# Software Requirements Specification (SRS)

CamTrade Hub — Multivendor E-Commerce and Logistics Platform

## Technology Stack

- Frontend: Next.js (React, TailwindCSS)  
- Backend / Database: Supabase (PostgreSQL, Auth, Storage, Realtime)  
- Hosting / Deployment: Vercel (frontend + API routes)  
- Integration APIs: MTN MoMo / Orange Money (payment), EmailJS or Resend (notifications)

## 1. Introduction

The project aims to build a web-based multivendor e-commerce platform connecting customers, vendors, and logistics providers between China and Cameroon. It automates the process of ordering, tracking, and managing shipments while providing local entrepreneurs the tools to operate online stores.

## 2. System Overview

The platform consists of four main user roles:  
- Customer: Browses and purchases products.  
- Vendor: Sells and manages store inventory.  
- Logistics Provider: Updates shipment and delivery details.  
- Administrator: Manages the overall system.

## 3. Functional Requirements

Customer Module  
- Register and log in via Supabase Auth.  
- Browse, search, and purchase products.  
- Add items to cart and checkout.  
- Track orders in real time.  
- Manage order history and profile.  
  
Vendor Module  
- Register as a verified vendor.  
- Create and manage a personal storefront.  
- Manage product listings and stock.  
- View and manage customer orders.  
- Access sales and revenue dashboard.  
  
Logistics Module  
- Register and list shipping services.  
- Update shipment status (processing, in transit, delivered).  
- Manage assigned deliveries.  
- Trigger notifications for order updates.  
  
Admin Module  
- Manage all user roles and permissions.  
- Approve or deactivate vendor stores.  
- Monitor platform analytics and transactions.  
- Handle disputes and refunds.

## 4. Non-Functional Requirements

- Performance: Page loads < 3 seconds, API response < 500ms.  
- Security: HTTPS, Supabase Row-Level Security (RLS).  
- Scalability: Support 5,000+ concurrent users.  
- Maintainability: Modular Next.js + Supabase architecture.  
- Usability: Responsive and mobile-friendly.  
- Reliability: Real-time data synchronization.

## 5. Database Schema (Supabase)

users(id, name, email, role, created\_at)  
products(id, vendor\_id, name, price, category, image\_url)  
orders(id, user\_id, product\_id, logistics\_id, status, total\_price)  
stores(id, vendor\_id, store\_name, description)  
logistics(id, company\_name, contact, region)  
shipments(id, order\_id, logistics\_id, current\_status, updated\_at)  
payments(id, order\_id, payment\_method, payment\_status, timestamp)

## 6. System Architecture

Frontend: Next.js with server-side rendering for SEO and performance.  
Backend: Supabase for database, authentication, and real-time updates.  
Hosting: Vercel with continuous deployment and SSL/CDN support.

## 7. Integration Plan

- MTN MoMo / Orange Money for payment.  
- EmailJS / Resend for notifications.  
- Google Maps API (optional) for shipment tracking visualization.

## 8. Testing & Deployment

- Unit Testing: Jest and React Testing Library.  
- Integration Testing: Postman / Thunder Client.  
- Deployment: GitHub → Vercel auto-deploy.  
- Monitoring: Vercel Analytics and Supabase Logs.

## 9. Expected Outcomes

- 50% fewer customer inquiries via real-time tracking.  
- 30% faster order processing.  
- 100+ vendors and 10+ logistics partners onboarded.  
- Improved logistics visibility and digital business growth in Cameroon.