Національний університет «Одеська політехніка»

Інститут комп'ютерних систем

Кафедра інформаційних систем

**КУРСОВА РОБОТА**

з дисципліни «Об'єктно-орієнтоване програмування»

Тема: «Розробка онлайн магазину к»

Студента \_\_2\_\_ курсу АІ-224 групи

Спеціальності 122 – «Комп’ютерні науки»

\_\_\_\_\_\_\_\_\_Ревенко А.О.

(прізвище та ініціали)

Керівник ст.викл, к.т.н. Годовіченко М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

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

Кількість балів: \_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

Одеса – 2024

ЗМІСТ

[ЗАВДАННЯ НА КУРСОВУ РОБОТУ 3](#_Toc168737697)

[ВСТУП 6](#_Toc168737698)

[1 ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ВИКОРАСТАНІ ТЕХНОЛОГІЇ 7](#_Toc168737699)

[1.1 Теоретичні відомості 7](#_Toc168737700)

[2 ПРОГРАМНА РЕАЛІЗАЦІЯ 10](#_Toc168737701)

[3 ІНСТРУКЦІЯ КОРИСТУВАЧА 11](#_Toc168737702)

[3.1 Інструкція користувача 11](#_Toc168737703)

[ВИСНОВКИ 17](#_Toc168737704)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 18](#_Toc168737705)

[ДОДАТОК А 19](#_Toc168737706)

Національний університет «Одеська політехніка»

Інститут комп'ютерних систем

Кафедра інформаційних систем

ЗАВДАННЯ НА КУРСОВУ РОБОТУ

1. Розробка інтерфейсу для додавання товару в кошик та оформлення замовлення.

2. Реалізація перегляду товарів з можливістю відображення детальної інформації про кожен товар.

3. Реалізація функції перегляду товарів за категоріями.

4. Додавання нових товарів користувачами.

5. Зберігання товарів, замовлень, користувачів у базі даних.

6. Реалізація можливості додавання товарів до обраного для подальшого перегляду.

7. Реалізація пагінації для навігації між списками товарів.

8. Взаємодія з базою даних для ефективного зберігання та обробки інформації.

АНОТАЦІЯ

У цій курсовій роботі розроблено веб-додаток "CompShop", призначений для онлайн-продажу комп'ютерних комплектуючих. Архітектура додатку побудована за принципом Model-View-Controller (MVC), що забезпечує зручне та ефективне управління логікою програми та її представленням.

Backend реалізований за допомогою фреймворку Spring, що забезпечує стабільну та надійну обробку запитів користувачів, а також взаємодію з базою даних. Використання Spring дозволяє легко впроваджувати різні функціональні можливості та забезпечує високу масштабованість додатку.

Frontend розроблений на основі React, що забезпечує швидку та динамічну взаємодію з користувачем. Завдяки компонентній архітектурі React, додаток є модульним та легко підтримується.

Для зберігання даних використовується база даних PostgreSQL, яка забезпечує високу надійність та продуктивність при роботі з великими об'ємами даних. PostgreSQL також забезпечує потужні засоби для роботи з транзакціями та складними запитами.

ABSTRACT

In this course work, we have developed a web application “CompShop”, designed for online sales of computer components. The architecture of the application is based on the Model-View-Controller (MVC) principle, which provides convenient and efficient management of the program logic and its presentation.

The backend is implemented using the Spring framework, which ensures stable and reliable processing of user requests, as well as interaction with the database. The use of Spring makes it easy to implement various functionalities and ensures high scalability of the application.

The frontend is developed on the basis of React, which provides fast and dynamic user interaction. Thanks to the component architecture of React, the application is modular and easy to maintain.

For data storage, the PostgreSQL database is used, which provides high reliability and performance when working with large amounts of data. PostgreSQL also provides powerful tools for handling transactions and complex queries.

ВСТУП

Реалізація перегляду товарів з можливістю відображення детальної інформації про кожен товар: Забезпечення можливості користувачам переглядати доступні товари та отримувати детальну інформацію про кожен з них, включаючи опис, ціну та наявність на складі. Реалізація функції перегляду товарів за категоріями: Надання користувачам можливості фільтрувати товари за різними категоріями для полегшення пошуку необхідного товару. Додавання нових товарів користувачами: Створення функціоналу, який дозволяє користувачам з адміністративними правами додавати нові товари до каталогу магазину. Зберігання товарів, замовлень, користувачів у базі даних: Впровадження системи зберігання даних про товари, замовлення та користувачів у базі даних для забезпечення цілісності та доступності інформації. Реалізація можливості додавання товарів до обраного для подальшого перегляду: Надання користувачам можливості додавати товари до списку обраних для подальшого перегляду та порівняння. Реалізація пагінації для навігації між списками товарів: Впровадження механізму пагінації для зручної навігації між великими списками товарів, що покращує користувацький досвід. Взаємодія з базою даних для ефективного зберігання та обробки інформації: Розробка методів взаємодії з базою даних для ефективного зберігання та обробки інформації, забезпечуючи швидкий доступ до необхідних даних та їх актуальність.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ВИКОРАСТАНІ ТЕХНОЛОГІЇ

* 1. Теоретичні відомості

Spring Framework

Spring — це потужний і гнучкий фреймворк для розробки Java-додатків, що надає всебічну інфраструктуру для розробки корпоративних додатків. Він забезпечує модульність, інверсію управління (IoC), аспектно-орієнтоване програмування (AOP) та інтеграцію з різними технологіями.

Основні компоненти Spring включають:

* Spring Core: Забезпечує основні функції, такі як Inversion of Control (IoC) та Dependency Injection (DI);
* Spring MVC: Модуль для створення веб-додатків, який реалізує шаблон проектування Model-View-Controller (MVC);
* Spring Security: Потужний і гнучкий модуль для аутентифікації та авторизації додатків;
* Spring Data: Підтримка для роботи з базами даних, включаючи JDBC, JPA, та інші.

У даному проекті використано Spring Security з JWT (JSON Web Tokens) для забезпечення безпеки. Spring Security забезпечує аутентифікацію та авторизацію користувачів, захист від атак типу CSRF, а також інтеграцію з різними механізмами аутентифікації.

JWT (JSON Web Tokens) — це відкритий стандарт (RFC 7519), який визначає компактний та самодостатній спосіб для безпечної передачі інформації між сторонами як JSON-об'єкт. Ця інформація може бути перевірена та довірена завдяки цифровому підпису. Використання JWT дозволяє уникнути необхідності зберігання сесій на сервері, що покращує масштабованість додатка.

React

React — це бібліотека для створення інтерфейсів користувача, розроблена компанією Facebook. Вона дозволяє створювати динамічні та інтерактивні веб-додатки шляхом використання компонентів, які можуть мати свій стан (state) та життєвий цикл (lifecycle).

Основні особливості React включають:

* Компонентний підхід: UI розбивається на окремі компоненти, кожен з яких відповідає за свою частину інтерфейсу;
* JSX (JavaScript XML): Синтаксичне розширення JavaScript, яке дозволяє писати HTML-подібний код в JavaScript;
* Virtual DOM: Механізм, який мінімізує кількість маніпуляцій з реальним DOM, що покращує продуктивність додатків.

Vite — це сучасний інструмент для збірки проектів на JavaScript, який забезпечує швидку компіляцію та покращений досвід розробки. Vite використовує сучасні браузерні можливості для швидкої збірки та запуску проектів.

Основні переваги Vite:

* Швидкий запуск серверу розробки: Завдяки використанню нативних ES-модулів;
* Швидка збірка: Використання Rollup під капотом для оптимізації та мінімізації коду;
* Модульність: Підтримка різних плагінів та налаштувань для розширення функціональності.

Взаємодія Spring та React

Веб-додаток "CompShop" поєднує в собі потужність Spring на бекенді та гнучкість React на фронтенді. Бекенд відповідає за обробку запитів, управління безпекою, зберігання даних та виконання бізнес-логіки, тоді як фронтенд забезпечує інтерактивний інтерфейс користувача та динамічну взаємодію з сервером через API.

Ця комбінація дозволяє створити масштабовану, надійну та ефективну систему для управління інтернет-магазином, забезпечуючи при цьому високий рівень користувацького досвіду та безпеки.

2 ПРОГРАМНА РЕАЛІЗАЦІЯ

У цій архітектурі клієнт (браузер) надсилає запити до сервера, а сервер обробляє запит, якщо запит є дійсним, то він відповідає клієнту із запитуваними даними. На клієнті розміщується інтерфейс користувача (UI), а на сервері - бізнес-логіка та база даних.

Переваги:

* Ця архітектура має чіткий розподіл обов'язків між клієнтом і сервером;
* Сервер перевантажується великою кількістю клієнтських запитів;
* Ним легко керувати, а дані можуть бути легко доставлені клієнту.

Недоліки:

* Клієнтські системи можуть отримати вірус або будь-які шкідливі скрипти, якщо такі працюють на сервері;
* Необхідно додати додатковий захист, щоб дані не були підроблені між передачею;
* Основною проблемою може бути непрацездатність сервера. Коли сервер не працює, клієнт втрачає з'єднання і не отримує доступ до даних.

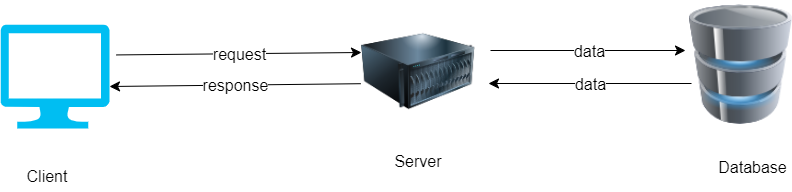


Рисунок 2.1 – Client-Server Architecture

3 ІНСТРУКЦІЯ КОРИСТУВАЧА

3.1 Інструкція користувача

Програма реалізує онлайн магазин, у якому можливо додати та оформити заказ, шукати товари серед усіх товарів, за категоріями та реалізує реєстраці., логін, підтвердження аккаунту, через пошта (SMTP) (рис. 3.1, 3.2).

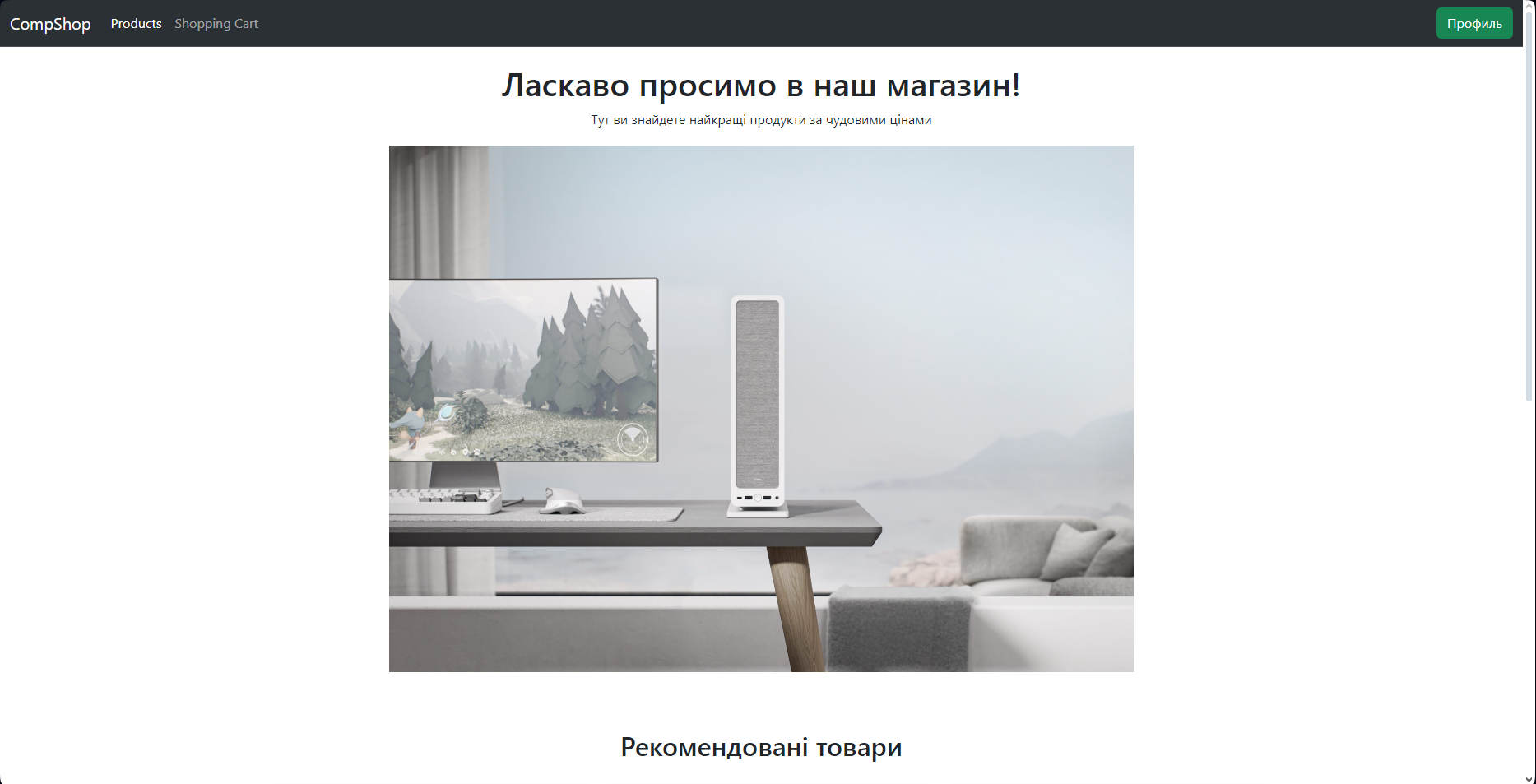


Рисунок 3.1 – Меню програми

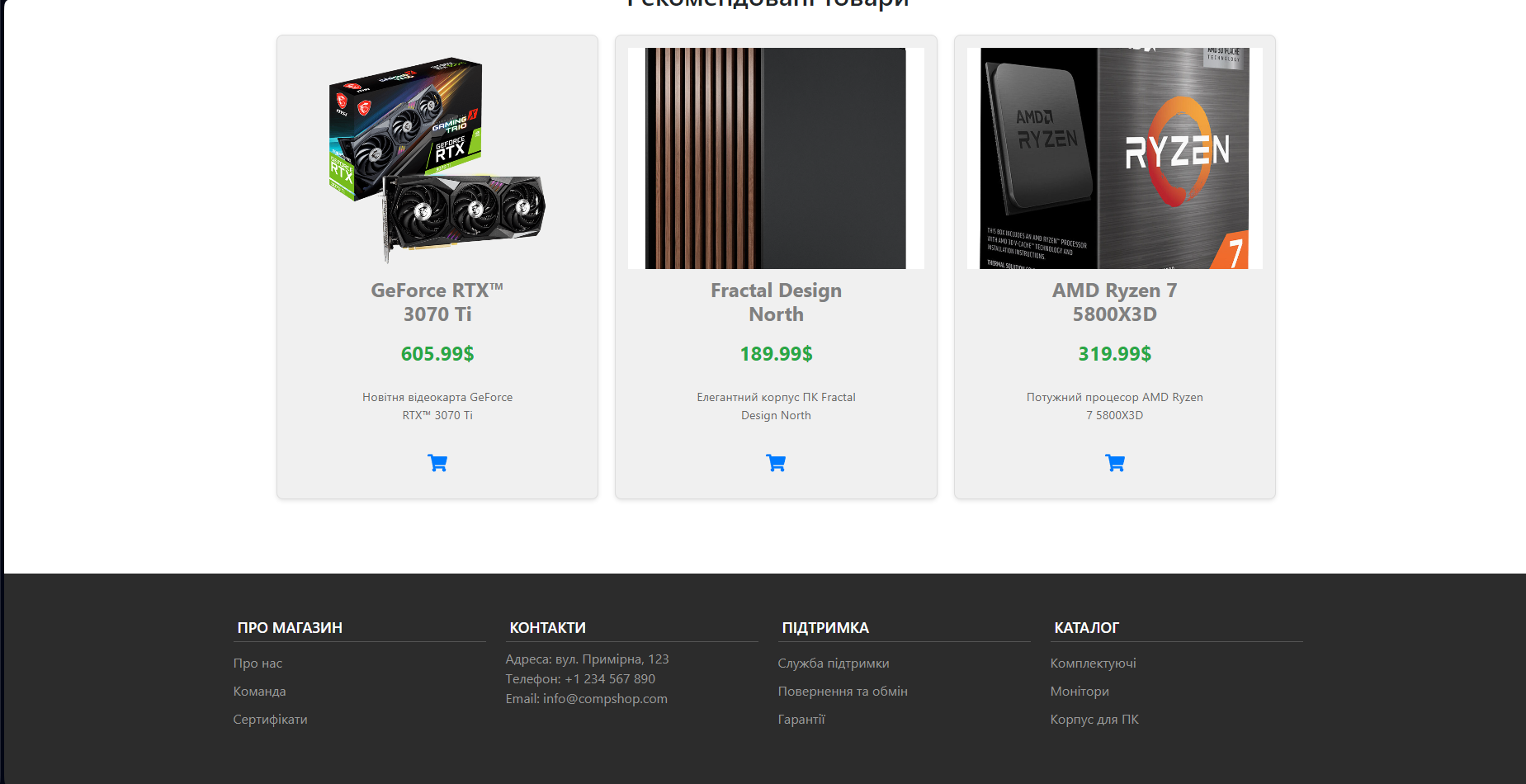


Рисунок 3.2 – Меню програми(продовження)

Натиснемо на кнопку профіль і обираємо реєстрацію, бо профілю у нас ще немає, і заповнюємо поля (рис. 3.3, 3.4).

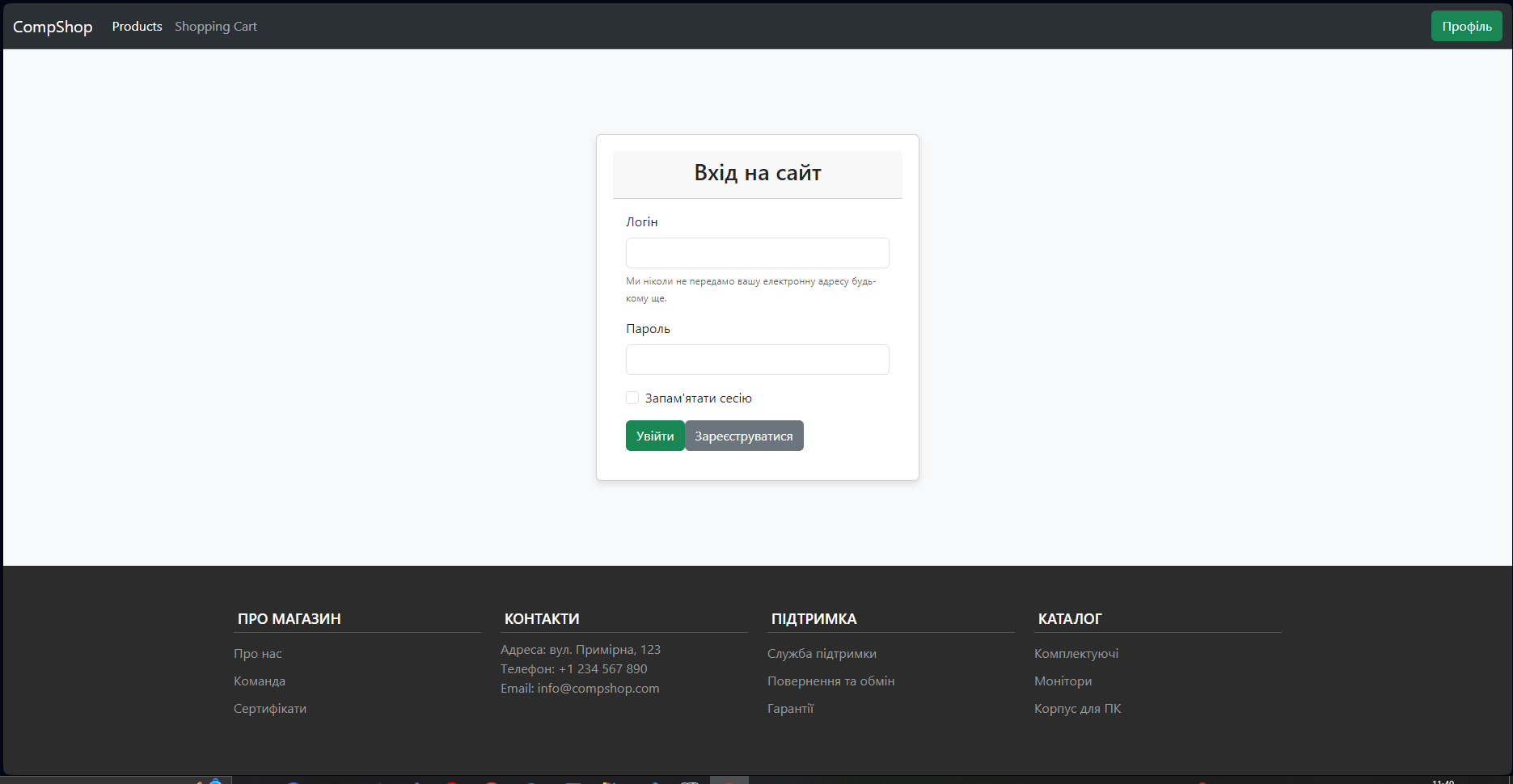


Рисунок 3.3 – Вхід на сайт

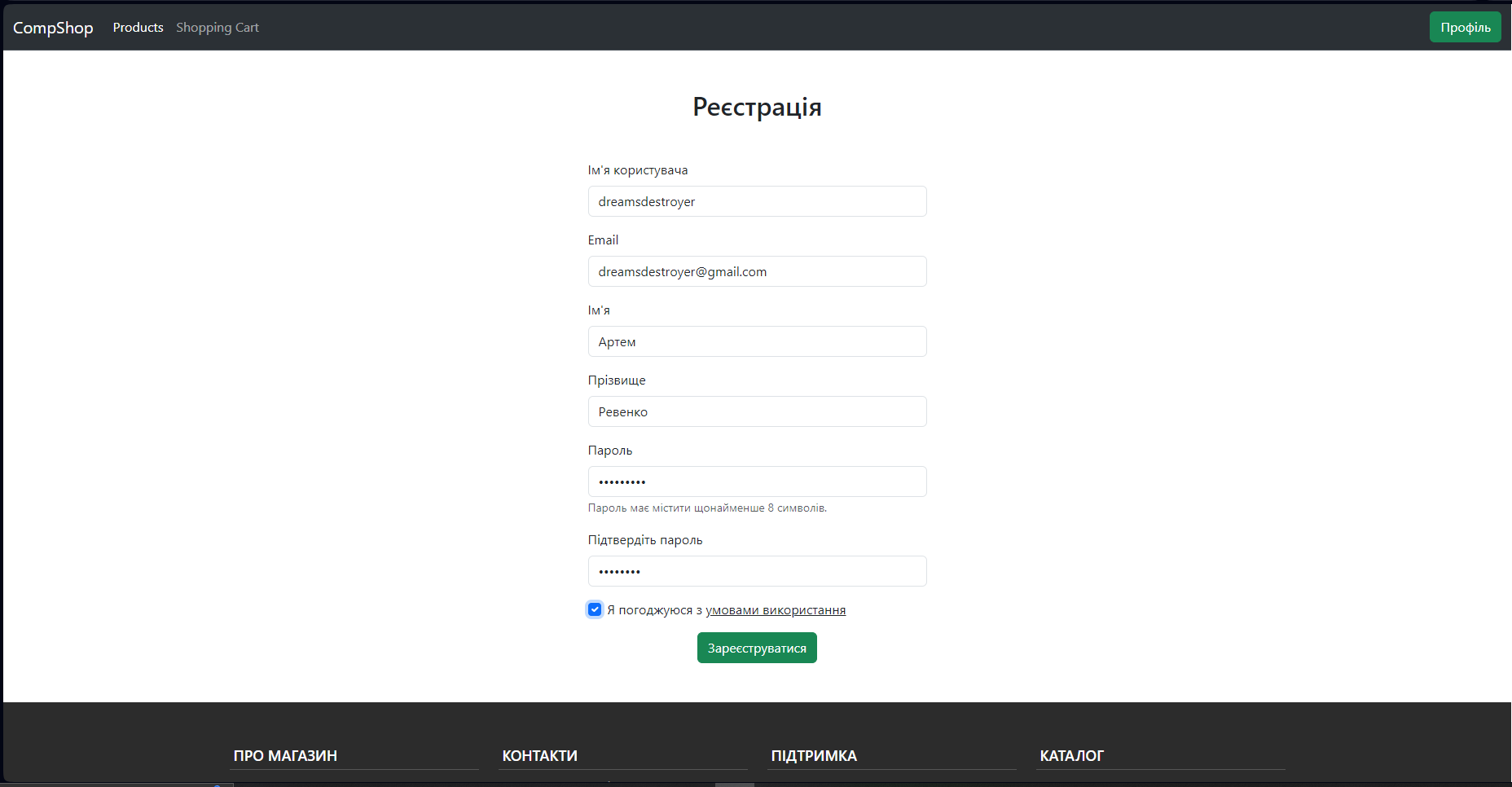


Рисунок 3.4 – Реєстрація профілю

Нам приходить повідомлення на пошту, з повідомленням, будь ласка, підтвердіть пошту і токен.

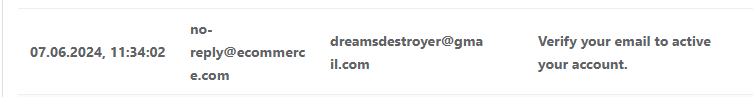


Рисунок 3.5 – Повідомлення підтвердження пошти

Використовуємо токен і підтверджуємо пошту

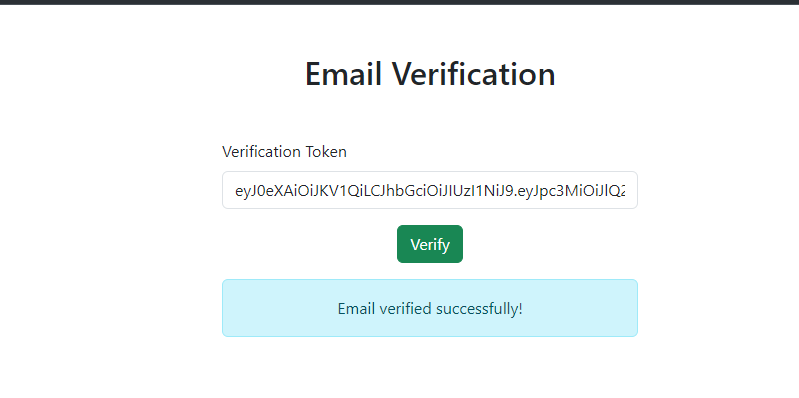


Рисунок 3.6 – Підтвердження пошти

Заповнюємо поля для входу в щойно створений акаунт.

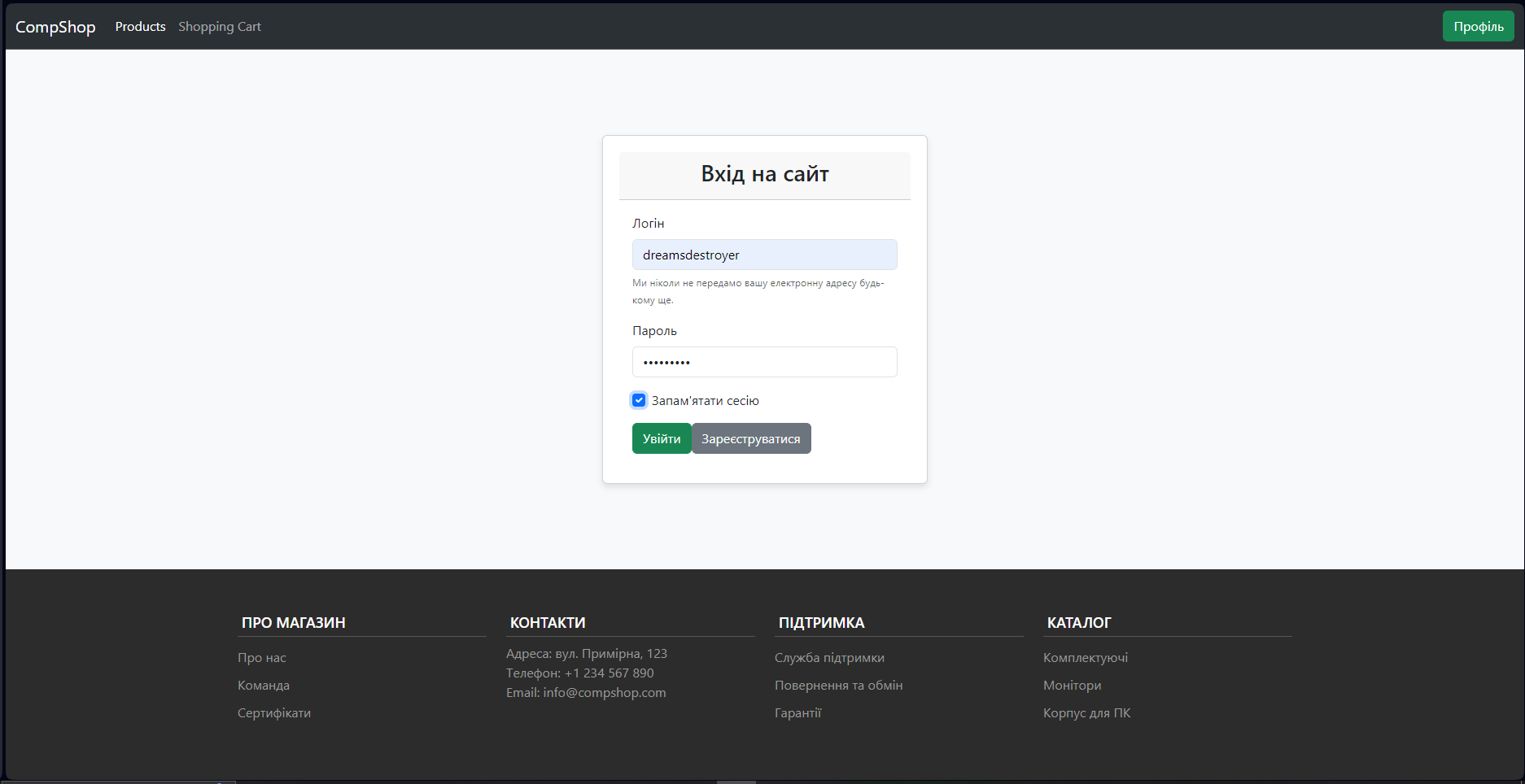


Рисунок 3.7 – Логінімся у профіль

Після успішного входу, нас спрямовує на сторінку з продуктами, обираємо категорію яку хочемо, зараз це буде комплектуючі, і додаємо до кошика, що хочемо.

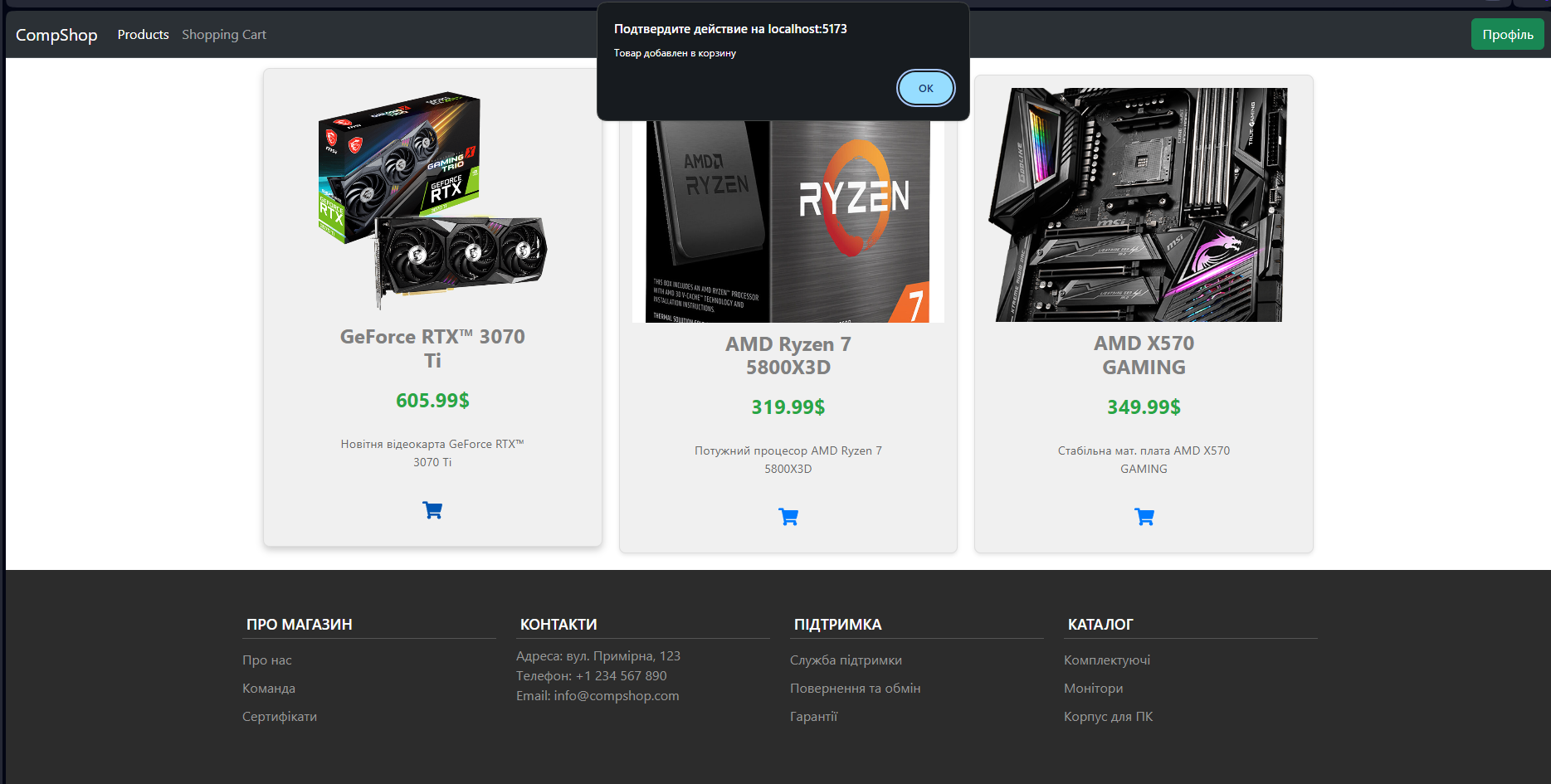


Рисунок 3.8 – Додавання до кошику товарів

Переходимо до кошику, та змінюємо кількість бажаних товарів.

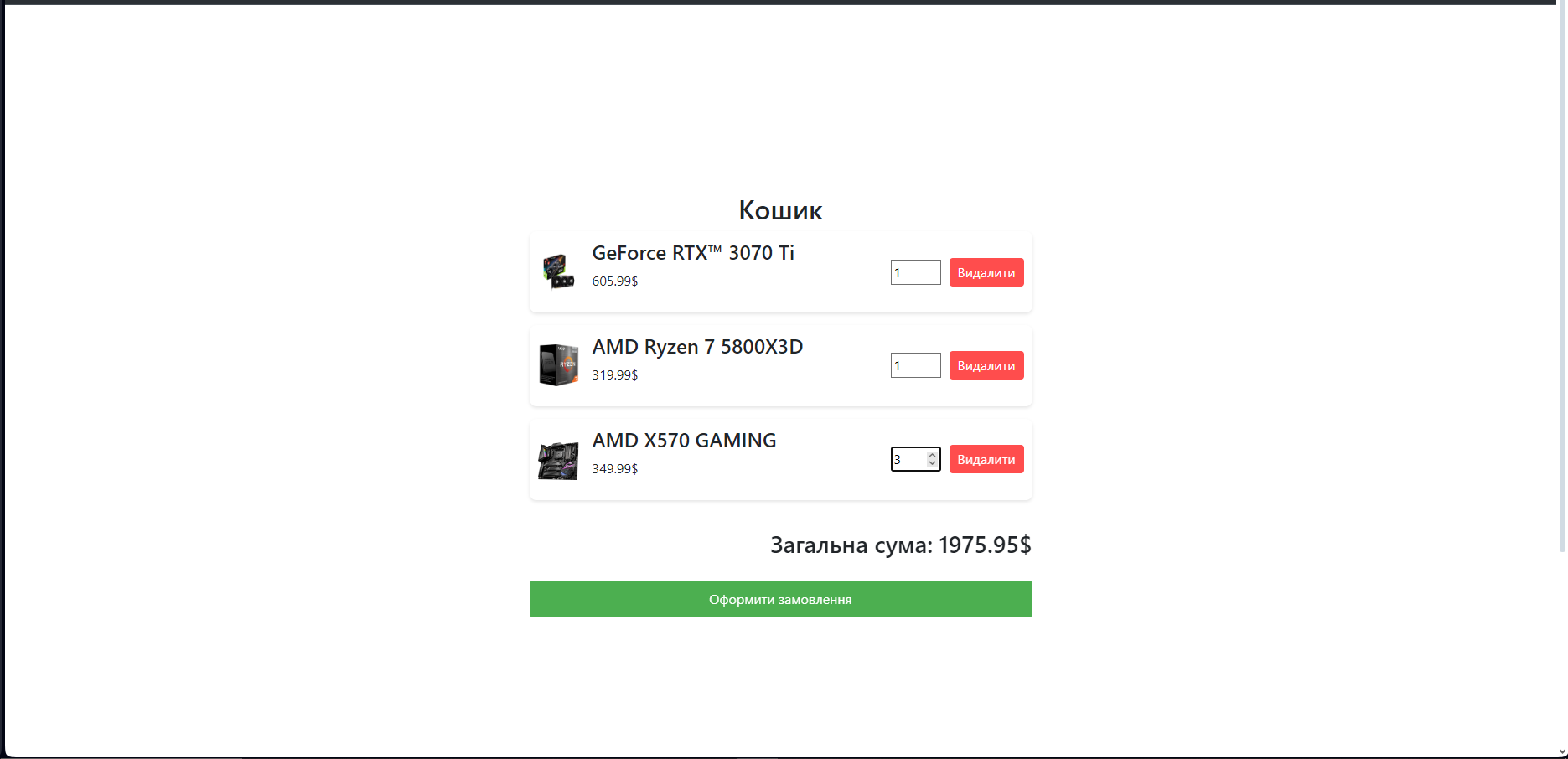


Рисунок 3.9 – Зміна кількості товару у кошику

Видалення бажаного товару з кошику, за допомогою кнопки видалити.

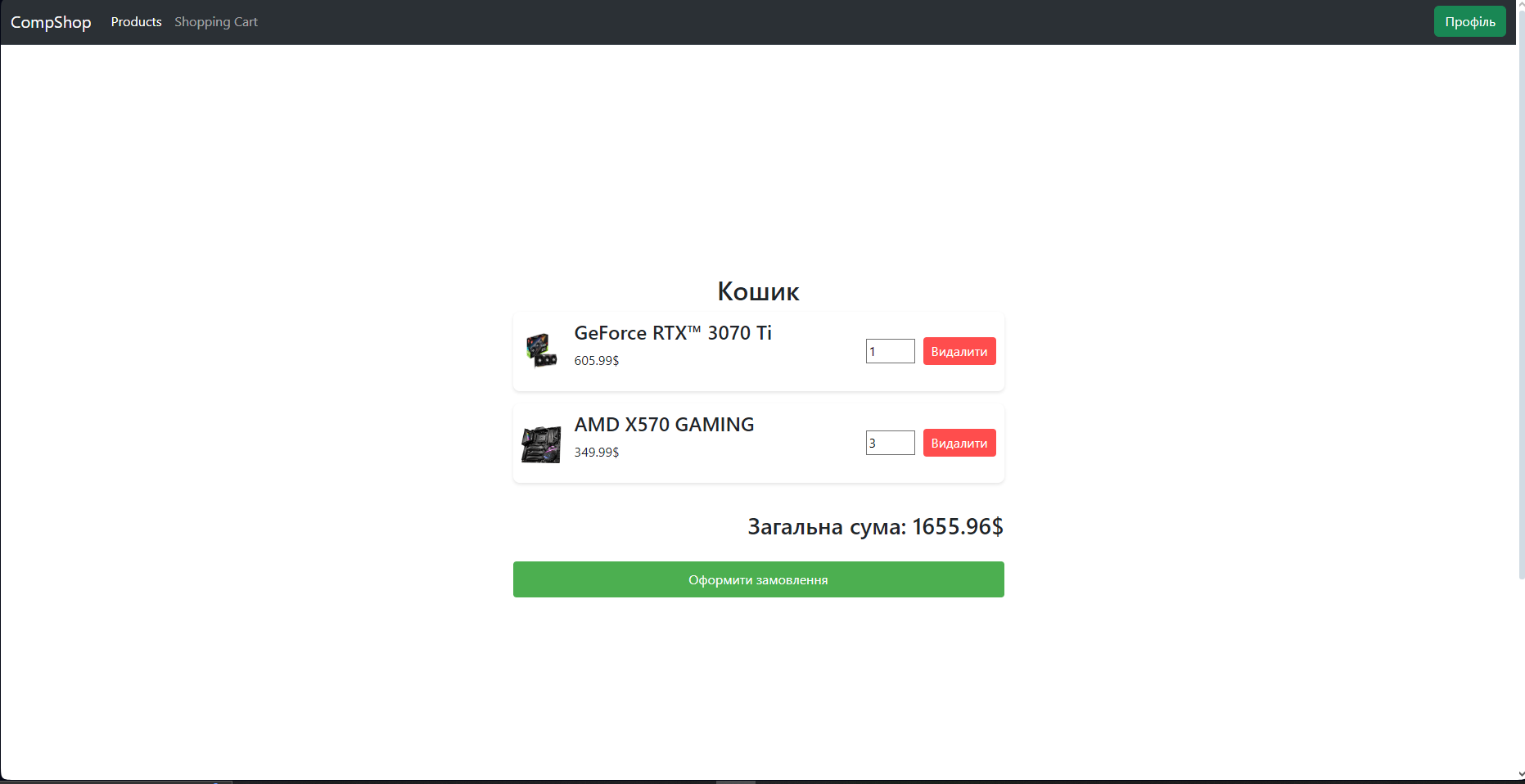


Рисунок 3.10 – Видалення одного товару з кошику

Заповнюємо усі поля для адреси доставки та підтверджуємо наш заказ, якщо все вірно.

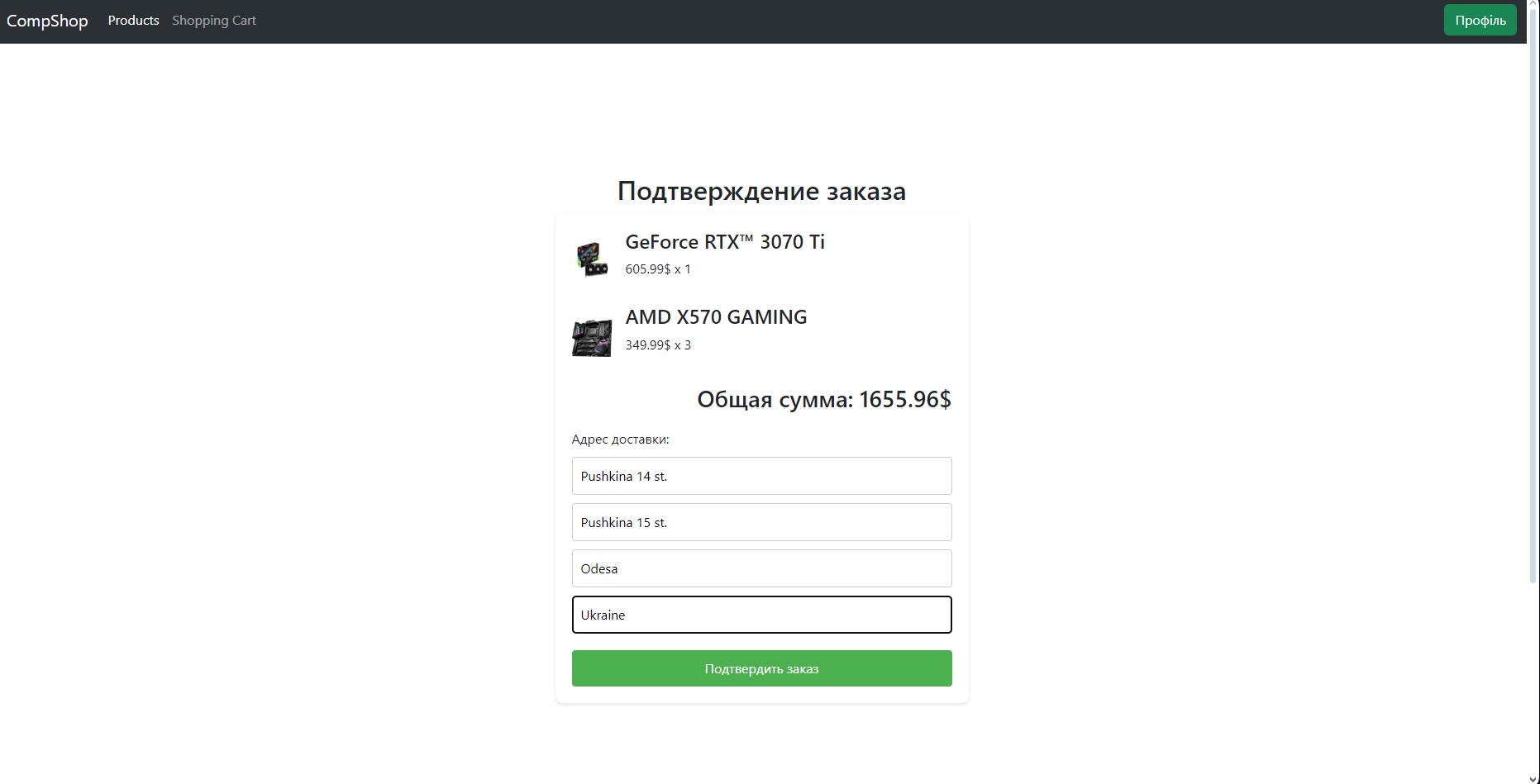


Рисунок 3.11 – Підтвердження заказу

Отримуємо це, якщо формування заказу пройшло успішно.

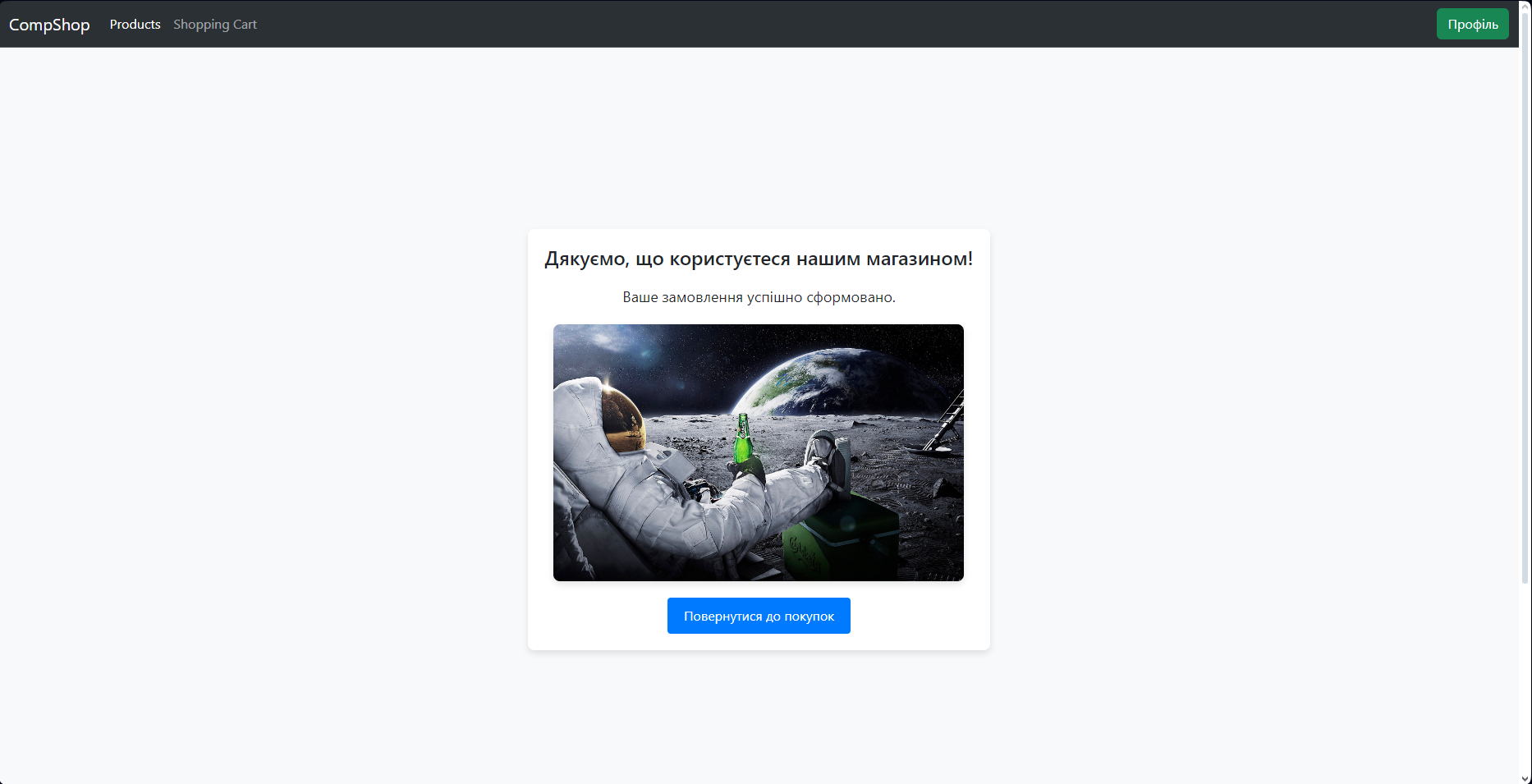


Рисунок 3.12 – Успішне формування заказу

Після успішного формування закаказу ми можемо придбати більше товарів.

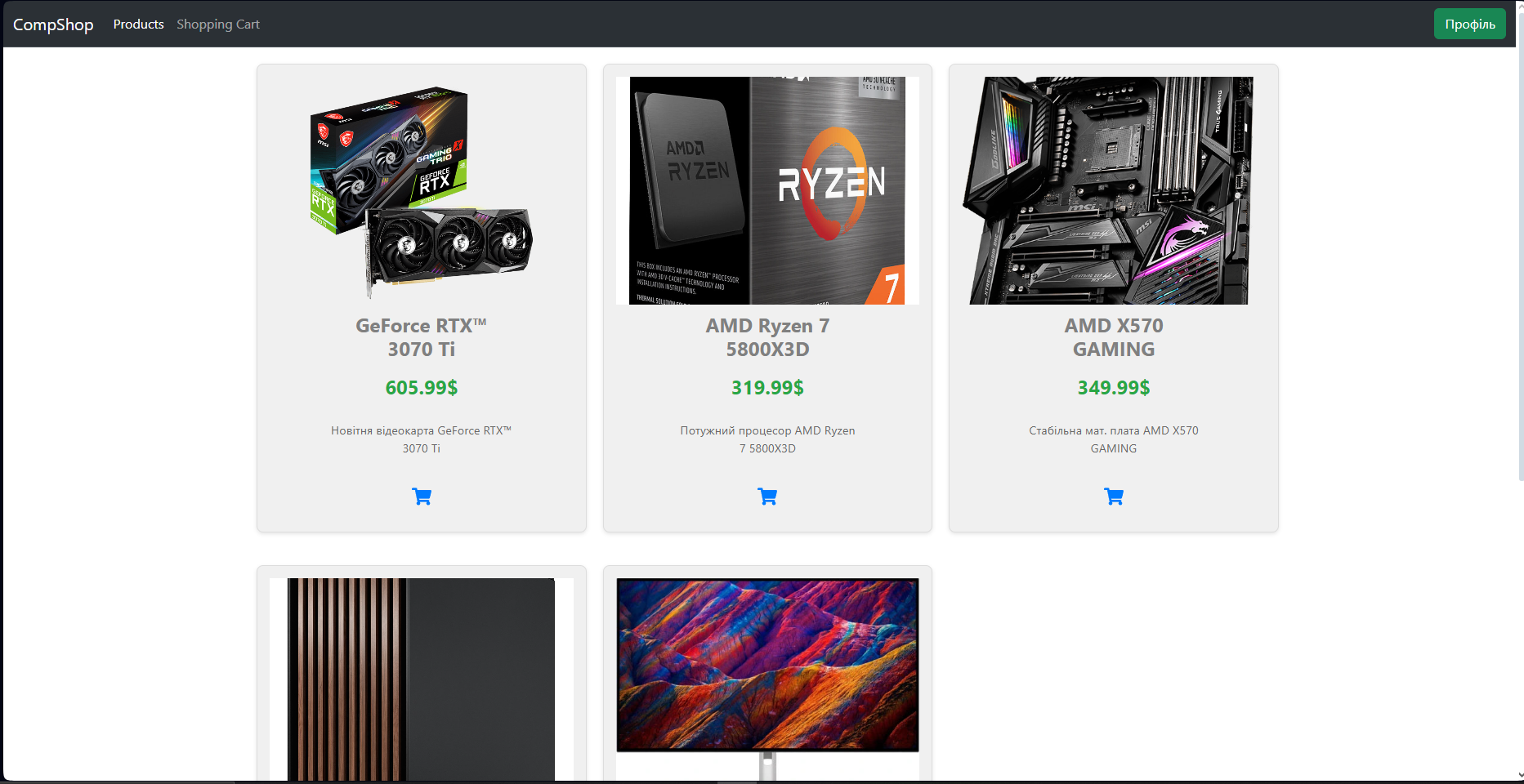


Рисунок 3.13 – Повернення до покупок

ВИСНОВКИ

У даній курсовій роботі було, створено зручний та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко додавати товари в кошик та оформлювати замовлення. Це забезпечує зручність користування та підвищує рівень задоволеності користувачів. Реалізовано функціонал, який дозволяє користувачам переглядати товари з детальною інформацією про кожен з них. Це включає опис товару, ціну, наявність на складі та інші характеристики, що допомагає користувачам зробити обґрунтований вибір. Запроваджено можливість переглядати товари за різними категоріями, що дозволяє користувачам швидко знаходити потрібні товари та робить процес пошуку більш ефективним. Забезпечено функціонал, що дозволяє користувачам додавати нові товари. Це сприяє наповненню каталогу товарів та підвищує рівень залученості користувачів. Реалізовано надійну систему зберігання даних, яка включає товари, замовлення та інформацію про користувачів. Це забезпечує цілісність даних та їх доступність для подальшої обробки. Користувачі можуть додавати товари до списку обраних для подальшого перегляду, що покращує їхній досвід взаємодії з додатком та допомагає зберегти інтерес до конкретних товарів. Впроваджено пагінацію, яка дозволяє користувачам легко переміщуватись між сторінками зі списком товарів, що покращує навігацію та робить процес перегляду товарів більш зручним. Розроблено ефективну взаємодію з базою даних, яка забезпечує швидке та надійне зберігання та обробку інформації. Це включає використання сучасних технологій для оптимізації запитів та забезпечення високої продуктивності системи.В цілому, веб-сайт має високий рівень функціональності та структури, що дозволяє забезпечити зручну та ефективну взаємодію з користувачем.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

* 1. Vite, URL: <https://vitejs.dev/>
  2. Spring, URL: <https://spring.io/>
  3. React-Dev, URL: <https://react.dev/>

ДОДАТОК А

Увесь код програми можна подивитись за посиланням:

<https://github.com/projectApelsin/spring-react-e-commerce>