E-Commerce Web Application with Admin Seller Functionality

Malik Stewart
Applied Computing: Business Concentration
December 2024

Statement of Purpose

- This project aims to develop a versatile e-commerce platform that is initially tailored to meet the needs of small clothing brand owners, providing them with a credible space to showcase and sell their products. For the current phase, the platform supports a single admin user as the sole seller, but it is designed to scale and accommodate future enhancements such as the ability for users to create their own shops and sell products and serve different niche groups.

Problem Statement

- Small businesses, particularly clothing brand owners, often struggle to set up professional e-commerce websites due to high costs and technical barriers.
- This project solves that problem by providing an easy-to-use platform where sellers can display and sell their products while ensuring a seamless shopping experience for customers.

Research & Background

- E-commerce platforms are essential tools for businesses, especially small retailers. This project focuses on creating an e-commerce solution for small clothing brand owners, but ensures that the system is adaptable to any niche market.
- The platform uses Ruby on Rails, Stripe API, and Tailwind CSS for backend development, secure payment processing, and modern, responsive frontend design.

Project Requirements

- - Product Catalog: Browse, view details, and add products to the cart.
- - Shopping Cart System: Add/remove items, view total price, proceed to checkout.
- - Payment Integration: Secure payment via Stripe API.
- - Admin Interface: Manage products, orders, and inventory.
- - Scalability: Future multi-seller functionality.

Project Implementation & Technical Details

- - Product Catalog: Uses ActiveRecord models to retrieve product data from PostgreSQL.
- - Shopping Cart System: Uses Rails sessions for storing cart items.
- - Payment Integration: Stripe Checkout API for secure payments.
- - Admin Interface: Rails Admin simplifies product and order management.

Test Plan & Results

- - Unit Testing: Rails Console used to test models like Product, Cart, and Order.
- - Integration Testing: Capybara for simulating user behavior.
- - UI/UX Testing: Ensures responsiveness on mobile and desktop devices.
- - Security Testing: Focused on secure user authentication and payment processing.

Challenges Overcome

- - Design Flexibility: Ensuring platform adaptability for any seller.
- - Stripe Integration: Ensuring secure and seamless payment processing.
- - Responsive Design: Achieving full responsiveness using Tailwind CSS.

Future Enhancements

- - Multi-Seller Functionality: Enabling users to create their own shops.
- - Mobile App: Developing a mobile app for enhanced user experience.
- - Analytics Dashboard: Providing admins with insights into sales and customer behavior.

Conclusion

- Successfully created a scalable e-commerce platform for small clothing brand owners.
- Platform leverages Ruby on Rails, Stripe API, and Tailwind CSS for security, payment processing, and responsiveness.
- Future enhancements will include:
  - Multi-seller functionality
  - Mobile app integration
- Overcame challenges related to design flexibility and secure payment integration.
- Platform is ready for future growth and expansion.