

## **Business Requirements Document (BRD)**

### **Project: New E-commerce Website (Python Django backend + React frontend)**

#### **1) Executive Summary**

Build a secure, scalable, multi-device e-commerce platform for selling products online. Backend will be Python/Django (Django REST Framework) with PostgreSQL; frontend will be React (TypeScript). Phase 1 focuses on core shopping flow (browse → cart → checkout → payment), followed by enhancements (promotions, reviews, analytics, multi-warehouse, etc.).

---

#### **2) Objectives & KPIs**

##### **Primary Objectives**

- Enable online browsing and purchasing with <3 min average checkout time.
- Support 10k MAU at launch; scale to 100k MAU within 12 months.
- Achieve conversion rate ≥ 2.0% within 3 months.
- Page load (LCP) ≤ 2.5s on 4G; API p95 latency ≤ 300ms.

##### **KPIs**

- Sessions, add-to-cart rate, checkout start rate, conversion, AOV, revenue, CAC/LTV, cart abandonment, refund %, NPS/CSAT, p95 API latency, error rate, uptime.
- 

#### **3) In-Scope (Phase 1)**

- Product catalog with categories, variants (size/color), inventory.
- Search & browse (filters, sort, pagination, suggestions).
- Product detail page (images, pricing, stock, delivery ETA, specs).
- Cart, checkout, guest checkout, discount codes.
- Payments (Razorpay/Stripe), COD toggle (configurable by region).
- Shipping (rate calculation, tracking IDs), returns (RMA) basic.
- Customer accounts (profile, addresses, orders, wishlist).
- Admin portal (products, inventory, orders, coupons, users, content blocks).
- Email/SMS/push notifications (order, shipping, password, abandoned cart).
- Analytics events and dashboards (basic sales & funnel).

##### **Out of Scope (Phase 1)**

- Marketplace (multi-seller), subscriptions, B2B pricing tiers, loyalty points.
- Advanced personalization, headless CMS, international tax engines.

---

#### **4) Assumptions & Constraints**

- Single brand/storefront initially; multi-store ready in data model.
  - Single currency in Phase 1; multi-currency in Phase 2.
  - English first; i18n framework ready.
  - Mobile-first responsive UI, WCAG 2.1 AA target.
- 

#### **5) User Personas**

- **Guest Shopper:** discovers products and checks out quickly without account.
  - **Registered Customer:** tracks orders, saves addresses, manages wishlist.
  - **Operations Manager:** manages catalog, inventory, returns.
  - **Customer Support:** views orders, issues refunds/returns.
  - **Admin:** system config, user roles, promotions, content blocks.
- 

#### **6) User Journeys (Happy Paths)**

1. **Browse → PDP → Add to Cart → Checkout → Pay → Confirmation**
  2. **Search → Filter → Compare (via PDP) → Add to Wishlist**
  3. **Login → View Orders → Track Shipment → Submit Return (RMA)**
  4. **Admin: Create Product → Upload Images → Set Inventory → Publish**
- 

#### **7) Functional Requirements**

##### **7.1 Catalog & Product Management**

- Categories (tree), collections, SEO slugs.
- Products with variants (SKU level), attributes, pricing tiers (MRP, sale price), tax class.
- Media gallery (primary image + alt images, video URL), alt text required.
- Inventory per SKU; backorder policy by SKU (block/allow/notify date).
- Related products, upsells/cross-sells.

##### **7.2 Search & Discovery**

- Full-text search with typo tolerance, autocomplete suggestions.
- Filters: category, price range, brand, rating, attributes; sort by relevance/price/newest.
- Recently viewed, trending, and bestsellers modules.

### **7.3 Cart & Checkout**

- Persistent cart (guest via cookie + server; user via DB).
- Line-item quantity edit, remove; coupon application (fixed %, fixed value, free shipping, min order).
- Shipping address, billing address, shipping method, tax total, order summary.
- Guest checkout; account creation post-order optional.
- Fraud checks (velocity, AVS if available, IP risk list).

### **7.4 Payments**

- Gateways: Razorpay (India) and/or Stripe (global). Webhooks for payment success/failure/refund.
- Wallets/UPI/cards/netbanking (gateway dependent). COD (configurable).
- Refunds: full/partial from admin with audit log.

### **7.5 Shipping & Fulfillment**

- Shipping rate calculation (table-based; later carrier APIs).
- Order statuses: Pending → Paid → Picking → Shipped → Delivered → Returned/Refunded.
- Tracking number capture; customer notifications at ship/delivery.

### **7.6 Returns & Cancellations (RMA)**

- Self-service return request with reason, condition, evidence images.
- Approve/Reject; generate RMA; restock on completion; refund path.

### **7.7 Accounts & Authentication**

- Email/password + OAuth (Google/Apple optional), 2FA optional.
- Address book, default address, saved payment token (gateway vaulted).
- Order history with invoice/receipt PDFs; download GST invoice (if India).

### **7.8 Promotions & Content**

- Coupons (code, duration, usage limits, user eligibility).
- Site-wide banners, homepage sections, content blocks (admin-editable).

### **7.9 Reviews & Ratings (Phase 2 optional)**

- Verified-purchase reviews, star ratings, moderation queue.

### **7.10 Notifications**

- Email (SendGrid/Mailgun), SMS (Twilio/MSG91), web push (Phase 2).
- Templates for order confirmations, shipping, delivery, password reset.

### **7.11 Analytics & Reporting**

- Event tracking (view\_item, add\_to\_cart, begin\_checkout, purchase).
  - Admin dashboards: Sales, AOV, top products, conversion funnel.
  - Export CSV for finance.
- 

## 8) Non-Functional Requirements

- **Performance:** LCP ≤ 2.5s; TTI ≤ 3s; p95 API ≤ 300ms; image optimization (WebP/AVIF).
  - **Security:** OWASP ASVS L2; HTTPS everywhere; CSP; rate limiting; CSRF on forms; encryption at rest for secrets; rotate keys.
  - **Compliance:** GDPR-ready, cookie consent; PCI-DSS SAQ-A (redirect/hosted fields); Indian tax (GST) support.
  - **Availability:** 99.9% monthly; zero-data-loss RPO with daily backups; RTO ≤ 4h.
  - **Scalability:** Horizontal scaling via containers; CDN for static/media; cache with Redis.
  - **Accessibility:** WCAG 2.1 AA; keyboard nav; alt text; color contrast.
  - **Localization:** i18n framework, currency/number formatting.
- 

## 9) Tech Stack (Baseline)

- **Backend:** Python 3.12, Django 5.x, Django REST Framework, PostgreSQL, Redis, Celery (async tasks), Pillow, drf-spectacular (OpenAPI).
  - **Frontend:** React 18 (TypeScript), Vite or Next.js (SSR optional), React Router, Zustand/Redux Toolkit, React Query, TailwindCSS, Headless UI.
  - **Infra/DevOps:** Docker, Nginx, CDN (Cloudflare), GitHub Actions CI/CD, Sentry, Prometheus + Grafana, ELK/Opensearch for logs.
- 

## 10) High-Level Architecture

- SPA/SSR React app consumes REST API from Django.
  - Auth via JWT (access/refresh) or cookie-based session; CSRF handled.
  - Media stored in S3-compatible storage; signed URLs; image thumbnailing via worker.
  - Webhooks for payments; Celery workers process emails, thumbnails, exports.
- 

## 11) Data Model (Core Entities)

- **User**(id, email, password\_hash, role, is\_active, created\_at)
- **Address**(id, user\_id, name, phone, line1, line2, city, state, country, pincode, type)
- **Category**(id, parent\_id, name, slug, meta)

- **Product**(id, name, slug, description, brand, tax\_class, status)
  - **ProductVariant**(id, product\_id, sku, attributes(jsonb), price\_mrp, price\_sale, barcode, weight)
  - **Inventory**(id, variant\_id, qty\_available, backorder\_policy, warehouse\_id)
  - **Media**(id, product\_id, url, alt, sort)
  - **Cart**(id, user\_id/null, session\_key)
  - **CartItem**(id, cart\_id, variant\_id, qty, unit\_price, total)
  - **Coupon**(id, code, type, value, min\_subtotal, start\_at, end\_at, usage\_limit, per\_user\_limit)
  - **Order**(id, user\_id/null, number, status, subtotal, discount\_total, tax\_total, shipping\_total, grand\_total, currency, payment\_status)
  - **OrderItem**(id, order\_id, variant\_id, sku, name, attributes, qty, unit\_price, total)
  - **Shipment**(id, order\_id, carrier, tracking\_no, status, shipped\_at, delivered\_at)
  - **ReturnRMA**(id, order\_id, status, reason, approved\_by, refund\_amount)
- 

## 12) API Surface (Examples)

### Public

- GET /api/v1/catalog/categories/
- GET /api/v1/catalog/products/?q=&category=&sort=&min\_price=&max\_price=
- GET /api/v1/catalog/products/{slug}/
- POST /api/v1/cart/items/ (variant\_id, qty)
- PATCH /api/v1/cart/items/{id}/
- POST /api/v1/checkout/ (addresses, shipping\_method, coupon)
- POST /api/v1/payments/create-intent/
- POST /api/v1/payments/webhook/

### Admin (JWT role=admin)

- POST /api/v1/admin/products/
  - PATCH /api/v1/admin/products/{id}/
  - POST /api/v1/admin/coupons/
  - POST /api/v1/admin/orders/{id}/refund/
- 

## 13) Roles & Permissions

- **Admin:** full control; manage users/roles.

- **Ops:** manage orders, shipments, returns, inventory.
  - **Content Manager:** banners, pages, collections, SEO.
  - **Support:** read-only orders, create refunds/returns with approval.
- 

## 14) Reporting & Finance

- Daily sales report (CSV), tax summary by state, refund log, coupon usage.
  - Reconciliation report: gateway vs orders.
- 

## 15) Integrations

- Payments: Razorpay/Stripe.
  - Email: SendGrid/Mailgun; SMS: MSG91/Twilio.
  - Shipping: manual CSV export in Phase 1; Shiprocket/Delhivery API in Phase 2.
  - Analytics: GA4 + server-side events; Meta/Ads pixels.
- 

## 16) Content & SEO

- SEO fields per page/product (title, meta, og tags), XML sitemap, robots.txt.
  - Clean URLs, breadcrumbs, canonical links, schema.org markup.
- 

## 17) Monitoring & Alerts

- SLOs for latency, error rate; alerts via Slack/Email.
  - Audit logs for admin actions (who/what/when), immutable.
- 

## 18) Testing & QA

- Unit tests (backend 80%+, frontend key flows); integration tests for APIs.
  - E2E tests (Cypress/Playwright): browse, add to cart, checkout, payment webhook.
  - Load test (k6/Locust) to validate  $p95 \leq 300\text{ms}$  up to 200 RPS.
- 

## 19) Release Plan & Milestones (Indicative)

- **M1 (Weeks 1-2):** Project setup, auth, catalog read APIs, basic UI shell.
- **M2 (Weeks 3-4):** Cart/checkout APIs & UI, payment integration.
- **M3 (Weeks 5-6):** Admin portal (products, inventory), shipping & order states.

- **M4 (Weeks 7-8):** Coupons, analytics events, SEO, notifications.
  - **M5 (Weeks 9-10):** QA, perf hardening, content load, go-live.
- 

## 20) Risks & Mitigations

- **Payment failures/chargebacks:** webhook retries; idempotent ops; fraud rules.
  - **Inventory mismatch:** stock reservations at checkout; oversell protection.
  - **SEO underperformance:** pre-render/SSR for key pages; sitemaps, Core Web Vitals.
  - **Scale spikes:** CDN cache, read replicas, autoscaling.
- 

## 21) Open Questions (to refine with stakeholders)

- Target regions, currencies, and payment methods for launch?
  - Shipping partners & service levels (SLA, COD pickup)?
  - Tax treatment (GST rates, HSN/SAC mapping) and invoice format needs?
  - Return policy window and conditions?
  - Brand style guide, tone of voice, content governance?
- 

## 22) Acceptance Criteria (Go-Live)

- A shopper can find a product, add to cart, pay, and receive an order confirmation email.
  - Admin can create a product, manage inventory, fulfill an order, and process a refund.
  - Core KPIs dashboard shows orders, revenue, conversion.
  - Security review passed (OWASP ASVS L2 checks) and load test p95 ≤ 300ms @ 200 RPS.
- 

## 23) Appendix: Environment & Ops

- **Environments:** Dev, Staging, Prod with isolated DBs & S3 buckets.
- **CI/CD:** PR checks (lint, tests), auto-deploy to Staging, manual Prod gate.
- **Backups:** Nightly DB backups retained 30 days; restore drills quarterly.
- **Config:** 12-factor; secrets via environment manager; feature flags.