



ApparelDesk – Store, Orders & Bills

1. Overview:

- Build a web-based clothing e-commerce system where customers can browse products (shirts, pants, t-shirts, kurtas, etc.), add items to a cart, apply coupon codes during checkout, make payments, and later view/download their sale orders and invoices.
- In the backend, users manage products, contacts/users, sale orders, purchase orders, customer invoices, vendor bills, payments, payment terms and discount offers. Product stock should update automatically based on purchase and sales transactions, and products can be published or unpublished on the storefront.
- The system should connect all these records end-to-end so that:
 - Customer orders and payments on the frontend create corresponding sale orders, invoices and payments in the backend.
 - Backend users can configure payment terms (including early payment discount or installment options) and discount offers (coupon codes with fixed percentage discounts).
 - Backend users can generate sales and purchase reports by product, vendor and customer.

2. Primary Actors

1. Internal Users (Backend)

- Manage products (create, edit, stock, publish/unpublish).
- Manage contacts/users.
- Create and manage sale orders, purchase orders, customer invoices, vendor bills.
- Record payments, set up payment terms and discount offers (coupon codes).
- View sales and purchase reports.

2. Portal Users (Customers)

- Browse clothing products and use filters.
- Add items to cart, apply coupon codes, and make payments.
- View their own sale orders and invoices, and download invoices from the portal.

3. Master Data Modules

1. Users

- a. Stores all system users (internal and portal).
- b. **Fields:** name, role (internal/portal), email, password, mobile, address(city, state,pincode)
- c. For every User, a linked Contact record must exist.
- d. When a user signs up from the e-commerce website (portal user), a Contact is created automatically with type = Customer.

2. Contacts

- a. Stores business contacts used in sales and purchase documents.
- b. **Fields:** name, type (customer/vendor/both), email, mobile, address(city, state,pincode)
- c. Contacts are used on Sale Orders, Purchase Orders, Customer Invoices, Vendor Bills and Payments.
- d. Every User must have a corresponding Contact record.
- e. Portal signups create a Contact with Type = Customer.

3. Products

- a. Stores all clothing items available for sale or purchase
- b. **Fields:** product_name, product_category (men/women/children/...), product_type (shirt/pant/kurtas/...), material (cotton/nylon/...), colors (multi), current_stock, sales_price, sales_tax, purchase_price, purchase_tax, published, images (multi)
- c. Current Stock is updated based on purchase and sales transactions.
- d. Only products with Published = true are visible to customers on the website.

4. Payment Terms

- a. Defines payment conditions and optional early payment discount.
- b. **Fields:**name, early_payment_discount (boolean), discount_percentage, discount_days, early_pay_discount_computation (base_amount/total_amount), example_preview
- c. Behaviour:

- **Example scenario:**
 - **Payment term:** “15 days”
 - If the customer pays **within 10 days** from the invoice date, they get **2% discount**.
 - **Then:**
 - **Discount Percentage** = 2%
 - **Discount Days** = 10
 - **Early Pay Discount Computation** defines whether 2% is applied on:
 - **Base Amount** (product value), or
 - **Total Amount** (base + tax).
 - **Example Preview** is used to show a human-readable text on the PDF of sale orders and invoices. For example, for an invoice dated **16/12/2025**:

Payment Terms: 15 days
Early payment discount: 100 if paid before 26/12/2025
 - If **Early Payment Discount = true**, then **Discount Percentage, Discount Days** and **Early Pay Discount Computation** fields are required and visible.
 - Orders from the **website** use the payment term **“Immediate Payment”** by default (no early payment discount).
- Then **Example Preview:**

Payment Terms: Immediate Payment

5. Discount Offers

- a. Defines discount programs that can generate multiple coupon codes.
- b. **Fields:** name, discount_percentage, start_date, end_date, available_on (sales/website), coupons (one2many)
- c. **Behaviour:**
 - **Available On = Sales**
Discount is only applied from the **backend**, on **Sale Orders**.
 - **Available On = Website**
Customers can apply coupon codes directly on the **website checkout**.
 - **Start Date** and **End Date** define the active period of the discount program.
 - One **Discount Offer** can have **multiple Coupon Codes**.

6. Coupon Codes

- a. Individual coupon codes are linked to a Discount Offer.
- b. **Fields:** code, expiration_date, status (unused/used), contact(many2one)
- c. **Behaviour:**
 - o A coupon with **Status = Used** cannot be applied again.
 - o Contact-based restriction:
 - If **Contact** is selected on the coupon, **only that contact** can use this coupon code.
 - If **Contact** is not selected, then **any customer** can use this coupon code.
 - o Coupon validity is controlled by both:
 - Its own **Expiration Date**, and
 - The parent **Discount Offer's Start/End Date**.

7. Transaction Flow

Process	Details
Purchase Order	Select Vendor, Product, Quantity, Unit Price, Tax(e.g. 5% / 10%).
Vendor Bill	Convert Purchase Order -> Vendor Bill, set invoice date, due date, and register payment details.
Sales Order	Select Customer, Payment Term, Product, Quantity, Unit Price, Tax, and optionally apply coupon codes.
Customer Invoice	Generate Invoice from Sales Order, confirm tax amounts and receive payment.
Payment	Register payment against a Vendor Bill or Customer Invoice.

(Stock of products is updated based on confirmed purchase and sales flows.)

Automatic Invoicing

- In the backend, there is a setting "**Automatic Invoicing**" (boolean).
- If **Automatic Invoicing = false**: after website payment, only the **Sale Order** is created; the internal user must manually create the **Customer Invoice** in the backend.

- If **Automatic Invoicing = true**: after a successful website payment, the system automatically generates the **Customer Invoice** from the Sale Order.

8. Reporting Requirements

For a **selected date period**, the system must generate:

a. Sales Report by Products

Shows: Product Name, Sold Quantity, Total Received Amount

b. Purchase Report by Products

Shows: Product Name, Purchased Quantity, Total Paid Amount

c. Sales Report by Customers

Shows: Customer Name, Total Orders, Paid Amount, Unpaid Amount

d. Purchase Report by Vendors

Shows: Vendor Name, Total Orders, Paid Amount, Unpaid Amount

Mockup- <https://link.excalidraw.com/l/65VNwvy7c4X/87OCIG1HG7n>