсистеми електронної комерції наведено на рисунку 1.2.

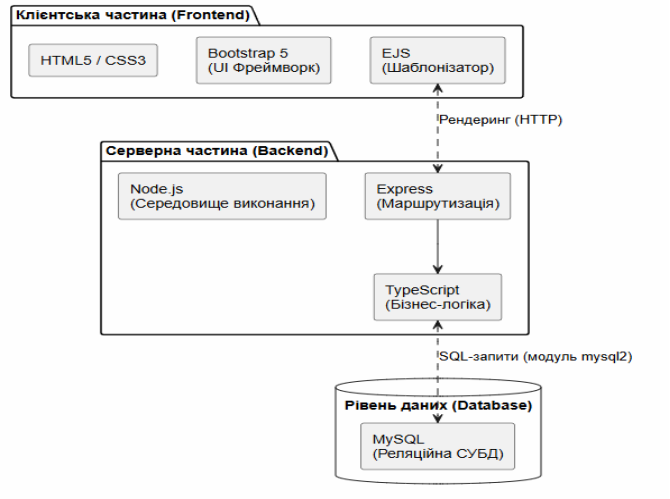


Рисунок 1.2 – Структура технологічного стеку вебсистеми електронної комерції

На рисунку 1.2 показано поділ системи на три основні рівні: клієнтську частину, серверну частину та рівень даних. Клієнтська частина відповідає за відображення сторінок і взаємодію користувача з інтерфейсом. Серверна частина обробляє запити, виконує бізнес-логіку, працює із сесіями та передає дані у шаблони сторінок. Рівень даних представлений базою MySQL, у якій зберігається основна інформація системи.

Обраний стек технологій відповідає поставленим завданням кваліфікаційної роботи. Він дозволяє реалізувати каталог товарів, електронний кошик, оформлення замовлень, авторизацію користувачів, особистий кабінет та адміністративну панель. Крім того, структура проєкту може бути розширена в майбутньому без повної зміни архітектури системи.

2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА БАЗИ ДАНИХ СИСТЕМИ

2.1 Проєктування функціональних можливостей системи

Проєктування вебсистеми електронної комерції передбачає визначення основних користувачів системи, їхніх можливостей, структури бази даних та загальної логіки взаємодії між клієнтською частиною, сервером і базою даних. На цьому етапі формується загальна модель майбутнього вебдодатка, яка надалі використовується під час програмної реалізації. Правильно виконане проєктування дозволяє заздалегідь визначити основні модулі системи, уникнути дублювання функцій і забезпечити логічний зв’язок між окремими частинами програмного продукту.

Розроблювана система PC PartsPro призначена для продажу комп’ютерних комплектуючих. Вона має забезпечувати роботу з каталогом товарів, електронним кошиком, оформленням замовлень, реєстрацією та авторизацією користувачів, переглядом профілю покупця і керуванням товарами через адміністративну частину. Така структура дає змогу охопити основні процеси роботи інтернет-магазину та забезпечити зручну взаємодію як для покупця, так і для адміністратора.

У системі передбачено три основні типи користувачів: неавторизований користувач, зареєстрований клієнт та адміністратор. Неавторизований користувач може переглядати каталог товарів, сортувати товари та додавати їх до кошика. Зареєстрований клієнт має розширені можливості: він може оформлювати замовлення, переглядати особистий профіль та історію покупок. Адміністратор відповідає за керування асортиментом товарів, зокрема додавання, редагування та видалення позицій у каталозі. Розмежування ролей дозволяє обмежити доступ до службових функцій системи та забезпечити безпечнішу роботу з даними. Розподіл функцій між ролями користувачів наведено в таблиці 2.1.

Таблиця 2.1 – Ролі користувачів та їхні функціональні можливості

Роль користувача	Основні можливості
Неавторизований користувач	Перегляд каталогу товарів, сортування товарів, додавання товарів до кошика
Зареєстрований клієнт	Оформлення замовлення, перегляд профілю, перегляд історії покупок
Адміністратор	Авторизація в прихованій панелі, додавання товарів, редагування даних товарів, видалення товарів

Діаграму варіантів використання вебсистеми PC PartsPro наведено на рисунку 2.1.

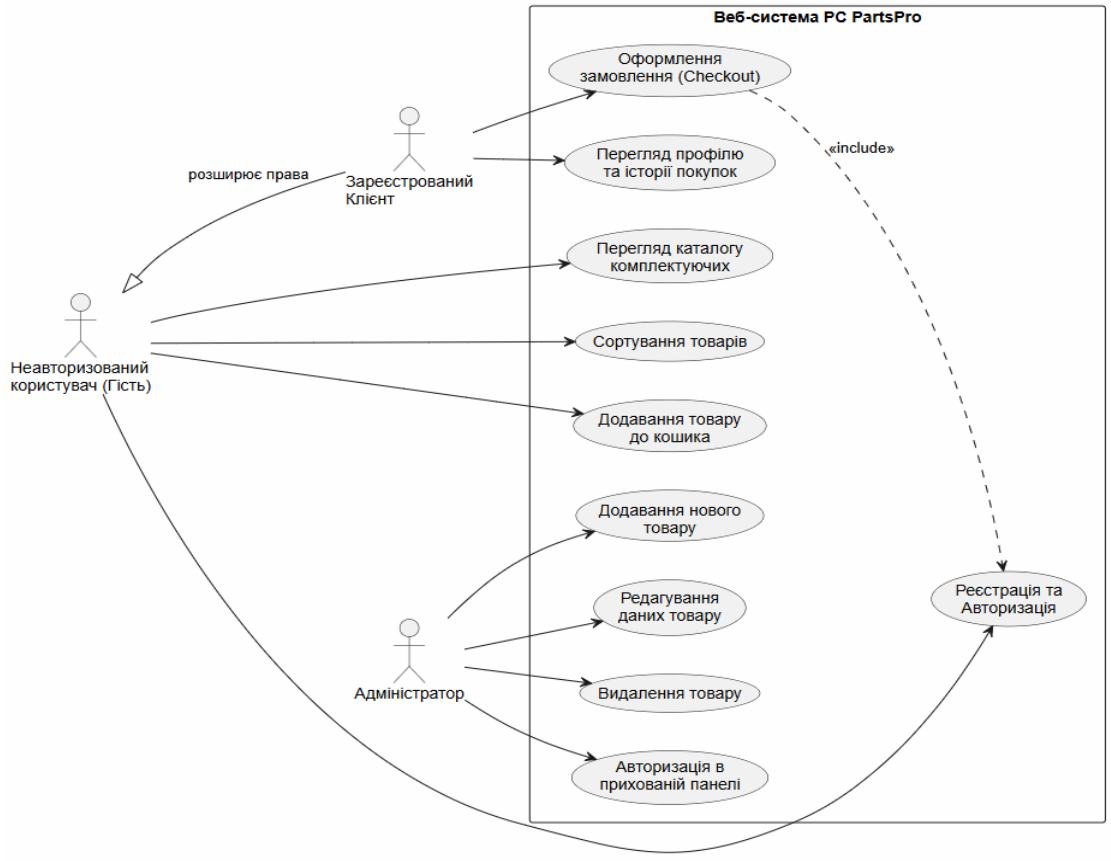


Рисунок 2.1 – Діаграма варіантів використання вебсистеми PC PartsPro

На рисунку 2.1 показано основні сценарії взаємодії користувачів із системою. Неавторизований користувач має доступ до базових дій, пов’язаних із переглядом каталогу, сортуванням товарів і додаванням позицій до кошика.

Після реєстрації або авторизації користувач отримує можливість оформлювати замовлення та переглядати профіль з історією покупок. Адміністратор має окремі функції для керування товарами, що дозволяє підтримувати актуальність каталогу.

2.2 Проєктування структури бази даних

База даних є однією з основних частин вебсистеми електронної комерції, оскільки саме в ній зберігається інформація про користувачів, товари, замовлення та склад замовлень. Для розроблюваної системи було обрано реляційну модель даних, оскільки вона дозволяє чітко визначити зв’язки між сутностями та забезпечити цілісність інформації.

У межах системи виділено чотири основні сутності: користувачі, товари, замовлення та позиції замовлення. Користувачі зберігаються в таблиці users. У цій таблиці містяться дані, необхідні для реєстрації та авторизації: ідентифікатор користувача, електронна пошта, пароль і ім’я. Товари зберігаються в таблиці items, яка містить назву товару, опис, ціну та кількість на складі.

Замовлення користувачів зберігаються в таблиці orders. Кожне замовлення пов’язане з конкретним користувачем через зовнішній ключ user_id. Також у таблиці зберігаються дата створення замовлення та його статус. Для зберігання складу замовлення використовується таблиця order_items. Вона пов’язує конкретне замовлення з товарами, які були додані до нього, і містить кількість товару та його ціну на момент оформлення.

Структура бази даних побудована так, щоб уникнути дублювання інформації та забезпечити зручне отримання даних під час роботи основних сторінок вебдодатка. Збереження ціни товару в таблиці order_items дозволяє фіксувати вартість позиції на момент оформлення замовлення, навіть якщо надалі ціна товару в каталозі буде змінена.

ER-модель бази даних вебсистеми наведено на рисунку 2.2.

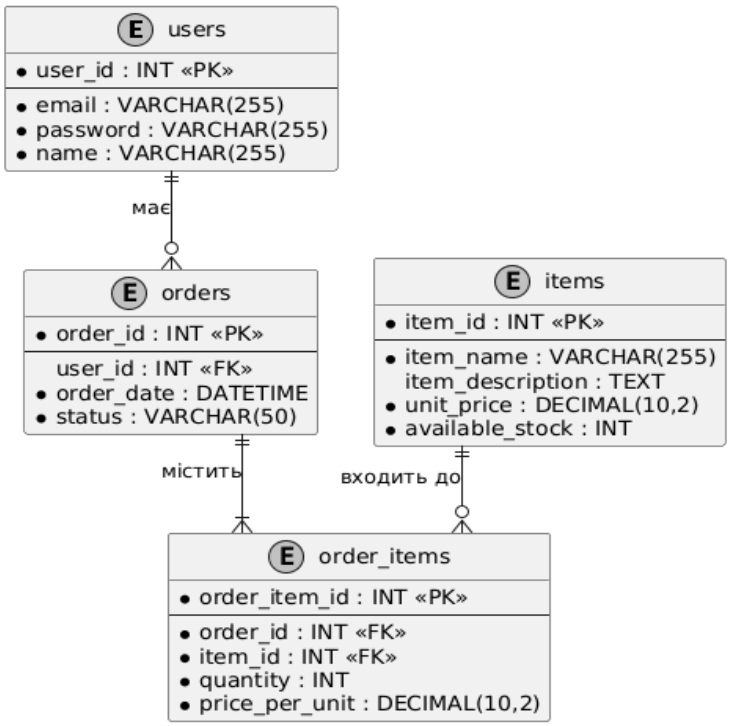


Рисунок 2.2 – ER-модель бази даних вебсистеми PC PartsPro

На рисунку 2.2 показано основні таблиці бази даних та зв’язки між ними. Один користувач може мати багато замовлень, тому між таблицями users і orders існує зв’язок один-до-багатьох. Одне замовлення може містити декілька товарів, тому таблиця orders пов’язана з таблицею order_items. Кожна позиція замовлення відповідає одному товару з таблиці items, але один товар може входити до багатьох замовлень. Така структура дозволяє зберігати історію покупок і коректно відображати склад кожного замовлення.

2.3 Проєктування архітектури вебдодатка

Під час розробки системи використовується клієнт-серверна архітектура. Клієнтська частина відповідає за відображення сторінок

користувачу, серверна частина обробляє запити та виконує бізнес-логіку, а база даних забезпечує зберігання інформації.

Клієнтська частина представлена EJS-шаблонами, які формують HTML-сторінки для користувача. Через ці сторінки користувач переглядає каталог товарів, працює з кошиком, проходить авторизацію, переглядає профіль або взаємодіє з адміністративною панеллю. Серверна частина побудована на Node.js та Express. Вона приймає HTTP-запити, визначає потрібний маршрут, викликає відповідні методи моделей і передає дані у шаблони для відображення.

Окремий рівень системи становлять TypeScript-класи моделей. Вони відповідають за роботу з основними сутностями: товарами, користувачами та замовленнями. Такий поділ дозволяє не змішувати маршрутизацію, бізнес-логіку та SQL-запити в одному місці. Для зберігання інформації використовується база даних MySQL, з якою серверна частина взаємодіє через SQL-запити.

Архітектура системи має логічний поділ на такі складові: представлення, контролери маршрутів, моделі та базу даних. Представлення відповідають за інтерфейс, маршрути Express – за приймання й обробку запитів, моделі – за виконання операцій над даними, а база даних – за зберігання інформації. Такий підхід спрощує підтримку проєкту та дозволяє розширювати функціонал без повної перебудови системи [11].

2.4 Проєктування взаємодії компонентів системи

Для кращого розуміння роботи вебдодатка було побудовано діаграму послідовності. Вона показує, як відбувається обмін даними між користувачем, EJS-шаблонами, серверними маршрутами, TypeScript-моделями та базою даних MySQL. Діаграма також дозволяє простежити, на якому етапі

виконується обробка запиту, перевірка сесії, звернення до бази даних і формування відповіді для користувача.

Діаграму послідовності взаємодії компонентів наведено на рисунку 2.3.

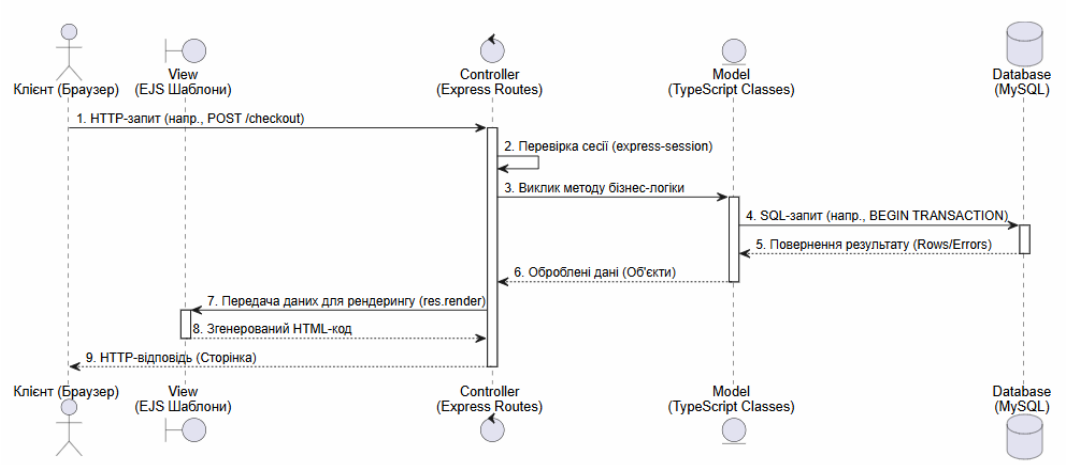


Рисунок 2.3 – Діаграма послідовності взаємодії компонентів вебсистеми

На рисунку 2.3 показано загальний порядок обробки запиту в системі. Користувач через браузер надсилає HTTP-запит до вебдодатка. Запит потрапляє до маршруту Express, який виконує перевірку сесії, визначає потрібну дію та звертається до відповідної моделі. Модель формує SQL-запит до бази даних MySQL, отримує результат і повертає його назад до маршруту.

Після отримання даних серверна частина передає їх у EJS-шаблон. Шаблон формує HTML-сторінку, яка повертається користувачу як HTTP-відповідь. Така послідовність використовується під час перегляду каталогу, роботи з кошиком, авторизації, перегляду профілю та взаємодії з адміністративною панеллю.

Під час оформлення замовлення взаємодія між компонентами має особливе значення, оскільки потрібно не лише створити запис про замовлення, а й додати товарні позиції та оновити залишки на складі. Для цього використовується транзакційний підхід. Він дозволяє виконати всі пов’язані операції як єдиний процес і запобігти частковому збереженню некоректних даних у разі помилки.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБСИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

3.1 Налаштування серверної частини та маршрутизації

Програмна реалізація вебсистеми електронної комерції PC PartsPro виконувалася з використанням платформи Node.js, мови TypeScript, фреймворку Express, шаблонізатора EJS та системи управління базами даних MySQL. На початковому етапі було створено структуру проєкту, налаштовано серверну частину, підключено необхідні модулі та визначено основні маршрути вебдодатка.

Основним файлом серверної частини є index.ts. У ньому підключаються бібліотеки Express, express-session, dotenv, path, а також імпортуються класи моделей, які відповідають за роботу з товарами, замовленнями та користувачами. Для зручного збереження конфігураційних параметрів використовується файл .env, з якого завантажуються змінні середовища.

Фрагмент налаштування серверної частини наведено в лістингу:

```
import express, { Request, Response, NextFunction } from ‘express’;
import session from ‘express-session’;
import dotenv from ‘dotenv’;
import path from ‘path’;
import { Product } from ‘./models/Product’;
import { Order } from ‘./models/Order’;
import { User } from ‘./models/User’;

dotenv.config();

const app = express();
const port = process.env.PORT || 3000;

app.set(‘view engine’, ‘ejs’);
```

```
app.set('views', path.join(__dirname, 'views'));
app.use(express.json());
app.use(express.urlencoded({ extended: true }));

app.use(session({
  secret: process.env.SESSION_SECRET || 'super_secret_diploma_key',
  resave: false,
  saveUninitialized: true,
  cookie: { secure: false }
}));

const productModel = new Product();
const orderModel = new Order();
const userModel = new User();
```

У наведеному фрагменті виконується підключення основних модулів, налаштовується шаблонізатор EJS, задається папка з представленнями та підключається middleware для обробки JSON-даних і даних із HTML-форм. Окремо налаштовано механізм сесій, який використовується для збереження кошика, даних авторизованого користувача та стану доступу до адміністративної частини.

Для роботи з базовими сутностями створено окремі моделі: Product, Order та User. Такий підхід дозволяє відокремити маршрутизацію від бізнес-логіки та зробити структуру проєкту більш зрозумілою. Маршрути Express відповідають за приймання HTTP-запитів, а моделі виконують операції з базою даних. Повний лістинг основних файлів програмного коду вебсистеми наведено в додатку А.

Основні маршрути вебдодатка наведено в таблиці 3.1.

Таблиця 3.1 – Основні маршрути вебсистеми PC PartsPro

Маршрут	Призначення
/	Відображення головної сторінки та каталогу
/cart	Перегляд електронного кошика
/checkout	Оформлення замовлення
/register	Реєстрація нового користувача
/login	Авторизація користувача
/profile	Перегляд особистого кабінету та історії замовлень
/logout	Вихід із системи
/system-management	Прихована адміністративна панель
/system-management/add	Додавання нового товару
/system-management/edit/:id	Редагування товару

3.2 Реалізація роботи з каталогом товарів

Однією з основних частин вебсистеми є каталог товарів. Він призначений для відображення комп’ютерних комплектуючих, які доступні користувачу для перегляду та додавання до кошика. На головній сторінці системи користувач бачить назву проєкту, верхню панель навігації, кнопку входу, кнопку переходу до кошика, головний банер та блок із товарами. Кожен товар подано у вигляді окремої картки з назвою, коротким описом, ціною та статусом наявності. Це дозволяє користувачу швидко переглядати асортимент і обирати потрібні комплектуючі без зайвих переходів між сторінками.

Головна сторінка вебсистеми наведена на рисунку 3.1.

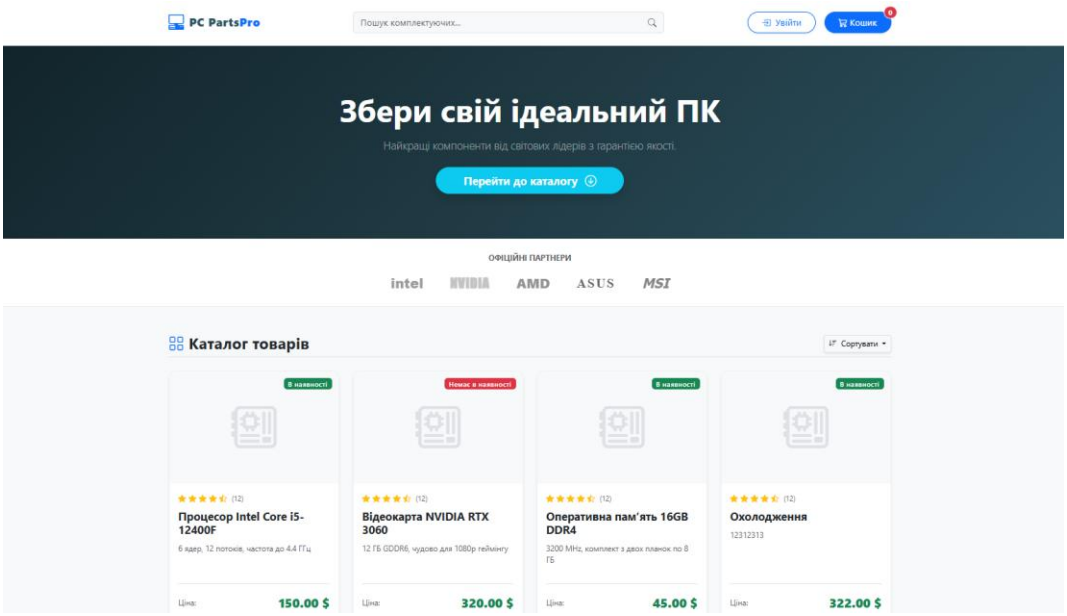


Рисунок 3.1 – Головна сторінка вебсистеми PC PartsPro

Каталог товарів складається з карток, у яких відображаються назва товару, короткий опис, ціна, поле для вибору кількості, кнопка додавання до кошика та позначка наявності. Такий формат подання є зручним для користувача, оскільки вся основна інформація про товар розміщується в одному блоці. Сторінку каталогу товарів наведено на рисунку 3.2.

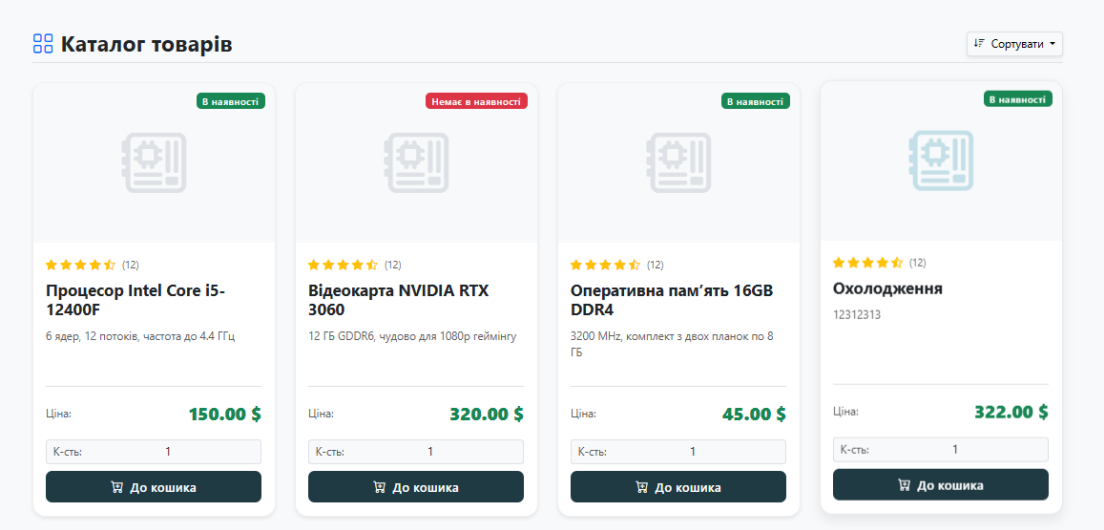


Рисунок 3.2 – Відображення товарів у каталозі вебсистеми

Для отримання товарів із бази даних використовується модель Product. Вона містить методи для отримання списку товарів, пошуку товару за

ідентифікатором, створення, редагування та видалення записів. Під час відображення головної сторінки сервер звертається до бази даних, отримує список товарів і передає його у EJS-шаблон для формування HTML-сторінки.

У системі також реалізовано сортування товарів. Користувач може змінити порядок відображення позицій, а серверна частина формує SQL-запит з відповідним параметром сортування. Наприклад, товари можуть бути відсортовані за зростанням або спаданням ціни, а також за новизною додавання до каталогу.

Для товарів у каталозі відображається стан наявності. Якщо кількість товару на складі більша за нуль, користувач бачить позначку «В наявності». Якщо товар відсутній, система відображає відповідний статус. Це дозволяє покупцю швидше оцінити, які позиції доступні для замовлення.

3.3 Реалізація кошика та оформлення замовлення

Електронний кошик є важливою частиною інтернет-магазину, оскільки він забезпечує тимчасове збереження товарів, які користувач планує придбати. У розробленій системі кошик реалізовано за допомогою сесій. Це дозволяє зберігати вибрані товари під час переходу між сторінками вебдодатка. Завдяки цьому користувач може продовжувати перегляд каталогу, не втрачаючи вже додані позиції.

Після додавання товару до кошика користувач може перейти на сторінку перегляду кошика. На цій сторінці відображаються назва товару, ціна, кількість, сума за позицією та загальна вартість замовлення. Також передбачено кнопку очищення кошика та кнопку оформлення замовлення. Така структура сторінки дає змогу користувачу перевірити склад замовлення перед його підтвердженням.

Сторінку кошика наведено на рисунку 3.3.

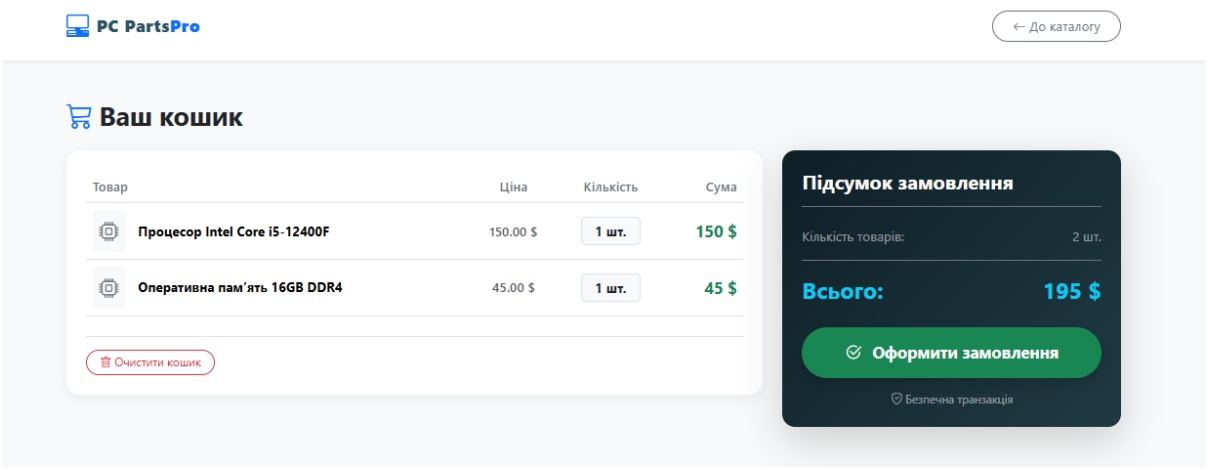


Рисунок 3.3 – Сторінка електронного кошика вебсистеми

Фрагмент шаблону cart.ejs, який відповідає за відображення товарів у кошику, наведено в лістингу:

```
<table class=»table table-hover align-middle»>
  <thead class=»table-light»>
    <tr>
      <th>Товар</th>
      <th>Ціна</th>
      <th style=»width: 150px;»>Кількість</th>
      <th>Сума</th>
    </tr>
  </thead>
  <tbody>
    <% cartItems.forEach(item => { %>
      <tr>
        <td>
          <h6 class=»mb-0»><%= item.product.item_name %></h6>
        </td>
        <td>$<%= item.product.unit_price.toFixed(2) %></td>
        <td>
          <input type=»number» name=»quantity» value=»<%= item.quantity
%>»

                 class=»form-control form-control-sm» readonly>
        </td>
```

```
            <td class=»fw-bold»>$<%= item.itemTotal.toFixed(2) %></td>
          </tr>
        <% }) %>
    </tbody>
</table>

<div class=»text-end mt-4»>
    <h4>Разом: <span class=»text-primary»>$<%= totalPrice.toFixed(2)
%></span></h4>
</div>
```

У цьому фрагменті використовується цикл forEach, який проходить по всіх товарах, доданих до кошика. Для кожної позиції виводиться назва товару, його ціна, кількість і загальна сума. Значення totalPrice використовується для відображення підсумкової вартості замовлення.

Оформлення замовлення реалізовано через модель Order. Під час підтвердження покупки система створює запис у таблиці orders, додає позиції замовлення до таблиці order_items і списує відповідну кількість товарів із таблиці items. Для забезпечення цілісності даних використовується SQL-транзакція. Діаграму процесу оформлення замовлення наведено на рисунку 3.4.

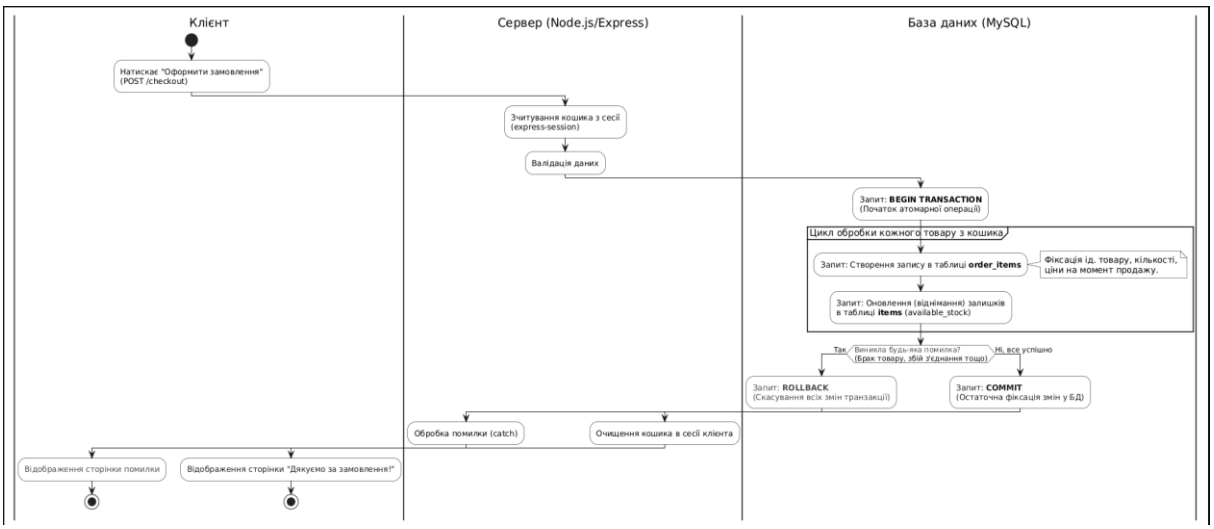


Рисунок 3.4 – Процес транзакційного оформлення замовлення

Транзакційна логіка оформлення замовлення наведена в лістингу:

```
async checkout(cartItems: ICartItem[], userId: number | null = null) {
   const connection = await this.db.getConnection();
   try {
      await connection.beginTransaction();
      const [orderResult]: any = await connection.query(
         'INSERT INTO orders (user_id) VALUES (?)',
         [userId]
      );
      const orderId = orderResult.insertId;
      for (const item of cartItems) {
         await connection.query(
            'INSERT INTO order_items (order_id, item_id, quantity, price_per_unit) VALUES (?, ?, ?, ?)',
            [orderId, item.item_id, item.quantity, item.price_per_unit]
         );
         await connection.query(
            'UPDATE items SET available_stock = available_stock - ? WHERE item_id = ?',
            [item.quantity, item.item_id]
         );
      }
      await connection.commit();

      return orderId;
   } catch (error) {
      await connection.rollback();
      throw error;

   } finally {
      connection.release();
   }
}
```

У цьому фрагменті спочатку відкривається окреме з'єднання з базою даних, після чого запускається транзакція. Далі створюється головний запис

замовлення, а всі товари з кошика послідовно додаються до таблиці order_items. Після цього в таблиці items оновлюється кількість товару на складі.

Якщо всі запити виконуються успішно, транзакція підтверджується за допомогою commit. Якщо під час оформлення виникає помилка, виконується rollback, і зміни не зберігаються частково. Після завершення операції з'єднання з базою даних звільняється.

Після успішного оформлення замовлення користувачу відображається сторінка підтвердження. На ній повідомляється, що замовлення зареєстровано в системі, а транзакцію успішно підтверджено. Сторінку успішного оформлення замовлення наведено на рисунку 3.5.

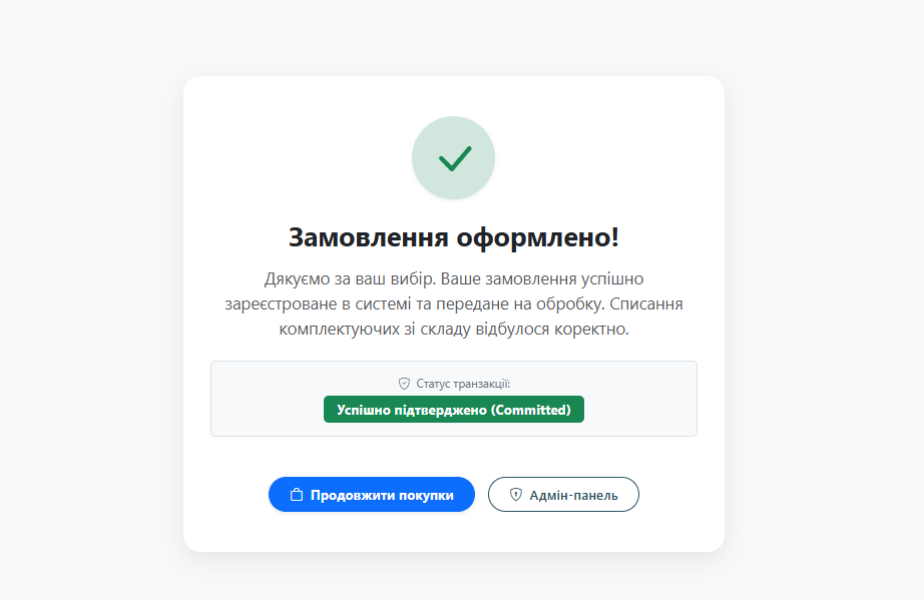


Рисунок 3.5 – Сторінка успішного оформлення замовлення

3.4 Реалізація реєстрації, авторизації та особистого кабінету

Для персоналізації роботи з вебсистемою реалізовано механізм реєстрації та авторизації користувачів. Реєстрація дозволяє створити новий

обліковий запис, а авторизація – отримати доступ до особистого кабінету та історії замовлень.

Після входу в систему ідентифікатор користувача та його ім’я зберігаються в сесії. Це дозволяє визначати поточного користувача під час переходу між сторінками та отримувати з бази даних саме його замовлення. Для обмеження доступу до сторінки профілю використовується перевірка авторизації. Якщо користувач не увійшов до системи, він перенаправляється на сторінку входу.

Особистий кабінет містить інформацію про користувача та таблицю з історією замовлень. У ній відображаються номер замовлення, дата, сума та статус. Це дозволяє користувачу переглядати інформацію про попередні покупки. Сторінку особистого кабінету користувача наведено на рисунку 3.6.

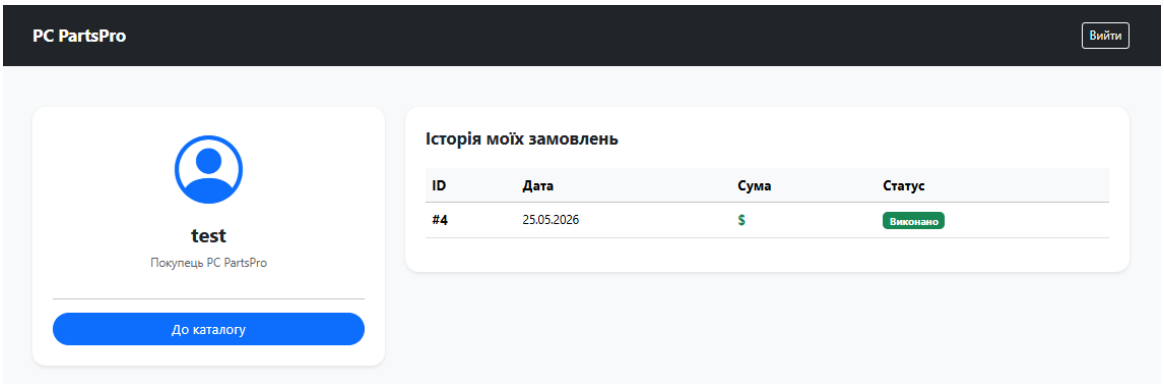


Рисунок 3.6 – Особистий кабінет користувача з історією замовлень

У моделі User реалізовано логіку створення користувача та входу в систему. Під час реєстрації перевіряється, чи не використовується вже введена електронна пошта. Пароль користувача не зберігається у відкритому вигляді, а хешується перед записом у базу даних. Під час входу введений пароль порівнюється з хешем, який зберігається в таблиці users.

Такий підхід підвищує безпеку системи та дозволяє відокремити персональні дані користувача від загальної логіки роботи інтернет-магазину.

3.5 Реалізація адміністративної панелі

Для керування товарами в системі реалізовано адміністративну панель. Вона призначена для перегляду списку товарів, додавання нових позицій, редагування наявних товарів і видалення записів із каталогу. Доступ до адміністративної частини обмежений, тому звичайний користувач не може змінювати інформацію про товари.

Адміністративна панель має окремий маршрут і відкривається через прихований шлях /system-management. Перед доступом до панелі перевіряється, чи має користувач права адміністратора. Якщо доступ не підтверджено, система перенаправляє користувача на сторінку входу до адміністративної частини. Сторінку адміністративної панелі наведено на рисунку 3.7.

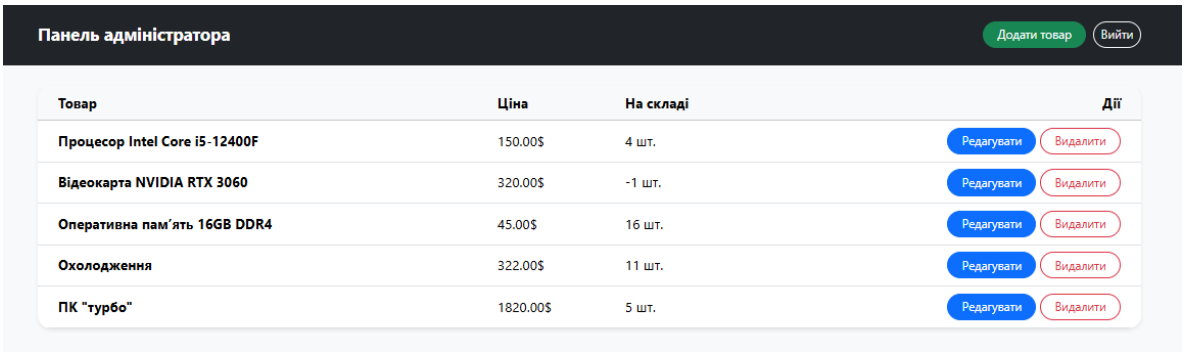


Рисунок 3.7 – Адміністративна панель керування товарами

Керування товарами реалізовано через модель Product. Вона містить методи для додавання нового товару, оновлення інформації про товар і видалення товару з бази даних. Фрагмент програмної логіки керування товарами наведено в лістингу:

```
async create(product: Iproduct) {
   return await this.db.query(
      'INSERT INTO items (item_name, item_description, unit_price, available_stock) VALUES (?, ?, ?, ?)',
      [
         product.item_name,
         product.item_description,
         product.unit_price,
         product.available_stock
      ]
   );
}
async update(id: number, product: Iproduct) {
   return await this.db.query(
      'UPDATE items SET item_name = ?, item_description = ?, unit_price = ?, available_stock = ? WHERE item_id = ?',
      [
         product.item_name,
         product.item_description,
         product.unit_price,
         product.available_stock,
         id
      ]
   );
}
async delete(id: number) {
   return await this.db.query(
      'DELETE FROM items WHERE item_id = ?',
      [id]
   );
}
```

У наведеному фрагменті метод create додає новий товар до таблиці items. Метод update оновлює назву, опис, ціну та кількість товару на складі. Метод delete видаляє товар за його ідентифікатором. Використання параметризованих SQL-запитів дозволяє передавати значення окремо від тексту запиту та робить роботу з базою даних більш безпечною.

Адміністративна панель спрощує керування асортиментом магазину, оскільки всі основні операції з товарами виконуються через вебінтерфейс без прямого редагування бази даних.

3.6 Тестування роботи системи

Після завершення програмної реалізації основних модулів було проведено локальне тестування вебсистеми через браузер. Перевірялася коректність роботи каталогу товарів, сортування, електронного кошика, оформлення замовлень, авторизації користувача, особистого кабінету та адміністративної панелі.

Під час тестування було перевірено, чи правильно товари відображаються на головній сторінці, чи додаються вони до кошика, чи підраховується загальна вартість замовлення та чи створюються відповідні записи в базі даних після оформлення покупки. Також перевірялося оновлення залишків товарів на складі.

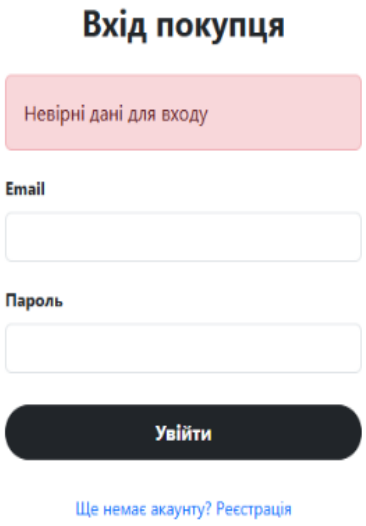


Рисунок 3.8 – Спрацювання системи валідації під час некоректного введення даних авторизації

Окремо протестовано авторизацію користувача. У разі введення правильних даних користувач отримував доступ до особистого кабінету, а при некоректному email або паролі система відображала повідомлення про помилку й не відкривала профіль. Приклад спрацювання системи валідації наведено на рисунку 3.8.

Також було протестовано процес оформлення замовлення: після додавання товару до кошика система коректно відображала його назву, ціну, кількість і загальну суму, а після підтвердження створювала запис про покупку та очищувала кошик. Окремо перевірено адміністративну частину: додавання, редагування й видалення товарів виконувалися коректно, а зміни відображалися в каталозі вебдодатка. Результати перевірки основних функцій системи наведено в таблиці 3.2.

Таблиця 3.2 – Результати тестування основних функцій вебсистеми

Функція	Очікуваний результат	Результат перевірки
Відображення каталогу	Товари виводяться на головній сторінці	Виконано
Сортування товарів	Порядок товарів змінюється відповідно до вибраного критерію	Виконано
Додавання до кошика	Товар додається до сесійного кошика	Виконано
Перегляд кошика	Відображаються товари, кількість і загальна сума	Виконано
Оформлення замовлення	Створюється запис у таблицях orders та order_items	Виконано
Списання залишків	Кількість товарів у таблиці items зменшується після замовлення	Виконано
Авторизація користувача	Користувач отримує доступ до профілю після входу	Виконано
Авторизація з некоректними даними	Система не відкриває профіль і показує повідомлення про помилку	Виконано
Перегляд профілю	Відображається інформація про користувача та історія замовлень	Виконано
Адміністративна панель	Адміністратор може переглядати список товарів	Виконано

Проведене тестування показало, що основні функції вебсистеми працюють відповідно до поставлених вимог. Користувач може переглядати каталог, додавати товари до кошика, оформлювати замовлення та переглядати історію покупок. Адміністратор має можливість керувати товарами через окрему панель. Система може бути використана як основа для подальшого розвитку інтернет-магазину комп’ютерних комплектуючих.

ВИСНОВКИ

У даній роботі було виконано проєктування та програмну реалізацію вебсистеми електронної комерції для продажу комп’ютерних компонентів. Розроблена система призначена для автоматизації основних процесів інтернет-магазину: перегляду каталогу товарів, додавання комплектуючих до кошика, оформлення замовлень, реєстрації та авторизації користувачів, перегляду особистого кабінету й керування товарами через адмін-панель.

У першому розділі було проаналізовано предметну область вебсистем електронної комерції та особливості продажу комп’ютерних комплектуючих. Було розглянуто потребу в актуальному відображенні асортименту, контролі залишків товарів і зручному оформленні замовлень. Також було проаналізовано існуючі аналоги інформаційних систем, серед яких OpenCart, WooCommerce, Shopify і Magento. На основі цього визначено основні вимоги до розроблюваної системи та обґрунтовано вибір технологій: Node.js, TypeScript, Express, EJS, HTML5, CSS3, Bootstrap 5 та MySQL.

У другому розділі було виконано проєктування архітектури та бази даних вебсистеми. Визначено основні ролі користувачів: неавторизований користувач, зареєстрований клієнт та адміністратор. Для зберігання даних було виділено основні сутності системи: користувачі, товари, замовлення та позиції замовлення. Їх реалізовано за допомогою таблиць users, items, orders та order_items. Також було описано загальну архітектуру вебдодатка, яка передбачає поділ на клієнтську частину, серверну логіку, моделі та базу даних.

У третьому розділі було описано програмну реалізацію вебсистеми. Було налаштовано серверну частину проєкту, підключено шаблонізатор EJS, механізм сесій та створено моделі для роботи з товарами, користувачами й замовленнями. У межах реалізації створено каталог товарів із можливістю сортування, електронний кошик, сторінку оформлення замовлення, механізм реєстрації та авторизації, особистий кабінет користувача й адмін-панель.

Оформлення замовлення реалізовано з використанням SQL-транзакцій, що дозволяє створювати запис про покупку, додавати товарні позиції та оновлювати складські залишки як єдиний процес. У разі виникнення помилки зміни не зберігаються частково, що підвищує надійність роботи системи. Адміністративна частина надає можливість додавати, редагувати та видаляти товари без прямого редагування бази даних.

Після завершення реалізації було проведено тестування основних функцій системи. Перевірено роботу каталогу, сортування, кошика, оформлення замовлення, авторизації користувача, особистого кабінету та адміністративної панелі. Також було перевірено спрацювання системи валідації під час введення некоректних даних авторизації. Результати тестування показали, що основні функції вебсистеми працюють відповідно до поставлених вимог.

Розроблена вебсистема має практичне значення, оскільки може бути використана як основа для інтернет-магазину комп’ютерних комплектуючих. Вона забезпечує зручну взаємодію користувача з каталогом, автоматизує оформлення замовлень, зберігає дані в базі та надає адміністратору інструменти для керування товарами. Мету кваліфікаційної роботи досягнуто: було спроєктовано та реалізовано вебсистему електронної комерції, яка виконує основні функції інтернет-магазину та може бути розширена в майбутньому.

.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація середовища виконання Node.js. URL: https://nodejs.org/docs/ (дата звернення: 10.04.2026).

2. Офіційна документація вебфреймворку Express. URL: https://expressjs.com/ (дата звернення: 15.04.2026).

3. Документація мови програмування TypeScript. URL: https://www.typescriptlang.org/docs/ (дата звернення: 12.04.2026).

4. Модуль mysql2. MySQL client for Node.js. URL: https://www.npmjs.com/package/mysql2 (дата звернення: 22.04.2026).

5. Офіційна документація СУБД MySQL. URL: https://dev.mysql.com/doc/ (дата звернення: 18.04.2026).

6. CSS-фреймворк Bootstrap 5. Документація та приклади. URL: https://getbootstrap.com/docs/5.3/ (дата звернення: 20.04.2026).

7. EJS. Шаблонізатор для вбудовування JavaScript. URL: https://ejs.co/ (дата звернення: 25.04.2026).

8. MDN Web Docs. Вебтехнології для розробників. URL: https://developer.mozilla.org/ (дата звернення: 28.04.2026).

9. Специфікація стандарту HTML5. URL: https://html.spec.whatwg.org/ (дата звернення: 01.05.2026).

10. Модуль express-session. Управління сесіями у Express. URL: https://www.npmjs.com/package/express-session (дата звернення: 03.05.2026).

11. Патерн проєктування MVC (Model-View-Controller). Вікіпедія : вільна енцикл. URL: https://uk.wikipedia.org/wiki/Model-View-Controller (дата звернення: 05.05.2026).

12. Реляційні бази даних та мова SQL. Навчальні матеріали W3Schools. URL: https://www.w3schools.com/sql/ (дата звернення: 06.05.2026).

ДОДАТОК А

Лістинг коду головного модуля (index.ts)

```
import express, { Request, Response, NextFunction } from 'express';
import session from 'express-session';
import dotenv from 'dotenv';
import path from 'path';

// Імпорт моделей бізнес-логіки
import { Product } from './models/Product';
import { Order } from './models/Order';
import { User } from './models/User';

dotenv.config();
const app = express();
const port = process.env.PORT || 3000;

// Налаштування шаблонізатора EJS
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

// Middleware для обробки даних
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use(express.static(path.join(__dirname, 'public')));

// Налаштування сесій
app.use(session({
    secret: process.env.SESSION_SECRET || 'pc_parts_pro_secure_key_2026',
    resave: false,
    saveUninitialized: true,
    cookie: { maxAge: 3600000 } // 1 година
```

```
}));

// Ініціалізація моделей
const productModel = new Product();
const orderModel = new Order();
const userModel = new User();

// Middleware для захисту маршрутів
const requireAuth = (req: Request, res: Response, next: NextFunction) => {
    if (req.session && req.session.userId) return next();
    res.redirect('/login');
};

const requireAdmin = (req: Request, res: Response, next: NextFunction) => {
    if (req.session && req.session.isAdmin) return next();
    res.status(403).render('error', { message: 'Доступ заборонено' });
};

// --- МАРШРУТИ (ROUTES) ---

// Головна сторінка - Каталог
app.get('/', async (req: Request, res: Response) => {
    const sort = req.query.sort as string || 'item_id';
    const products = await productModel.getAll(sort);
    res.render('index', { products, user: req.session.user });
});

// Реєстрація з математичною капчею
app.get('/register', (req: Request, res: Response) => {
    const n1 = Math.floor(Math.random() * 10) + 1;
    const n2 = Math.floor(Math.random() * 10) + 1;
    req.session.captcha = n1 + n2;
    res.render('register', { q: `Скільки буде ${n1} + ${n2}?`, error: null });
});
```

```
app.post('/register', async (req: Request, res: Response) => {
    const { email, password, name, captcha } = req.body;
    if (parseInt(captcha) !== req.session.captcha) {
        return res.render('register', { q: 'Повторіть спробу', error: 'Невірна капча' });
    }
    const userId = await userModel.create(email, password, name);
    res.redirect('/login');
});

// Кошик (на базі сесій)
app.post('/add-to-cart', (req: Request, res: Response) => {
    const { itemId, price, name } = req.body;
    if (!req.session.cart) req.session.cart = [];
    req.session.cart.push({ itemId, price, name });
    res.redirect('/');
});

// Оформлення замовлення (Транзакція)
app.post('/checkout', requireAuth, async (req: Request, res: Response) => {
    if (!req.session.cart || req.session.cart.length === 0) return res.redirect('/');

    try {
        const orderId = await orderModel.createTransaction(
            req.session.userId,
            req.session.cart
        );
        req.session.cart = []; // Очищення кошика після успіху
        res.render('success', { orderId });
    } catch (error) {
        res.render('error', { message: 'Помилка транзакції бази даних' });
    }
});

// Прихована панель адміністратора (CRUD)
app.get('/system-management', requireAdmin, async (req: Request, res: Response) => {
```

```
    const items = await productModel.getAll();
    const orders = await orderModel.getAllWithDetails();
    res.render('admin-panel', { items, orders });
});

app.listen(port, () => {
    console.log(`Server running at http://localhost:${port}`);
});
```

Декларація

академічної доброчесності

здобувача освіти ВСП «Економіко-правничий фаховий коледж ЗНУ»

Я, Халач Дмитро Олександрович, здобувач освіти 4 курсу, спеціальності/освітньо-професійної програми 121 Інженерія програмного забезпечення/Розробка програмного забезпечення, групи К121-22, адреса електронної пошти dima0666948950@gmail.com

- підтверджую, що написана мною кваліфікаційна робота на тему «Архітектура та розробка веб-системи електронної комерції для комп'ютерних компонентів» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згоден на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 28.05.2026 Підпис _____ Дмитро ХАЛАЧ

Дата 28.05.2026 Підпис _____ Дмитро ДРАЄВСЬКИЙ