



E-Commerce Project Specification Document

💡 Project Overview

This project aims to build a multi-role E-commerce platform where Admins, Sellers, and Customers interact within a unified system. The platform includes authentication, product management, shopping cart, checkout, discounts, payments, orders, dashboards, and notifications.

The goal is to provide interns with real-world teamwork experience across frontend and backend development, covering React.js + Python (FastAPI/Flask), database integration, and deployment.

👥 Team Composition

- Frontend (React.js Developers – 3)
- Backend (Python FastAPI/Flask Developers – 2)

🔑 Core Modules & Features

Authentication & User Management

- Login, Signup with JWT Auth.
- Roles: Admin, Seller, Customer.
- Profile management (profile picture, details, addresses).
- Admin can block/unblock users.

Product Management

- Sellers can add, update, delete products.
- Bulk upload products via CSV/Excel.
- Customers can browse/search/filter products.
- Admin can edit/delete/approve products.

Cart & Checkout

- Customers can add/remove/update cart items.

- Checkout flow with order summary.
- Coupons/Discounts applied at checkout.

Coupons & Discounts

- Admin/Seller can create discount coupons.
- Percentage or flat discounts.
- Usage conditions: minimum cart value, expiry date, usage limit.
- Customers can apply coupons at checkout.

Payments

- Integration with Stripe (test mode) or Razorpay.
- Payment success/failure handling.
- Refund APIs (stretch goal).

Orders & History

- Customers: Place orders, view history, reorder, download invoices.
- Sellers: View/manage orders related to their products, update status.
- Admin: View/manage all orders, refund management (optional).

Dashboards & Analytics

- Admin: Revenue, sales reports, top-selling products, user growth.
- Seller: Sales performance, order breakdown.

Notifications

- Email notifications for order confirmations, status updates, coupon expiry.
- Seller notifications for new orders.

Wishlist & Reviews (Stretch)

- Customers: Add to Wishlist, post reviews & ratings.
- Sellers: Respond to reviews.
- Admin: Moderate/delete reviews.

LLM-Powered Review Summaries & Q/A

Customers: Can ask QA and reviews summaries for product.

Role-Based Access Control (RBAC)

Module	Admin	Seller	Customer
Authentication	Manage/block users	Register/Login	Register/Login
Profile Management	View/Edit all	Manage store profile	Manage profile, addresses
Product Management	Approve/Edit/Delete all	CRUD own products, bulk upload	Browse, search, filter
Cart & Checkout			Full cart & checkout flow
Coupons & Discounts	Create/manage global	Create/manage own	Apply at checkout
Orders & History	View/manage all	View/manage own	View history, reorder, invoice
Payments	Refunds (stretch)	Track order payments	Make payments
Dashboard & Analytics	Global sales & revenue	Own sales & orders	
Notifications	new sellers	New order alerts	Order updates, coupon expiry
Wishlist & Reviews	Manage/delete reviews	Respond to reviews	Add wishlist, post reviews

Sprint Plan (5 Weeks)

(LLM-Powered Review Summaries & Q/A needs to be implemented parallelly in each sprint)

Sprint 1: Setup & Auth (Week 1)

- Frontend: Setup project, routing, login/signup pages, role-based navigation.
- Backend: Setup project, JWT Auth(Cookie based), user schema.

Sprint 2: Products + Bulk Upload (Week 2)

- Frontend: Product listing, details, Seller CRUD UI, CSV upload UI.
- Backend:
 - Scrape existing e-commerce website for gathering data (Selenium, Beautiful soup,)
 - Product APIs, categories, bulk CSV parser.

Sprint 3: Cart & Orders (Week 3)

- Frontend: Cart page, checkout page, order history UI.
- Backend: Cart APIs, order schema, order history API.
- Extra: Reorder + invoice download.

Sprint 4: Coupons & Payments (Week 4)

- Frontend: Coupon UI at checkout, Stripe integration.
- Backend: Coupon APIs, validation, Stripe webhook.

Sprint 5: Dashboards & Extras (Week 5)

- Frontend: Admin/Seller dashboards with charts, Customer insights.
- Backend: Analytics APIs, notification APIs.
- Stretch: Wishlist & Reviews.

Tech Stack

- Frontend: React.js (Vite, Tailwind, Redux Toolkit, React Router)
- Backend: FastAPI/Flask, SQLAlchemy, JWT
- Database: PostgreSQL (Render/Heroku/Railway)
- Payments: Stripe / Razorpay (test)
- File Uploads: Cloudinary / AWS S3
- Deployment:
 - Frontend → Vercel
 - Backend → Render/Railway

Final Deliverables

- Live frontend deployed on Vercel.
- Live backend APIs deployed on Render/Railway.
- Fully working multi-role E-commerce platform.
- Swagger/OpenAPI docs for all APIs.
- Database schema documentation.
- GitHub repos with clear commit history & PR reviews.