Portfolio Part 3: ELOS Ecommerce Web Application

Overview

This project is focused on developing a functional and responsive eCommerce web application using Laravel 12 for the backend and Vue.js (with Inertia.js) for the frontend. The platform allows users to browse products, add them to a cart, and proceed to checkout using Stripe payment integration. MySQL was used as the primary database to store product, user, and order information.

Technology Stack & Approach

To begin, I conducted research on the official Laravel 12 documentation, Vue.js fundamentals, and how Inertia.js bridges Laravel with Vue without needing a full API. This simplified the process of connecting server-side logic with client-side views and enabled smooth, SPA-like navigation.

The backend was built with Laravel 12, leveraging its routing, controller, and authentication features. The frontend was handled with Vue.js, using Inertia.js for routing and reactivity. The database layer was managed using MySQL, where I defined schema migrations and relationships to manage product listings and user data.

Stripe was integrated for secure and modern payment processing. Admin users could add and manage products from a dashboard, while customers had a clean and intuitive interface for shopping.

What I Learned

- How to set up and structure a Laravel + Vue + Inertia project from scratch.
- Building CRUD functionality for product management.
- Handling frontend interactions using Vue's reactivity.
- Setting up secure payment flows using Stripe.
- Managing relational databases using Laravel Eloquent and MySQL.

I also improved my understanding of full-stack development principles and how frontend and backend technologies can work together efficiently.

Challenges & What Could Be Improved

One challenge was debugging errors between Laravel and Vue during data passing via Inertia. I realized the importance of consistent naming conventions and clear data structure. Another issue was relying too much on manual testing. I did not implement automated tests, which could have saved time and caught edge cases earlier.

In terms of design, although the app was functional, it lacked advanced UI polish and accessibility features. Error handling in some parts of the app (like failed payments or invalid input) also needs improvement.

Future Improvements

In future, I would:

- Implement automated testing (unit and feature tests) to improve reliability.
- Improve mobile responsiveness and UI consistency.
- Add user accounts and order tracking features.
- Include filtering and search for better product discovery.
- Use email notifications for order confirmations and updates.
- Strengthen error handling and validation for all inputs.

Conclusion

This project gave me practical experience in full-stack web development using modern tools. I now feel more confident working with Laravel and Vue in a real-world context. It was also a great introduction to handling real eCommerce workflows and integrating third-party services like Stripe. I plan to build on this project and improve it further based on the lessons learned.