E-COMMERCE

TRIBHUVAN INIVERSITY

AMRIT SCIENCE CAMPUS

Thamel, Kathmandu

Submitted By:
Bishnu Chalise
Arjun Mijar
Bijaya Dhakal
Lokesh Nath Yogi

Faculty: CSIT
Section: A
Combination: CSIT 6th Sem

Submitted To: Gyani Ray

Internal Examiner
Signature: _____

External Examiner
Signature: _____

Acknowledgment

We, the project team behind the E-commerce Website, would like to express our sincere gratitude to everyone who has supported and guided us during the development of this project.

We would like to thank our supervisor and faculty members for their valuable feedback and encouragement. We also acknowledge the support of our peers and the open-source communities behind the technologies we used: **React, Next.js, Prisma ORM, PostgreSQL, Zustand, and Vercel**.

Lastly, we are grateful to **Tribhuvan University, Institute of Science and Technology**, for providing us with this opportunity to apply our academic knowledge to practical application through this project.

Abstract

This project report presents the development and implementation of our **E-commerce Website**. The platform provides a modern and user-friendly online shopping experience.

The system was built using **React** for the frontend and **Next.js** for the backend, with **Prisma ORM** and **PostgreSQL** for efficient database management. **Zustand** is used for state management. Deployment is handled by **Vercel** for the application and **Neon** for the database.

Key features include a **product catalog, search and filter, cart and checkout system, caching strategies, and indexing**. This project demonstrates our understanding of full-stack web development, system design, database integration, and deployment in a real-world scenario.

Keywords: React, Next.js, Prisma ORM, PostgreSQL, Zustand, Vercel, Neon, E-commerce

Table of Contents

Acknowledgment 2

Abstract 3

1. Introduction 6

1.1 Background 6

1.2 Objectives 6

1.3 Scope 6

1.4 Methodology 7

1.5 Market Analysis 7

2. Features and Functionality 8

2.1 Intuitive Homepage Design 8

2.2 Product Catalog 8

2.3 Search and Filtering 8

2.4 Shopping Cart and Checkout 8

2.5 Caching and Personalization 8

2.6 Admin Panel 8

3. Tools and Technologies 9

3.1 Frontend: React 9

3.2 Backend: Next.js 9

3.3 Database: Prisma ORM with PostgreSQL 9

3.4 State Management: Zustand 9

3.5 Hosting: Vercel & Neon 9

3.6 Version Control: Git & GitHub 9

4. System Design 10

4.1 Architecture 10

4.2 Database Design 10

4.3 User Interface Design 10

System Design Diagrams 11

5. Implementation 15

5.1 Frontend (React) 15

5.2 Backend (Next.js) 15

5.3 Database (Prisma + PostgreSQL) 16

6. Testing 17
7. Deployment 18
9. Conclusion 20
References 21
Appendix (Screenshots) 22

1. Introduction

1.1 Background

The rapid growth of the internet and digital technologies has fundamentally transformed how businesses operate and how consumers shop. Traditional shopping methods are increasingly being replaced by **e-commerce platforms** that provide convenience, variety, and speed. In Nepal, too, the adoption of online shopping is rising as customers demand platforms that are secure, user-friendly, and optimized for performance.

Our project, the **E-commerce Website**, is developed to meet these needs. It allows users to **browse, search, filter, and purchase products online** with ease. By integrating modern frameworks such as **React** and **Next.js**, and efficient tools like **Prisma ORM** and **PostgreSQL**, the website demonstrates how academic knowledge can be applied to build practical, real-world solutions.

1.2 Objectives

The primary objectives of this project are:

- To **design and implement** a responsive and user-friendly e-commerce website.
- To integrate a **secure and scalable backend** using Next.js and Prisma ORM.
- To provide efficient **data management** using PostgreSQL.
- To implement **client-side caching** and personalization to improve performance.
- To ensure **fast indexing and optimized fetching** for product data.
- To deploy the system on **Vercel (frontend & backend)** and **Neon (database)** for global accessibility.

1.3 Scope

The scope of this project includes the **development of a full-stack e-commerce platform** that provides:

- **Frontend interface** for customers to browse, search, and purchase products.
- **Backend services** to handle authentication, orders, and database queries.
- **Database management** using PostgreSQL with Prisma ORM for structured data storage.
- **State management** using Zustand for cart and session handling.
- **Caching** mechanisms for personalized and faster user experiences.
- **Deployment** on cloud-based services ensuring scalability and availability.

1.4 Methodology

The development process followed a **semi-waterfall model** with iterative improvements. The phases included:

1. **Requirement Analysis** – Identified the core needs of users: browsing products, searching efficiently, and completing secure checkouts.
2. **Design** – Created wireframes, UI mockups, and the system architecture including database schema.
3. **Development** – Implemented frontend (React), backend (Next.js), and database (Prisma + PostgreSQL).
4. **Testing** – Performed unit, integration, performance, and security testing to ensure reliability.
5. **Deployment** – Hosted the frontend and backend on **Vercel**, and the database on **Neon**.

1.5 Market Analysis

The global e-commerce market continues to grow at a rapid pace. According to recent studies:

- The global e-commerce industry is projected to exceed **$8 trillion by 2027**.
- Customers increasingly prefer **mobile-friendly, fast, and personalized platforms**.
- In Nepal, the growth of **digital wallets, online banking, and delivery services** has created favorable conditions for e-commerce adoption.

Our project leverages these market trends by focusing on **user experience, caching, personalization, and indexing**, making it competitive and future-ready.

2. Features and Functionality

The **E-commerce Website** provides the following features:

2.1 Intuitive Homepage Design

- Clean, responsive, and visually appealing interface.
- Featured products section for promotions.
- Easy navigation through categories.

2.2 Product Catalog

- Products displayed with **images, descriptions, and prices**.
- Supports categorization for efficient browsing.

2.3 Search and Filtering

- Search products by **name, category, or keyword**.
- Filters by price, category, and relevance.

2.4 Shopping Cart and Checkout

- Add/remove products to cart.
- Real-time updates using **Zustand state management**.
- Smooth and secure checkout process.

2.5 Caching and Personalization

- **Client-side caching** improves load times.
- **Personal caching** stores frequently accessed products for each user.

2.6 Admin Panel

- Admins can **add, edit, or remove products**.
- User and order management features.
- Performance monitoring and indexing for database queries.

3. Tools and Technologies

3.1 Frontend: React – Used to build the user interface with a component-based design, ensuring responsiveness and smooth navigation for product browsing, cart, and checkout pages.

3.2 Backend: Next.js – Handles server-side rendering for performance and SEO, and provides API routes for authentication, product fetching, and order management.

3.3 Database: Prisma ORM with PostgreSQL – Prisma simplifies database queries and schema management, while PostgreSQL stores structured data such as users, products, and orders with indexing for faster searches.

3.4 State Management: Zustand – Manages global state like shopping cart and user session in a lightweight and efficient way.

3.5 Hosting: Vercel & Neon – Vercel hosts the frontend and backend with automatic deployment, while Neon provides cloud hosting for the PostgreSQL database.

3.6 Version Control: Git & GitHub – Used to track changes in the source code, collaborate in development, and integrate with Vercel for continuous deployment.

4. System Design

4.1 Architecture

- **React frontend** handles user interaction.
- **Next.js backend** provides API endpoints and business logic.
- **Prisma ORM** communicates with **PostgreSQL database**.
- **Zustand** manages application state (cart, user session).
- Deployment: **Vercel** (web app) + **Neon** (database).

4.2 Database Design

- **Users table** (user_id, name, email, password, role).
- **Products table** (product_id, name, category, price, stock).
- **Orders table** (order_id, user_id, total_price, status).
- **Cart table** (cart_id, user_id, product_id, quantity).
- **Categories table** (category_id, category_name).

4.3 User Interface Design

- **Responsive layout** using React.
- **Minimalist design** for fast navigation.
- **Mobile-first approach** ensures smooth experience across devices.

System Design Diagrams

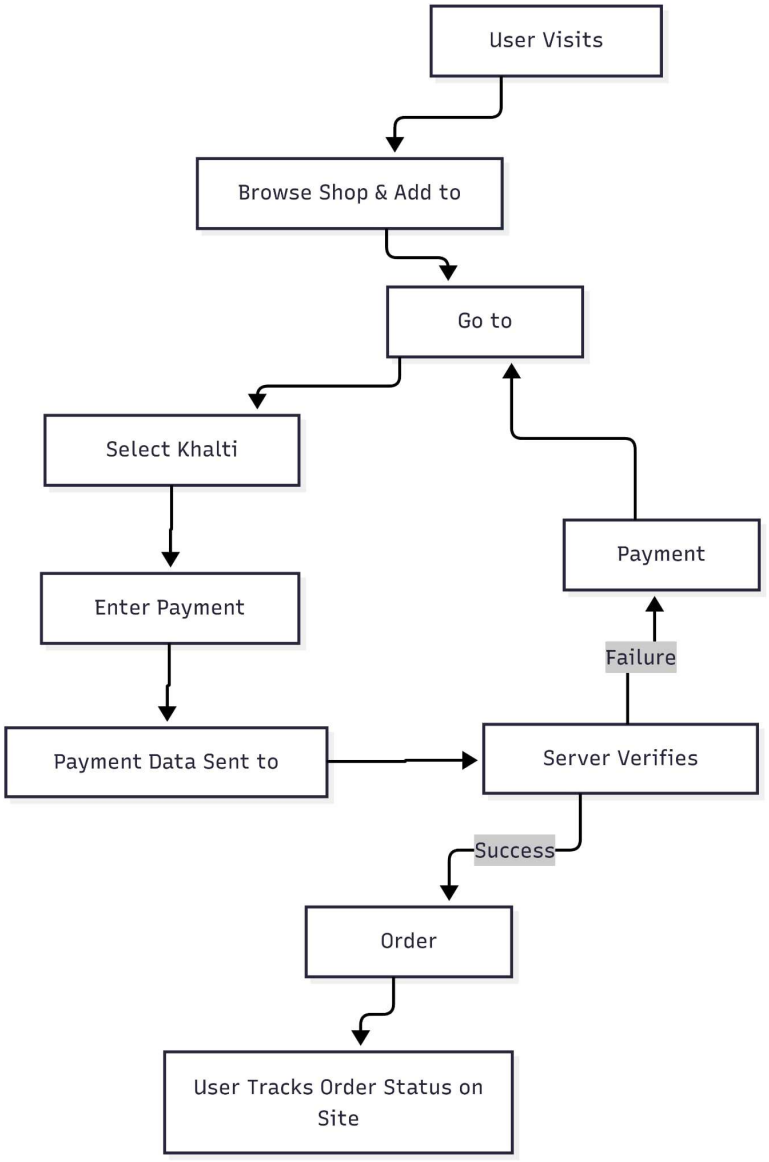

Fig: Flowchart Diagram

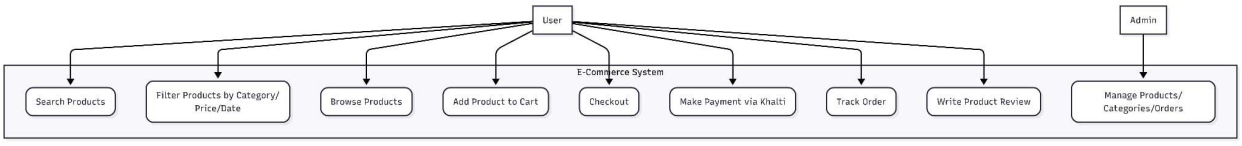

Fig: Use Case Diagram

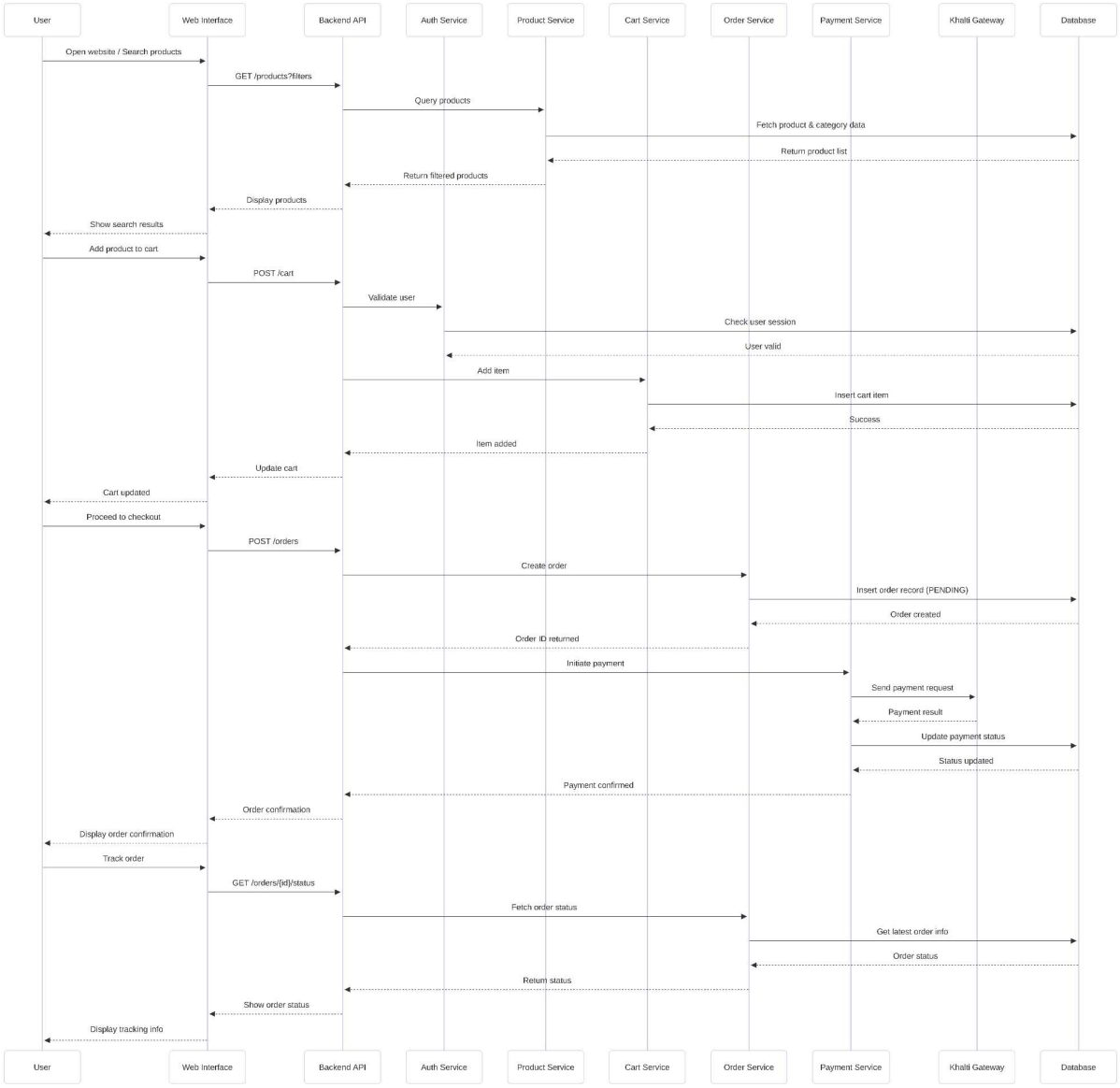

Fig: Sequence diagram

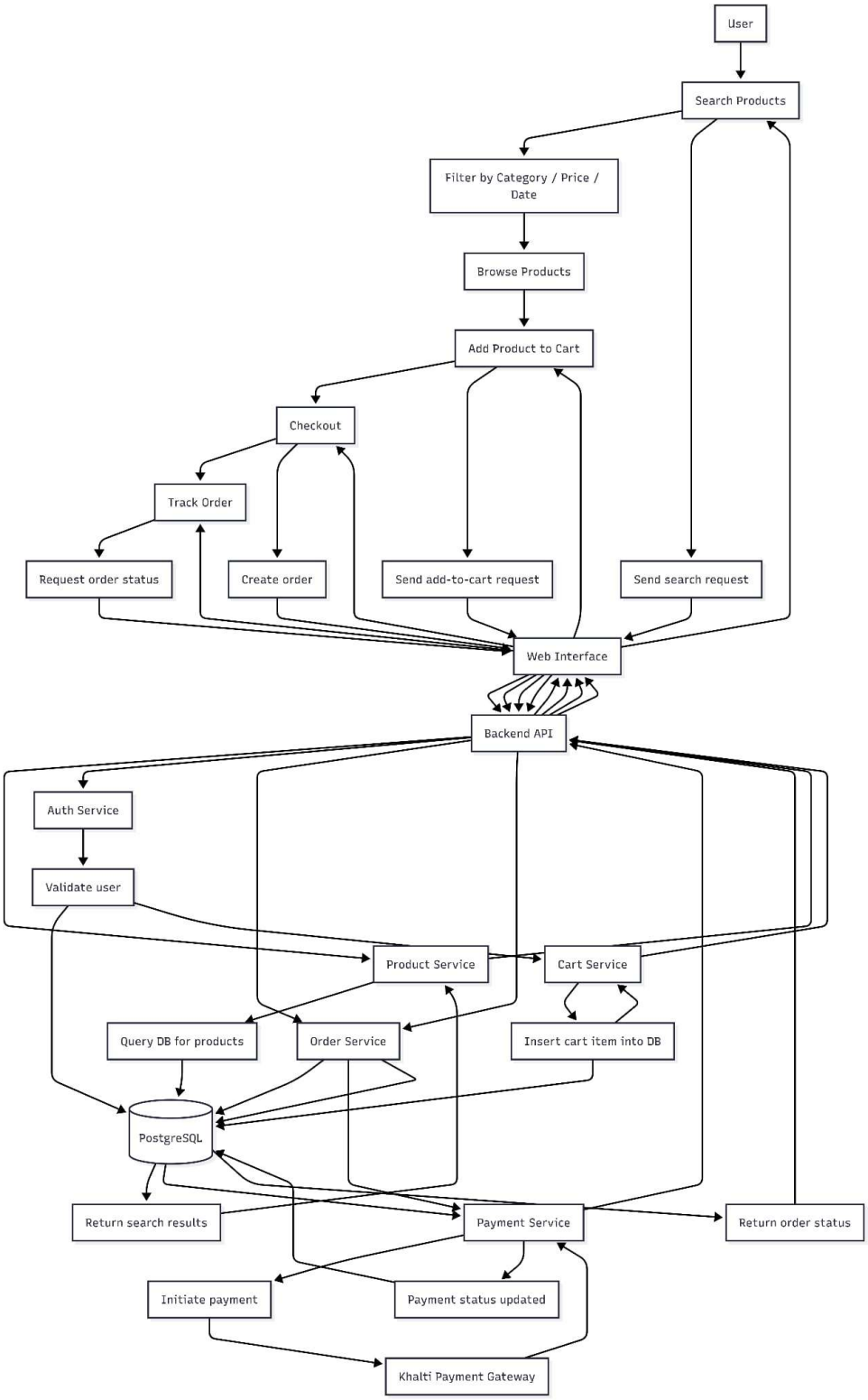

Fig: Workflow Diagram / Detailed Flowchart

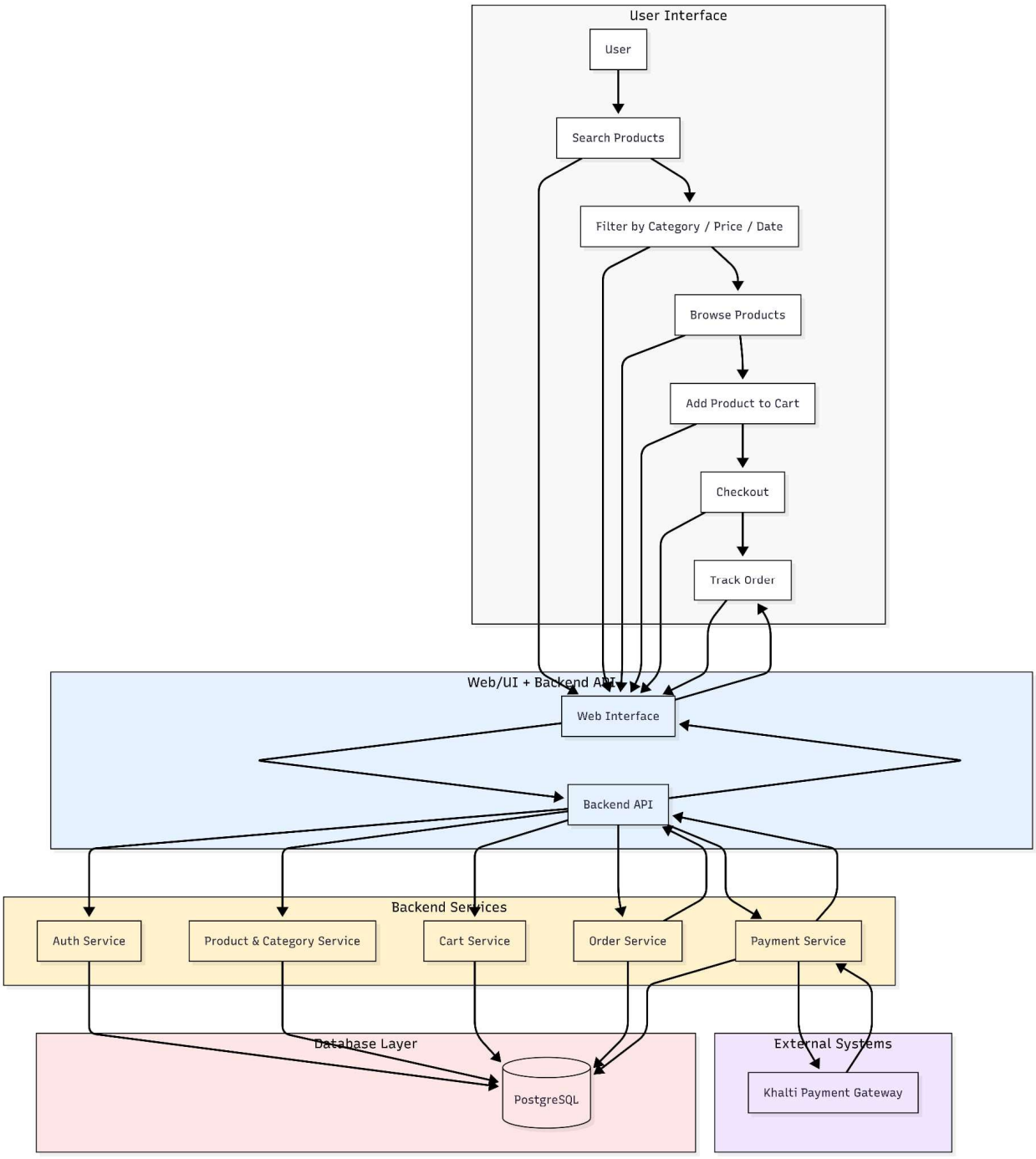

Fig: Architecture Diagram

5. Implementation

This chapter details the implementation of the **e-commerce** platform, highlighting key features and technologies used in the frontend, backend, and database layers.

5.1 Frontend (React)

Component-Based Architecture: The UI is built using reusable React components, ensuring modularity, reusability, and easier maintenance.

- **State Management with Zustand**: Provides a lightweight and efficient solution for managing global state, including cart items, authentication status, and user preferences.
- **Responsive Design with Tailwind CSS**: Tailwind's utility-first classes are used to ensure the platform adapts seamlessly to various screen sizes and devices.
- **Routing (Next.js)**: Client-side routing powered by Next.js ensures fast and smooth navigation.
- **Lazy Loading**: Components and images are loaded on demand to improve performance, especially on mobile devices.
- **API Integration (Axios/Fetch)**: The frontend communicates with backend API routes for data fetching and submission.

5.2 Backend (Next.js)

API Routes: Next.js API routes handle authentication, product management, checkout processes, and order tracking.

- **Server-Side Rendering (SSR)**: Improves SEO and initial load speed by rendering product pages and categories on the server.
- **Authentication Logic**: JWT-based authentication ensures secure login, signup, and session handling.
- **Business Logic**: Core functionalities such as cart management, order processing, and user accounts are implemented in backend routes.
- **Middleware**: Used for request validation, authentication checks, and logging.
- **Payment Gateway Integration**: APIs (Stripe or local providers like eSewa/Khalti) are connected for secure checkout and payment processing.
- **Security Practices**: Includes input validation, HTTPS, rate limiting, and protection against SQL injection and XSS attacks.

5.3 Database (Prisma + PostgreSQL)

Prisma Schema: Defines models for users, products, orders, reviews, and relationships between entities. Prisma migrations manage schema evolution.

- **Relational Database (PostgreSQL)**: Stores transactional and relational data such as product catalogs, user accounts, and order histories efficiently.
- **Indexed Queries**: Frequently queried fields like product titles, categories, and user IDs are indexed for faster searches and lookups.
- **Data Integrity**: Enforced through foreign keys, constraints, and relational mapping (e.g., one-to-many between users and orders).
- **Scalability**: PostgreSQL provides ACID compliance and can handle high transaction volumes.

6. Testing

To ensure the reliability, performance, and usability of our E-commerce Website, we carried out the following testing approaches:

- **Unit Testing** – Individual React components (such as product cards, cart items) and Next.js API routes were tested to confirm correct functionality in isolation.
- **Integration Testing** – Verified smooth interaction between the **frontend, backend, and database** by testing workflows like adding products to the cart and placing an order.
- **Performance Testing** – Measured page loading speed and API response times. Client-side caching and indexing were tested to ensure faster performance under multiple user requests.
- **Security Testing** – Checked authentication flow, login system, and protected API endpoints. SQL safety and input validation were tested to prevent common vulnerabilities.
- **User Acceptance Testing (UAT)** – The website was tested by peers to ensure it was **easy to use, responsive, and met the expected requirements**, with feedback used to improve the interface and flow.

7. Deployment

The deployment process ensured that the website is accessible, secure, and scalable for real users:

- **Frontend & Backend** – Deployed together on **Vercel**, which provides fast global delivery, automatic builds from GitHub, and continuous deployment.
- **Database** – Hosted on **Neon**, a cloud-based PostgreSQL platform offering scalability and backups for data reliability.
- **SSL (HTTPS)** – Enabled by default on Vercel to secure communication between client and server.
- **Caching** – Implemented client-side and personal caching strategies to improve response time and reduce server load.

8. Future Scope

To enhance the platform and make it more competitive, the following improvements can be implemented in the future:

- **AI-based Product Recommendations** – Suggest products based on user behavior and purchase history to personalize shopping.
- **Mobile Application (Android & iOS)** – Extend the platform to native mobile apps for better accessibility and user convenience.
- **Integration with Digital Wallets** – Add popular Nepali payment options such as **eSewa, Khalti, and FonePay** for seamless transactions.
- **Admin Analytics Dashboard** – Provide admins with real-time insights on sales, traffic, and customer behavior.
- **Multi-language & Multi-currency Support** – Expand reach by supporting different languages and currencies for diverse users.

9. Conclusion

The **E-commerce Website project** successfully demonstrates how modern web development frameworks and tools can be used to design and implement a scalable, user-friendly, and secure platform.

By combining **React, Next.js, Prisma ORM, PostgreSQL, Zustand, Vercel, and Neon**, we have created a complete system capable of handling **product browsing, searching, caching, checkout, and database operations**.

The project reflects both **academic learning and practical application**, providing a foundation for future enhancements such as AI, analytics, and mobile applications.

References

- React Documentation (react.dev)
- Next.js Documentation (nextjs.org)
- Prisma ORM Documentation (prisma.io)
- PostgreSQL Official Documentation (postgresql.org)
- Zustand State Management Docs
- Vercel Deployment Docs
- Neon Database Hosting Docs

Appendix (Screenshots)

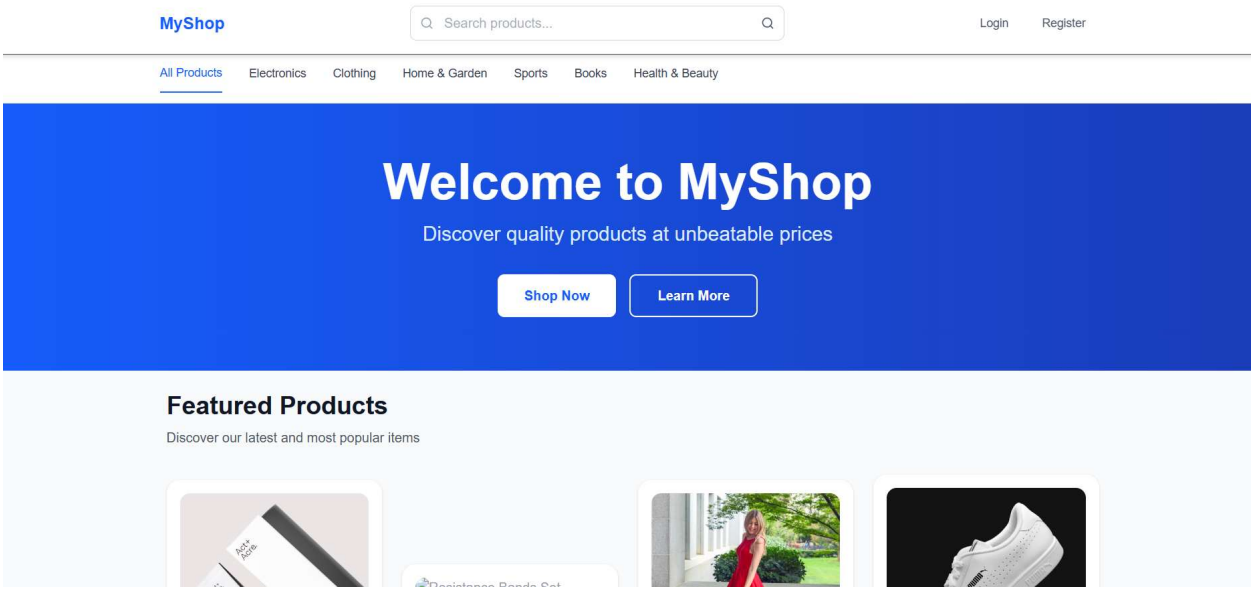

Fig: Home page

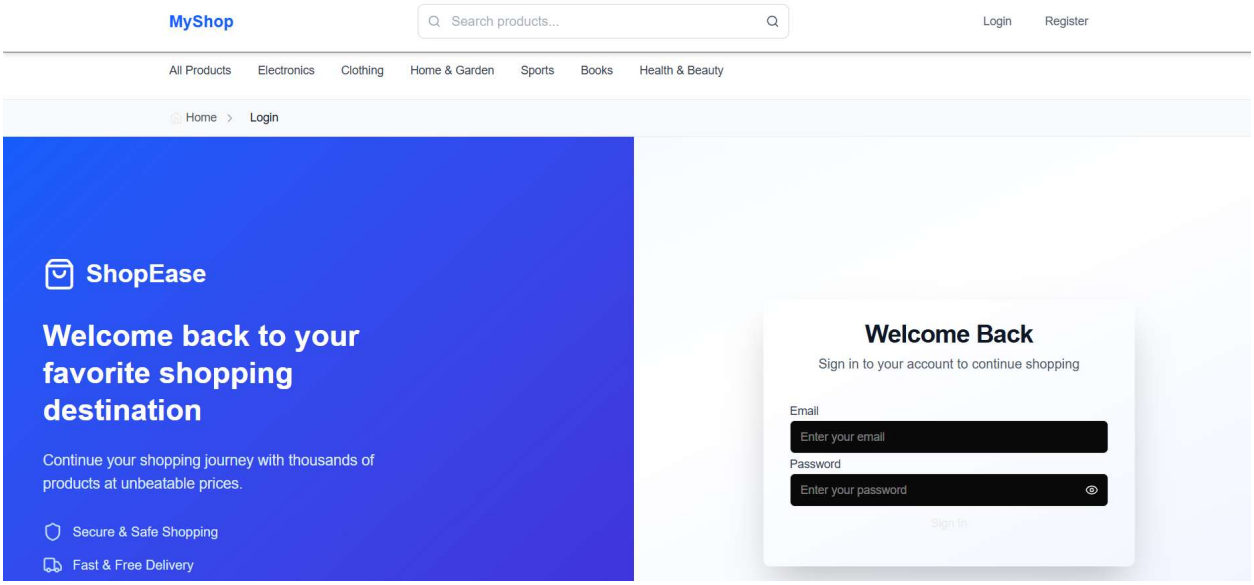

Fig: Login Page

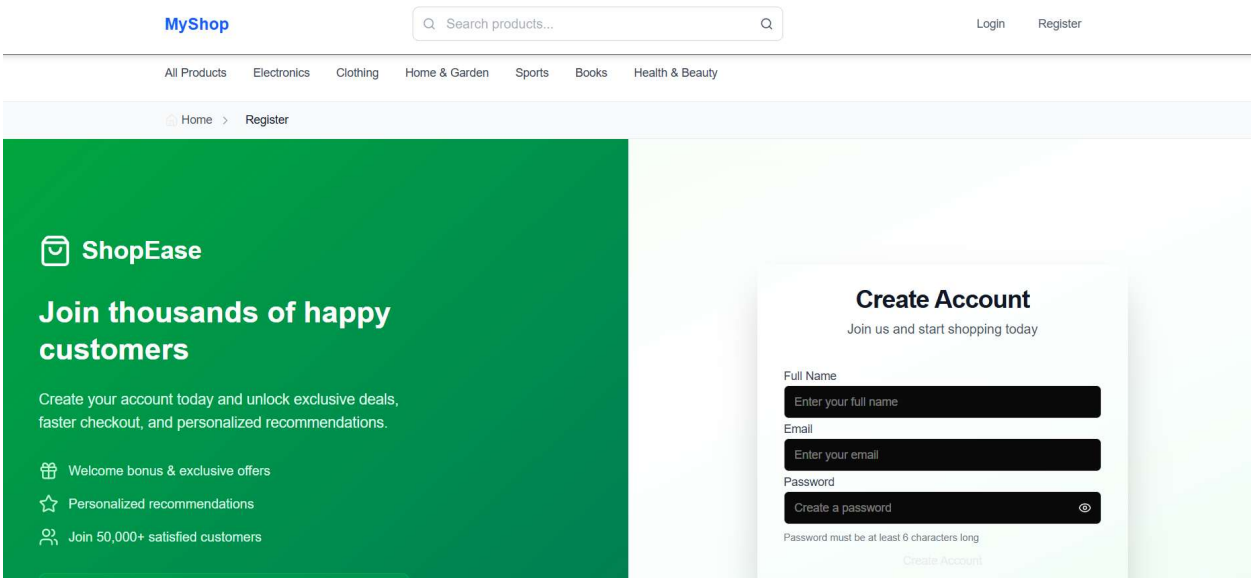

Fig: Register Page

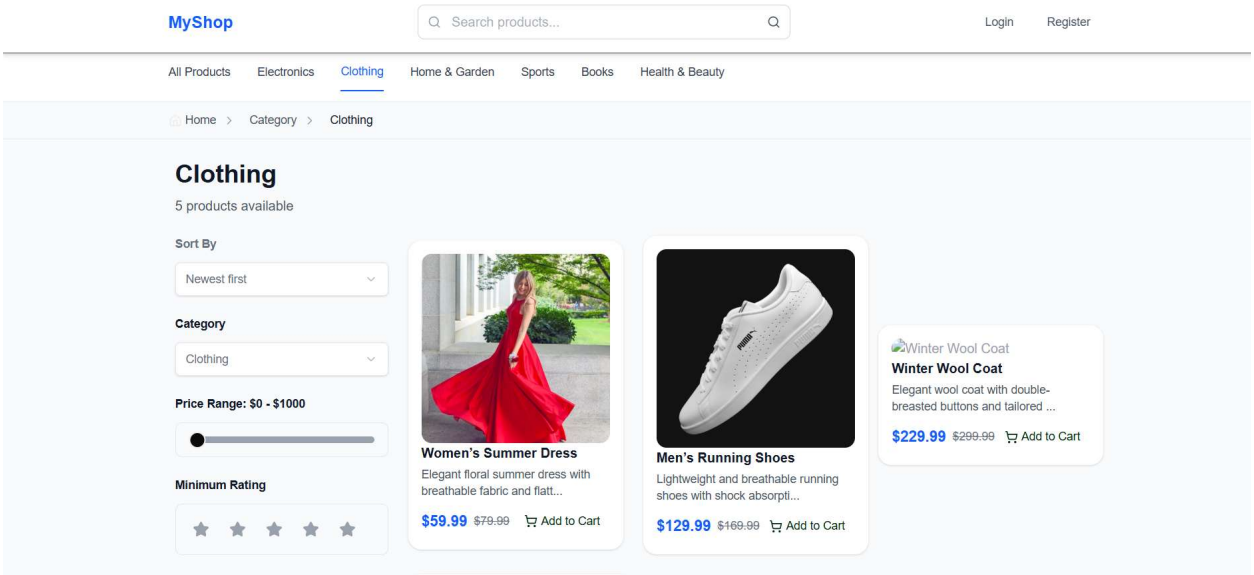

Fig: Product Page

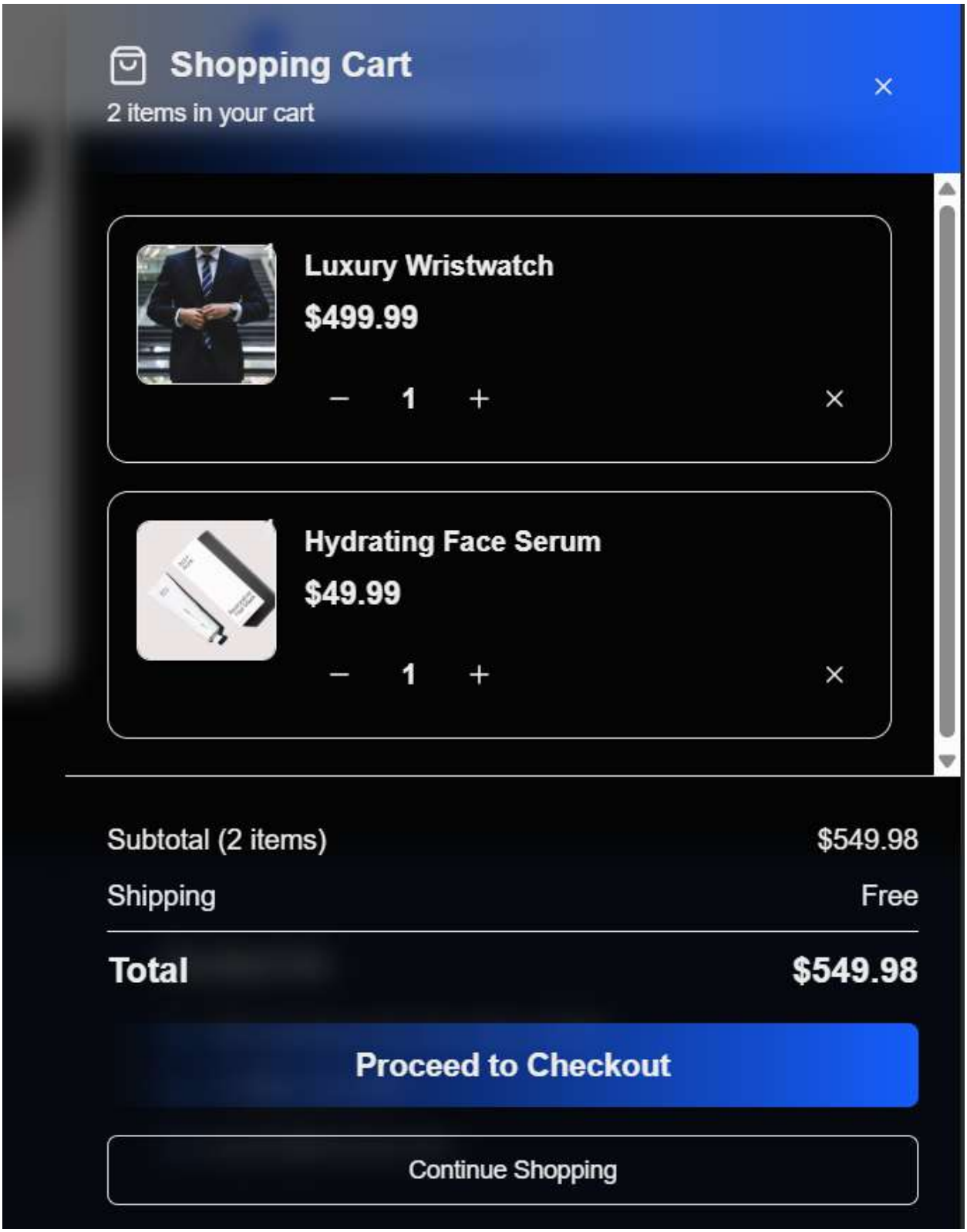

Fig: Card Page

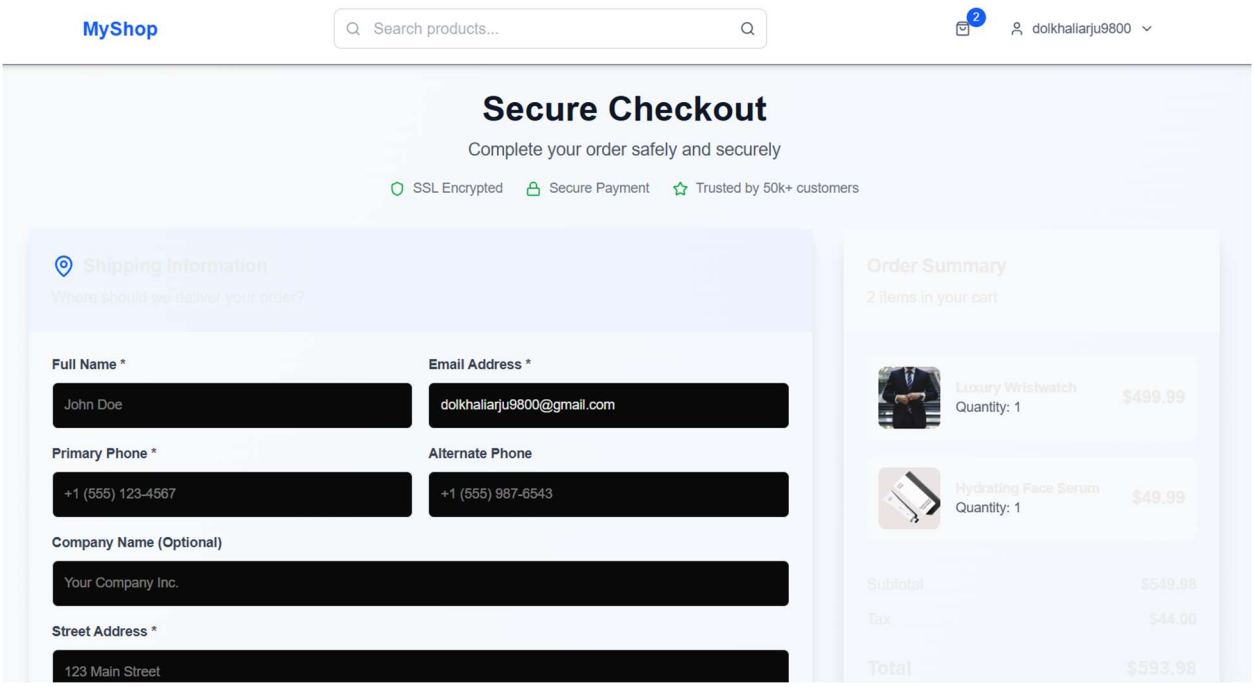

Fig: Checkout Page