PRD Sample: E-Commerce Web Application

ShopGrid: A product marketplace with order management

2026

ShopGrid: E-Commerce Platform

Document type: Product Requirement Document (Sample) **Version:** 1.0 **Scope:** Customer storefront + seller/admin back office

---

1. Overview

ShopGrid is an e-commerce platform where customers browse and purchase products from one or multiple sellers. It includes a public storefront, a shopping cart, checkout with online payment, and an admin panel for order and product management.

---

2. User Personas

Customer A person browsing and purchasing products. May or may not have an account; guest checkout is supported.

Seller / Admin Manages the product catalogue, processes orders, and views sales reports. For v1 there is a single admin; multi-seller is out of scope.

---

3. Data Models

3.1 Product

Field	Type	Notes
`id`	integer	
`name`	string	
`slug`	string	URL-friendly unique identifier
`description`	text	Rich text (HTML)
`short_description`	text	Plain text; shown in listing cards
`sku`	string	Stock keeping unit; must be unique
`price`	decimal	Base price before tax
`sale_price`	decimal	Nullable; shown when lower than `price`
`stock_quantity`	integer	Current inventory count
`category_id`	integer	Foreign key to Category
`images`	array	Ordered list of image file references
`is_active`	boolean	Controls storefront visibility
`weight_grams`	integer	Used for shipping calculation
`created_at`	timestamp	

Field	Type	Notes

3.2 Category

Field	Type	Notes
`id`	integer	
`name`	string	
`slug`	string	
`parent_id`	integer	Nullable; enables two-level hierarchy
`sort_order`	integer	Controls display order

3.3 Order

Field	Type	Notes
`id`	integer	
`order_number`	string	Human-readable e.g. `SG-00234`
`customer_name`	string	
`customer_email`	string	
`customer_phone`	string	
`shipping_address`	JSON	`{ line1, line2, city, state, pincode, country }`
`status`	enum	`pending`, `confirmed`, `processing`, `shipped`, `delivered`, `cancelled`, `refunded`
`payment_status`	enum	`unpaid`, `paid`, `refunded`
`payment_reference`	string	Gateway transaction ID
`subtotal`	decimal	Sum of line item prices before tax/shipping
`tax_amount`	decimal	
`shipping_amount`	decimal	
`discount_amount`	decimal	From coupon
`total`	decimal	Final amount charged
`coupon_id`	integer	Nullable
`notes`	text	Customer-entered order notes
`created_at`	timestamp	

3.4 Order Item

Field	Type	Notes
`id`	integer	
`order_id`	integer	
`product_id`	integer	
`product_name`	string	Snapshot at time of order
`sku`	string	Snapshot
`unit_price`	decimal	Snapshot
`quantity`	integer	
`line_total`	decimal	`unit_price × quantity`

3.5 Coupon

Field	Type	Notes
`id`	integer	
`code`	string	Case-insensitive; unique
`discount_type`	enum	`flat` or `percent`
`discount_value`	decimal	
`minimum_order_value`	decimal	Nullable
`max_uses`	integer	Nullable = unlimited
`use_count`	integer	
`valid_from`	date	
`valid_until`	date	Nullable
`is_active`	boolean	

---

4. Features

Feature 1: Product Catalogue

US-01: Browse the product listing As a customer, I can browse all available products so that I can discover what is for sale.

Acceptance criteria:

1. The listing page shows active products as cards with: product image, name, price (and sale price if set), and an "Add to Cart" button.
2. Products where `is_active = false` or `stock_quantity = 0` are hidden from the listing.
3. Products can be filtered by category. Selecting a category shows only products in that category (and its children).
4. Products can be sorted by: Newest, Price (low to high), Price (high to low).
5. The listing paginates at 24 products per page.

US-02: View a product detail page As a customer, I can view the full details of a product so that I can make an informed purchase decision.

Acceptance criteria:

1. The product detail page shows: all images (with thumbnail gallery), name, price/sale price, short description, full description, SKU, stock status.
2. An "Add to Cart" button is shown if `stock_quantity > 0`. If the product is out of stock, the button is replaced with "Out of Stock" (disabled).
3. A quantity selector (1 to a maximum of `stock_quantity`, capped at 10) is shown before the add-to-cart button.

Feature 2: Shopping Cart

US-03: Add and manage items in cart As a customer, I can add products to my cart and adjust quantities so that I can prepare my order before checkout.

Acceptance criteria:

1. Adding a product to the cart shows a visual confirmation (toast or badge update).
2. The cart is persistent for the browser session. It does not require login.
3. On the cart page, each item shows: image, name, unit price, quantity (editable), and line total.

4. Changing quantity to 0 removes the item. A “Remove” link also removes the item.
5. The cart shows: subtotal, a coupon field, estimated tax, and order total.
6. If a product’s stock drops to 0 after being added to the cart, it is flagged on the cart page with a warning and cannot proceed to checkout.

US-04: Apply a coupon As a customer, I can enter a coupon code to receive a discount so that I can use promotions.

Acceptance criteria:

1. Entering a valid, active coupon code applies the discount and updates the order total.
2. If the order subtotal is below the coupon’s `minimum_order_value`, an error is shown: “This coupon requires a minimum order of □[value].”
3. If the code is invalid or expired, an inline error is shown.
4. Only one coupon can be applied per order.

Feature 3: Checkout

US-05: Enter shipping details As a customer, I can enter my shipping address and contact information so that my order can be delivered.

Acceptance criteria:

1. Required fields: full name, email, phone, address line 1, city, state, pincode.
2. Address line 2 is optional.
3. Email format is validated. Phone must be at least 7 digits.
4. A returning customer can save their address by creating an account (optional, not required to complete purchase).

US-06: Pay and place order As a customer, I can complete payment to confirm my order so that my purchase is finalised.

Acceptance criteria:

1. Payment is processed via the configured gateway (Razorpay or Stripe).
2. On successful payment: an Order record is created with `status = confirmed` and `payment_status = paid`; stock is decremented for each ordered product.
3. The customer is redirected to a confirmation page showing the order number, summary, and delivery estimate.
4. A confirmation email is sent to the customer within 60 seconds.
5. On payment failure: the customer is returned to the payment step with an error. No order is created and no stock is decremented.

Feature 4: Admin Back Office

US-07: Manage the product catalogue As an admin, I can create, edit, and deactivate products so that the storefront stays up to date.

Acceptance criteria:

1. The product list shows all products (active and inactive) with: name, SKU, price, stock, status.
2. Creating/editing a product uses a form with all fields from the Product model. Image upload supports multiple images (JPEG/PNG/WebP, max 5 MB each, max 10 images).
3. Setting `is_active = false` immediately hides the product from the storefront.
4. Products cannot be deleted if they have associated orders. Deactivate instead.

US-08: Process orders As an admin, I can view and update the status of orders so that I can manage fulfilment.

Acceptance criteria:

1. The order list shows: order number, date, customer name, total, payment status, order status. Default sort is newest first.
2. Filters: date range, order status, payment status.
3. Clicking an order opens its detail view with: customer info, shipping address, line items, payment info.
4. Admin can update order status via a dropdown. The following transitions are valid: `confirmed → processing → shipped → delivered`; `confirmed → cancelled`; `shipped → cancelled` (triggers a refund prompt).
5. When cancelling an order with `payment_status = paid`, the admin is prompted to initiate a refund via the gateway or mark it as manually refunded.
6. The customer receives an email notification when the order status changes to `shipped` or `cancelled`.

US-09: View sales reports As an admin, I can view sales summaries so that I can understand revenue and product performance.

Acceptance criteria:

1. A reports page shows for a selectable date range: total orders, total revenue, average order value, cancelled orders.
2. A best-selling products table shows the top 10 products by units sold.
3. All report data is exportable as CSV.

---

5. Email Notifications

Trigger	Recipient	Required Content
Order confirmed	Customer	Order number, items, total, shipping address, estimated delivery
Order shipped	Customer	Order number, courier name, tracking number/link
Order cancelled	Customer	Order number, refund amount and timeline

---

6. Scope Boundaries

In scope: - Product catalogue with categories - Guest checkout (no mandatory account) - Online payment (one gateway) - Coupon/discount codes - Admin order management and fulfilment - Transactional emails (confirmation, shipped, cancelled) - Sales reports

Out of scope for v1: - Multi-seller marketplace - Product reviews and ratings - Wishlist / saved items - Loyalty or rewards programme - SMS notifications - Shipping rate calculation via third-party API (flat-rate shipping only in v1) - Mobile app

---

7. Prerequisites

1. UI designs for: homepage, product listing, product detail, cart, checkout, order confirmation, and admin dashboard.
2. Payment gateway choice and API credentials.
3. Product catalogue: initial product data (name, description, images, prices, SKUs) provided by client.
4. Transactional email provider and sender domain.
5. Shipping rate table or flat-rate decision.
6. Tax configuration (rate and applicability).

This is a sample document showing the level of detail expected in a PRD for an e-commerce platform. Replace "ShopGrid" and the illustrative product types with your actual business, and expand or trim sections to match your scope.