

E-commerce Website Development Report

Kevin Zhang,Adel Leshob,Hugo Dos Res



June 1, 2025

Seoul National University of Science and Technology (SeoulTech)

232 Gongneung-ro, Nowon-gu, Seoul, South Korea

| **Section** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **Page** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1. Project Overview** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **2** |
| **1.1 Context and Objectives** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **2** |
| **1.2 Scope and Expected Outcomes** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **2** |
| **2. Architecture and Design** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **3-5** |
| **2.1 Architecture Overview** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **3-4** |
| **2.2 Design** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **4-5** |
| **3. Key Features and Technologies** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **6-7** |
| **3.1 Main Functionalities** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **6** |
| **3.2 Technologies and Tools** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **7** |
| **4. Challenges and Solutions** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **8** |
| **4.1 Problems Encountered** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **8** |
| **4.2 Solutions Implemented** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **8** |
| **5. Individual Contributions** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **9-11** |
| **5.1 Team Member 1: Kevin Zhang** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **9** |
| **5.2 Team Member 2: Adel leshob** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **10** |
| **5.3 Team Member 3 Hugo Dos Res** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **11** |
| **6. Conclusion and Future Perspectives** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **12** |

**1. Project Overview**

**1.1 Context and Objectives**

This project involved designing and developing a fully functional e-commerce website as a group assignment. The main goal was to create a secure and user-friendly online shopping platform using Node.js and the Express framework, with SQLite as the database for persistent data storage. The core objective was to apply and integrate all the web development concepts learned throughout the course into a single, practical project, including modular code design, secure user authentication, and session management.

**1.2 Scope and Expected Outcomes**

The application aims to replicate core e-commerce functionalities expected in a real-world site. By the project’s completion, the team delivered a working prototype enabling users to:

* Register and log in securely
* Browse and search products
* Add, update, and remove items in a shopping cart
* Complete a simulated checkout process
* Persistently manage user and product data

This project also provided valuable experience in collaborative development, integrating front-end and back-end technologies, and implementing security measures in web applications.

**2. Architecture and Design**

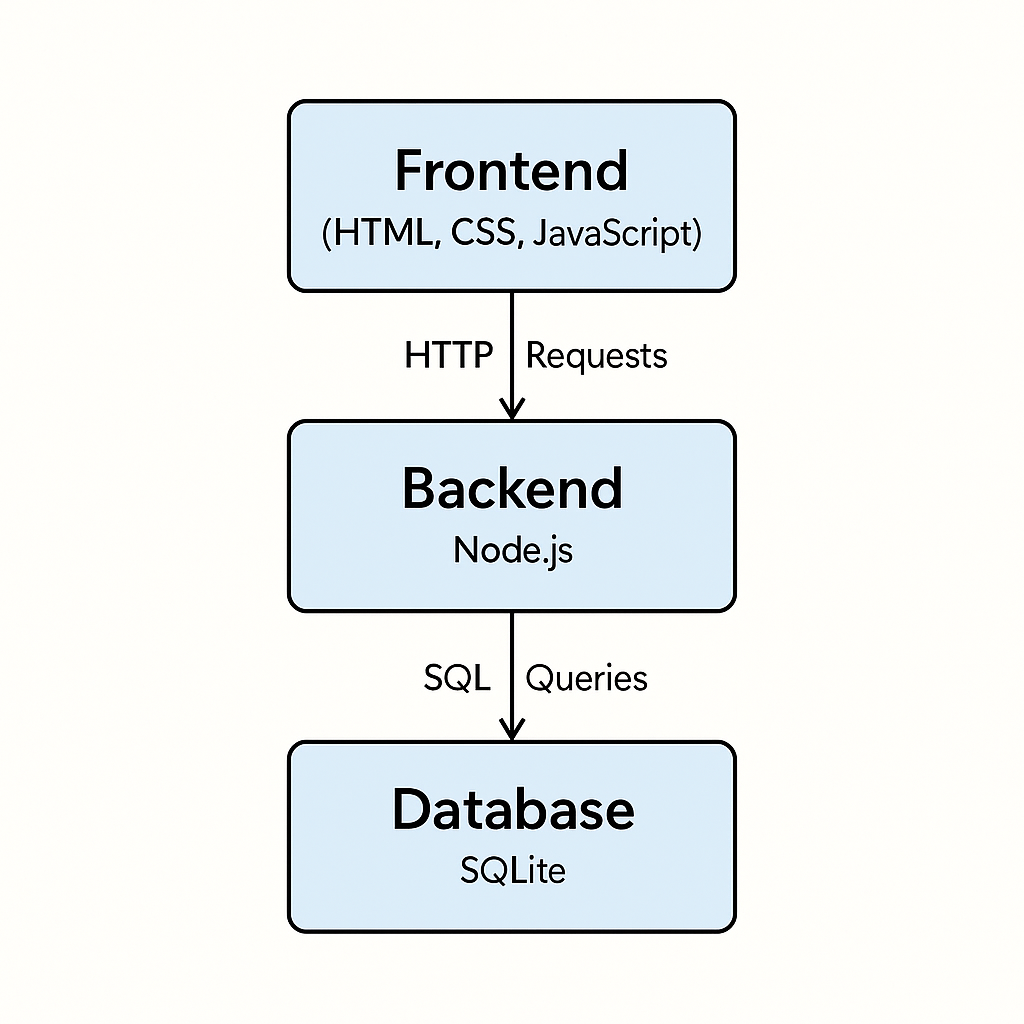
**2.1 Architecture Overview**

The e-commerce project is structured following a classic client-server architecture comprising:

* **Frontend (Client):** User interface built with HTML, CSS, and JavaScript, enabling browsing, searching, cart management, and user authentication.
* **Backend (Server):** Server-side application responsible for business logic, user management, product handling, order processing, and database communication.
* **Database:** Relational storage (sqlLite) for product, user, order, and session data.
* **External Services:** Integration of a mock payment gateway to securely handle the checkout process.

This layered architecture ensures clear separation of concerns, improves maintainability, and facilitates future scalability.

**Image: System Architecture**

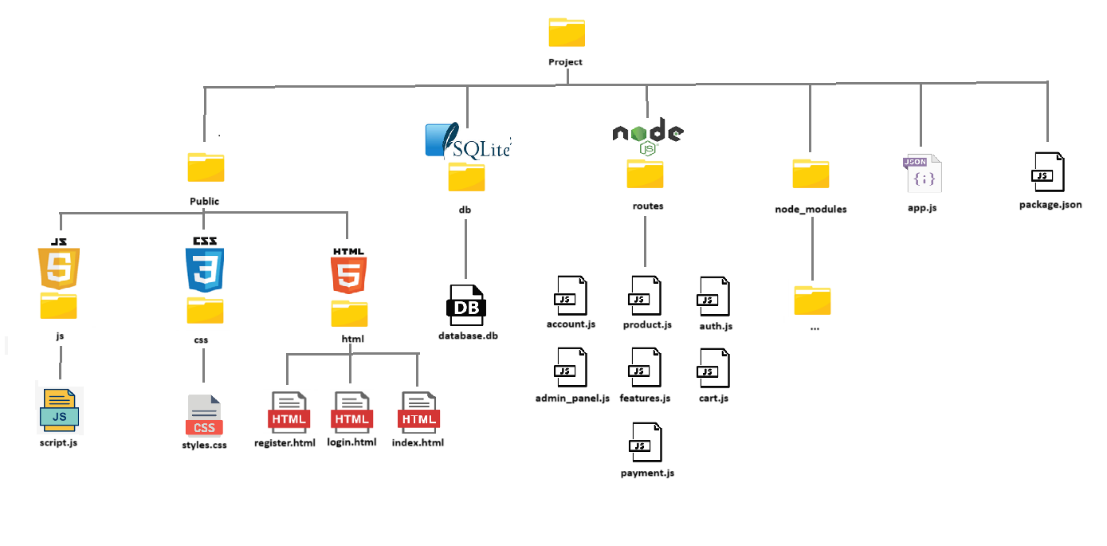


This diagram shows the main parts of the e-commerce app and how they work together. The front end talks to the back end, which handles user actions and connects to the database to store data.

**The Importance of Folder Structure**

A well-structured folder system is essential for keeping your project organized, readable, and easy to maintain. It helps teams collaborate efficiently, simplifies debugging, and ensures scalability as the project grows.

**Image: Files Architecture**

****

This structure shows how files are organized in our e-commerce website project. It separates code, assets, and configuration to keep the project clean and easy to manage.

**2.2 Design**

**About the E-commerce Website**

The e-commerce website features a clean and modern design with a homepage showcasing featured products and promotions. Users can browse a categorized product catalog, manage their cart, and complete purchases through a secure checkout. An admin panel allows product and order management, making the platform functional for both customers and administrators.

**Image: Website Overview**

**3. Key Features and Technologies**

**3.1 Main Functionalities**

This section highlights the main features of the e-commerce application and the key technologies used to build it. The focus is on usability, security, and performance.

**The main features of the application are:**

* **Homepage**: A landing page showcasing featured products, promotions, and a clear navigation menu.
* **Product Catalog**: Displays products with images, descriptions, and prices. Includes search and filtering options, and product categorization.
* **Shopping Cart**: Allows users to add, remove, and update items in their cart. Automatically calculates the total cost including taxes and shipping.
* **User Authentication**: Supports user registration, login, and profile management.
* **Checkout Process**: Provides a secure checkout process integrated with a mock payment gateway.
* **Admin Panel (Optional)**: Enables management of products, user accounts, and orders.

**Additional development goals included:**

* Writing clean, modular, and maintainable code.
* Implementing secure user authentication with password hashing using bcrypt.
* Managing user and product data through the SQLite database.
* Creating a responsive user interface for enhanced user experience.

**3.2 Technologies and Tools**

For the development of the e-commerce website, the following technologies and tools were used to ensure an efficient application:

* **JavaScript**: The core scripting language used to implement dynamic behavior and interactivity on the client side.
* **Node.js**: A JavaScript runtime environment used to build the server-side of the application, enabling fast and scalable back-end development.
* **Express**: A lightweight and flexible web framework for Node.js, used to simplify the creation of server-side logic, handle routing, and manage HTTP requests and responses efficiently.
* **HTML & CSS**: The foundational technologies for structuring and styling the web pages, ensuring a responsive and visually appealing user interface.
* **SQLite**: A lightweight, file-based relational database management system used for storing product data, user information, and transaction records.
* **GitHub**: The version control platform used to manage the project's source code, track changes, and facilitate collaboration through repositories.

These technologies were selected for their practicality and reliability, enabling efficient development and solid performance.

**4. Challenges and Solutions**

|  |
| --- |
| **4.1 Problems Encountered** |
| Understanding the Express.js  Library Express is a powerful and widely used library, but we found it difficult to understand at first due to its size and the number of features it offers. It took time to figure out how to use it correctly and efficiently.  Using SQLite and SQL Queries  We faced some challenges while trying to make the backend work properly with SQLite. Writing correct SQL queries and integrating them into the server logic was not always straightforward.  Implementing CSS for a Good Design  Styling the website to make it both attractive and interactive was not easy. Making the visual design look clean and appealing required multiple adjustments and testing.  **4.2 Solutions Implemented** |

Learning Express Step by Step

We took the time to learn Express gradually, by testing small parts of the code and reading documentation or examples. This helped us understand how to use it in our project.

Writing and Testing SQL Queries

We improved our SQL usage by testing queries and making sure they returned the expected results. We also ensured that SQLite was correctly connected to the server.

Improving CSS Styling

We made several improvements to the CSS to enhance the website’s appearance. We adjusted the layout, chose better colors, and added some interactive effects to make the site more dynamic.

|  |
| --- |
| **5. Individual Contributions** |
| **5.1 Team Member 1: Kevin Zhang** |

**5.2 Team Member 2: Adel Leshob**

|  |
| --- |
| **5.3 Team Member 3: Hugo Dos Reis** |

|  |
| --- |
|  |

|  |
| --- |
| **6. Conclusion and Future Perspectives** |

This e-commerce project was a good opportunity to practice full-stack web development using tools like Node.js, Express, JavaScript, SQLite, HTML, and CSS. The website includes important features such as a product catalog, shopping cart, user login, and a basic checkout process.

We built a simple and functional database, and the code works as intended, even if it could be improved for readability and organization. The design is basic but usable, and some adjustments may be needed for better responsiveness on all screen sizes.

Main features like product categories, shopping cart updates, and user accounts were implemented successfully. The optional admin panel was started and could be improved later. A test payment system was used, but the code is ready to connect to a real one in the future.

This project can be improved by adding stronger security, a better design, and more advanced features if needed.

Future improvements may include:

* Strengthening security by adding password hashing and role-based access control.
* Migrating to a more robust database solution for better scalability.
* Adding real-time features such as order tracking or live chat support.

Overall, this project serves as a strong foundation for building a professional e-commerce platform and demonstrates the practical application of modern web development practices.