**📅 Full Implementation Plan (Backend → Frontend)**

**Phase 1: Backend Foundation Setup**

🔹 **Goal**: Establish the core backend infrastructure and APIs.

**Tasks:**

* Set up **Laravel 11 project** with Docker.
* Configure **MySQL/MongoDB**, **Redis** (cache & queues).
* Implement **authentication system**:

I want to use OAuth so need change dependency sanctum is not matching for that currently I use "laravel/sanctum": "^4.2"

* + Customer login/register (Laravel Breeze).
  + Admin roles with RBAC (spatie/laravel-permission).
* Define **core database entities**:
  + Products, Variants, Categories.
  + Customers, Addresses.
  + Inventory (qty\_on\_hand, qty\_reserved).
  + Orders, Payments, Shipments, Coupons.
* Implement **migration & seeder scripts** for demo data.

**Deliverables:**

* Running backend container with DB.
* Auth APIs live.
* Initial seed data (sample products, categories, customers, inventory).

**Phase 2: Backend Core Business Logic**

🔹 **Goal**: Build APIs & logic that power the customer storefront.

**Tasks:**

* **Product & Catalog APIs** (GET /products, GET /categories, search & filters).
* **Inventory service**: stock tracking, reservations, audit logs.
* **Cart APIs**: add, update, remove items with reservation logic.
* **Coupon system**: validation, atomic usage tracking.
* **Checkout service**:
  + Validate stock.
  + Create order with PENDING status.
  + Apply taxes, shipping, discounts.

**Deliverables:**

* Tested APIs for catalog, inventory, cart, checkout.
* Reservation scheduler (auto-release expired).

**Phase 3: Backend Advanced Features**

🔹 **Goal**: Complete the business workflows.

**Tasks:**

* **Payment service**:
  + Integrate Stripe Payment Intents (test mode).
  + Webhook handling (/webhooks/stripe).
  + COD support.
* **Order Management**: status workflow (Pending → Paid → Fulfilled → Delivered).
* **Shipment APIs**: create shipment, update status, carrier/tracking.
* **Reporting APIs**: sales, top products, inventory status, CSV export.
* **Notification service**: order confirmation, shipping emails via SMTP (queued).

**Deliverables:**

* Fully working order → payment → shipment cycle.
* Reports APIs available.
* Emails sent asynchronously.

**Phase 4: Backend QA & Optimization**

🔹 **Goal**: Ensure stability, performance, and security.

**Tasks:**

* Unit tests (Pest/PHPUnit): inventory invariants, cart, payment idempotency.
* Integration tests: full order lifecycle.
* Security: CSRF, input validation, prepared statements, password hashing.
* Performance: query optimization, Redis caching.

**Deliverables:**

* Minimum 80% test coverage.
* Backend ready for frontend integration.

**🖥 Frontend Implementation Plan**

**Phase 5: Customer Storefront UI**

🔹 **Goal**: Build customer-facing frontend in Vue.js (Inertia + Tailwind).

**Tasks:**

* **Layout & Navigation**: responsive, mobile-first design.
* **Catalog pages**: product listing, sorting, filtering.
* **Product details page**: variants, stock display, add-to-cart.
* **Cart page**: update/remove items, show totals, coupon application.
* **Checkout flow**:
  + Guest checkout & account creation.
  + Address forms.
  + Shipping & tax calculation.
  + Payment (Stripe widget, COD option).
  + Progress indicators.
* **Customer accounts**: login/register, profile & address management, order history.
* **Notifications**: success/error messages, order confirmation view.

**Deliverables:**

* Fully functional storefront UI integrated with backend APIs.

**Phase 6: Admin Dashboard UI**

🔹 **Goal**: Provide management tools for staff.

**Tasks:**

* **Dashboard**: role-based widgets (sales summary, low stock alerts, pending orders).
* **Product & Catalog Management**: CRUD UI with forms, image upload, category management.
* **Inventory Management**: stock view, adjustments with audit reasons.
* **Orders Management**: filter/search orders, update status, cancel/refund.
* **Shipments Management**: create shipments, track status, bulk actions.
* **Promotions & Coupons**: create/manage coupons with limits & dates.
* **Reports & Analytics**: charts & tables for sales, inventory, top products, export CSV.

**Deliverables:**

* Responsive admin UI integrated with APIs.
* RBAC ensures only correct roles access features.

**Phase 7: Frontend QA & UX Polish**

🔹 **Goal**: Ensure usability, accessibility, and performance.

**Tasks:**

* End-to-end tests (Cypress):
  + Customer journey (browse → checkout → payment → order tracking).
  + Admin workflows (catalog, inventory, orders, shipments).
* Mobile responsiveness testing.
* Accessibility audit (WCAG 2.1 AA compliance).
* Performance optimization (lazy loading, image compression).

**Deliverables:**

* Polished, production-ready frontend.

**⚡️ Final Delivery:**

* **Backend**: Secure, tested APIs with business logic, inventory control, payments, and reporting.
* **Frontend**: Modern, responsive UIs for customers and admins with full feature coverage.
* **System Integration**: Fully working E-commerce + Stock Management flow from browsing to fulfillment.