SAREE4EVER — COMPLETE PROJECT SPECIFICATION (PLAIN TEXT DOCUMENT)

(Everything we discussed, consolidated in one place)

1.

Project Overview

saree4ever is a modern e-commerce platform focused on traditional and designer Indian sarees.

The website uses a **black-and-white editorial theme** for text and layout, similar to New York Times style, while **keeping product images in full color** to highlight saree beauty.

The platform includes:

- A full customer-facing storefront
- A powerful admin dashboard
- A product management system
- Inventory control
- Orders, payments, and tracking
- Variant management (color, blouse option, etc.)

- AI-powered search & chatbot
- CSV import tools
- Real-time updates
- User accounts with Amazon-style order tracking

2.

Technology Stack

Frontend:

- **Next.js** (Pages or App Router)
- **TailwindCSS** (black-white editorial theme)
- **React Server Components** where beneficial
- **Vercel** for deployment

Backend:

- **Express.js** running on **Render**
- **Prisma Client** for DB access
- **Supabase Postgres** as database
- **Supabase Storage** for product images
- **JWT-based Admin Auth**

Payments & AI:

- **Razorpay** for payment
- **OpenAI or your preferred LLM** + embeddings for AI chatbot

Real-time features:

- **SSE** or **Pusher/Ably** (optional)

3.

Major Features

Customer Frontend

- Landing page with hero banner
- Collections page
- Saree Types page
- Offers page
- Product listing pages
- Product detail page
- Variant selection (color + blouse)
- Cart & Checkout
- Razorpay payments
- Order status tracking (Amazon-style timeline)
- User profile with order history

- Search, filters, sorting

Admin Dashboard

- Create, edit, delete products
- Create variants (color, blouse/no blouse)
- Manage collections, types, categories
- Upload product images
- Inventory management
- Order management (update status, mark shipped, add tracking)
- CSV bulk import/export
- AI assistant to analyze stock, trends, demand

4.

Product Data Model

A. Product

This is the *parent* item.

Fields include:

- ID
- Title
- Description

- Collections (array)
- Categories (array)
- Types (array)
- Attributes (like weave, fabric, etc.)
- Created/updated timestamps

B. Product Variant

Each sellable variant of a saree:

- ID
- Product ID
- SKU
- Color
- With Blouse / Without Blouse
- Price + MRP
- Stock
- Reserved stock
- Images array
- Active/Inactive
- Created/updated timestamps

C. Inventory Transactions

Each stock change:

- Variant ID

- Quantity changed
- Reason
- Order reference
- Timestamp

5.

Taxonomy (Categorization System)

You will have 3 independent classification layers.

A product can belong to **multiple** items in each layer.

5.1 Collections (Marketing Groups)

Examples:

✔ Festive Collection

✔ Bridal Edit

✔ Silk Special

✔ Handloom Premium

✔ Officewear Essentials

✔ New Arrivals

5.2 Categories (Usage Intent)

Examples:

✔ Bridal

✔ Partywear

✔ Festival

✔ Everyday

✔ Office

✔ Designer

5.3 Types (Fabric / Weave)

Examples:

✔ Kanjivaram

✔ Banarasi

✔ Chiffon

✔ Georgette

✔ Paithani

✔ Linen

✔ Cotton Silk

✔ Organza

✔ Tussar

6.

Inventory Management (How It Works)

Inventory must be safe and accurate.

Step 1 — Available Stock

Total stock - Reserved stock = What customer can buy.

Step 2 — Reservation

When user starts checkout:

- Reserved stock increases
- Prevents overselling

Step 3 — Payment Success

Backend commits reservation:

- Decrease actual stock
- Decrease reserved stock
- Log inventory transaction

Step 4 — Payment Failure or Cancel

Reserved stock is released.

7.

Order & Shipment Workflow

1. User places order
2. Stock is reserved
3. User pays via Razorpay
4. Razorpay webhook confirms payment
5. Stock is finalized
6. Order status becomes:

    - Pending
    - Paid
    - Processing
    - Shipped
    - Delivered
7. Admin adds tracking number (Delhivery, BlueDart, Shiprocket etc.)
8. Customer can track timeline in profile

8.

Razorpay Payment Workflow (Simplified)

1. Frontend → Backend: “Create order”
2. Backend reserves stock, creates internal order
3. Backend → Razorpay: “Create payment order”
4. Razorpay → Frontend: returns order_id
5. Customer completes payment in Razorpay UI
6. Razorpay → Server webhook: “Payment success”
7. Server commits stock and marks order paid
8. Customer sees updated order status

9.

Frontend Structure

Pages required:

Public Pages

- **Home**
- **Collections**
- **Collections/[slug]**
- **Types**
- **Types/[type]**
- **Offers**
- **Product/[id]**
- **Search**
- **Cart**
- **Checkout**
- **Order/[id]**
- **Login / Signup**
- **User Profile**

Admin Pages

- Admin Login
- Dashboard Home

- Products (Add/Edit/Delete)
- Variants Manager
- Collections Manager
- Offers Manager
- Orders Manager
- Shipments Manager
- CSV Import
- AI Insights Panel

10.

AI Chatbot for saree4ever

The AI bot will:

- Analyze your database
- Understand stock & offers
- Recommend sarees
- Answer customer questions
- Help admin with insights

The system uses:

- Product embeddings (pgvector or Pinecone)
- OpenAI or compatible model

- A secure backend endpoint for answering queries

11.

Deployment Setup

Frontend:

- Deploy on **Vercel**
- Use environment variables for API URL & Supabase keys

Backend:

- Deploy on **Render**
- Use environment variables:
    - DATABASE_URL
    - SUPABASE_SERVICE_ROLE_KEY
    - RAZORPAY_KEY_ID
    - RAZORPAY_KEY_SECRET
    - JWT_SECRET

Database:

- Supabase hosted Postgres
- Daily backups enabled

Storage:

- Supabase Storage bucket: product-media

12.

Development Flow (Step-by-step Summary)

1. Project structure (frontend + backend folders)
2. Supabase setup (DB, auth, buckets)
3. Admin-auth protected backend APIs
4. Product CRUD
5. Variant CRUD
6. Inventory logic
7. Order creation
8. Razorpay payment integration
9. Shipment/tracking module
10. CSV bulk import
11. AI chatbot integration
12. Frontend components (cards, lists, filters)
13. User account & order tracking
14. Admin dashboard front-end
15. Global testing

16. Deployment

13.

Core Principles

- Editorial black & white theme
- Image-forward saree display
- Real-time accurate stock
- Clean and simple admin workflow
- Highly scalable architecture
- AI-powered shopping
- Amazon-style tracking
- Fast page loads (Next.js + Vercel)
- Secure OTP/email login (Supabase Auth)

14.

End Result

After building all steps, saree4ever becomes:

✔ A beautiful online saree store

✔ With high-performance search & filtering

✔ Multiple taxonomy layers (types, collections, categories)

✔ Safe inventory management

✔ Razorpay-powered checkout

✔ Order tracking like Amazon/Flipkart

✔ Strong admin tools

✔ AI assistant for customers and your back-office

This is the **complete blueprint** for the entire project.
