Contents

[Step 1: Define the Scope (Planning Phase) 1](#_Toc198385314)

[Project Overview 1](#_Toc198385315)

[Core Features 1](#_Toc198385316)

[Admin Features 2](#_Toc198385317)

[Design Responsiveness 2](#_Toc198385318)

[User Roles 👥 2](#_Toc198385319)

[🔐 Optional Future Features 2](#_Toc198385320)

[📄Deliverable: Basic Requirements Document (Summary) 2](#_Toc198385321)

# Step 1: Define the Scope (Planning Phase)

## Project Overview

You are building an eCommerce web application dedicated to **wedding attire** (dresses for men and women). The site will allow customers to browse products, manage wishlists, add to cart, checkout, and for admins to manage the catalog.

## Core Features

**🔐 User Management**

* **User registration/login (email, mobile, social login optional)**
* **User profile page (edit name, address, etc.)**
* **Forgot/reset password**

**🛒 Shopping Features**

* **Product Catalog:**
  + **Categories: Men, Women**
  + **Filters: Size, Price range, Style, Color**
  + **Sorting: Price (low to high), Popularity, Latest**
* **Product Details Page:**
  + **Description, Sizes available, Price, Images**
  + **Add to Wishlist**
  + **Add to Cart**

**❤️ Wishlist**

* **Add/remove items from wishlist**
* **Move wishlist item to cart**

**🛍️ Cart & Checkout**

* **Add/remove/update quantity in cart**
* **Cart page with total calculation**
* **Checkout page with:**
  + **Delivery address**
  + **Summary**
  + **Payment (initially mock or cash-on-delivery, can extend to Stripe/Razorpay later)**

**📦 Orders**

* **Order confirmation**
* **My Orders page (view past orders, status)**

## Admin Features

* Admin login
* Dashboard with analytics (total orders, users, best-selling items)
* Product management (CRUD)
  + Upload images
  + Set price, sizes, stock, etc.
* View and manage orders
* User management (view/block users optionally)

## Design Responsiveness

* ✅ Yes, the site should be **responsive** (mobile + tablet + desktop)
* Mobile-first design recommended

## User Roles 👥

| **Role** | **Description** |
| --- | --- |
| **Customer** | Can register, browse products, manage cart/wishlist, place orders |
| **Admin** | Can add/edit/delete products, manage orders, manage users |

## 🔐 Optional Future Features

* OTP login or Google sign-in
* Reviews and ratings on products
* Product recommendations
* Inventory management
* Multiple delivery addresses
* Coupons/discounts

## 📄Deliverable: Basic Requirements Document (Summary)

**Project: Wedding Dress eCommerce Site**

| **Section** | **Details** |
| --- | --- |
| **Target Users** | General public looking to buy wedding attire |
| **User Roles** | Customer, Admin |
| **Key Features** | Login, Wishlist, Cart, Checkout, Order History, Product Filters |
| **Admin Tools** | Product management, Order view, User list |
| **Design** | Fully responsive for mobile/desktop |
| **Tech Stack (next step)** | React, Spring Boot, MySQL/Postgres |

# Step 2: Database Design (ERD & Tables)

We'll use a **relational database** (MySQL/PostgreSQL — both are fine) and design normalized tables to handle products, users, carts, orders, etc.

## 🎯 **Key Entities**

| **Entity** | **Description** |
| --- | --- |
| users | Customers, admins (and sellers if needed) |
| products | Wedding dresses, with attributes |
| category | Men, Women, Bridal, Accessories etc. |
| wishlist | Products saved by a user |
| cart | What a user added to their cart |
| orders | Customer orders |
| order\_items | Items per order |
| product\_images | Multiple images per product |
| reviews | Customer reviews on products |
| address | Shipping address linked to users |
| payment | User can make payment using UPI/debit/credit |

## 🗃️ Database Tables

# Step 3 Backend API’s

**🧑 User APIs**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /api/users/register | Register a new user |
| POST | /api/users/login | Authenticate user |
| GET | /api/users/profile | Get current user profile |
| PUT | /api/users/profile | Update user profile |

**📍 Address APIs**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/addresses | Get all addresses for user |
| POST | /api/addresses | Add a new address |
| PUT | /api/addresses/:id | Update an existing address |
| DELETE | /api/addresses/:id | Delete an address |

**👗 Product & Category APIs**

| **Method** | | | **Endpoint** | **Description** |
| --- | --- | --- | --- | --- |
| GET | | | /api/products | List all products |
| GET | | | /api/products/:id | Get single product details |
| POST | | | /api/products | Add new product *(admin)* |
| PUT | | | /api/products/:id | Update product *(admin)* |
| DELETE | | | /api/products/:id | Delete product *(admin)* |
| GET | | | /api/categories | Get all categories |
|  |  |

**🛒 Cart APIs**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/cart | Get current user’s cart |
| POST | /api/cart | Add item to cart |
| PUT | /api/cart/:item\_id | Update quantity in cart |
| DELETE | /api/cart/:item\_id | Remove item from cart |

**❤️ Wishlist APIs**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/wishlist | Get wishlist items |
| POST | /api/wishlist | Add item to wishlist |
| DELETE | /api/wishlist/:product\_id | Remove item from wishlist |

**📦 Order APIs**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/orders | List user’s orders |
| POST | /api/orders | Place a new order |
| GET | /api/orders/:id | Get order details |

**🔍 Tracking APIs**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/orders/:id/tracking | Get tracking details for an order |

**💳 Payment APIs**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /api/payments | Process a payment for an order |
| GET | /api/payments/:order\_id | Get payment status for an order |