Drawing from the recommended high-performance tech stack of **Go (Golang)** for the backend, **Next.js 16** for the frontend, and **Flutter** for mobile, here is a comprehensive folder structure for the **NobleMart** multivendor marketplace. This structure is designed to fulfill the mission of being the "fastest and safest trade engine" in Africa while adhering to the established brand guidelines 1-3.

1. Frontend Structure (Next.js 16 Storefront)

Designed for SEO excellence and high-speed rendering to ensure the Largest Contentful Paint (LCP) stays under 2.5 seconds 4, 5.

```
/noblemart-frontend
├── /public
│   ├── /fonts (Poppins: Light, Regular, Bold, Black) [6]
│   ├── /images (Dark-skinned photography focus) [6, 7]
│   └── /icons (Brand-colored rounded icons) [8]
├── /src
│   ├── /components
│   │   ├── /ui (Buttons #01C2CA, Cards, Hero Gradients) [9, 10]
│   │   ├── /layout (Sticky Header, Bottom Nav, Jumia-style Sidebar)
│   │   ├── /product (Price display in Poppins Bold, Stock status)
│   │   └── /seo (JsonLdSchema.tsx, MetaManager.tsx) [4, 13]
│   ├── /hooks (useCart, useAuth, useGeoLocation) [14, 15]
│   ├── /lib (Analytics, Paystack/Flutterwave SDKs) [2, 16]
│   ├── /pages
│   │   ├── /category (Faceted filtering: Price, Brand, Rating) [12]
        ├── /product/[slug] (WebP gallery, "Save for Later" button)
     ├── /static pages/[slug] ("Save for about us page, contact us,
     privacy & more")
│   │   ├── /checkout (1-2 page multi-vendor cart split) [15-17]
│   │   └── /search (Smart Search with localized Nigerian keywords)
│   └── /styles (Tailwind config with brand palette: #01C2CA,
#5C1BE8) [9, 10]
```

2. Backend Structure (Go Microservices)

A microservices architecture built for massive concurrency and memory safety, essential for NobleMart-scale traffic 2, 3, 20.

```
/noblemart-backend
├── /cmd (Entry points for each microservice)
├── /internal
│   ├── /user (Auth, KYC for vendors, customer profiles) [21]
│   ├── /catalog (Inventory locks, category hierarchy, variants) [12,
22]
│   ├── /order (Multi-vendor splitting logic, status transitions)
[11, 17]
```

```
│   ├── /payment (Escrow logic, Paystack T+1 settlements) [2, 4, 16]
│   ├── /logistics (OTP verification, Route optimization) [18, 23,
24]
│   └── /notifications (Push notifications, WhatsApp API bridge) [25,
26]
├── /pkg (Shared libraries)
│   ├── /db (PostgreSQL & Redis configurations) [2, 27]
│   ├── /seo (Dynamic JSON-LD and XML Sitemap generators) [4, 13, 28]
│   └── /validator (Content scoring for vendor uploads) [4, 29]
└── /api (gRPC and REST protocol definitions) [20]
```

3. Admin Panel Structure (The Nerve Center)

The central "Control Center" for system-wide management and vendor vetting 21, 30.

```
/noblemart-admin-panel
├── /dashboard (GMV metrics, Sales by category/region) [21, 31]
├── /vendor-management
│   ├── /vetting (ID verification, Business permits review) [13, 21]
│   └── /commissions (Dynamic engine: Category-based percentages)
[13, 21]
├── /catalog-moderation (Product approval, Fraud detection AI) [6,
22]
├── /finance (Escrow balance, Payout reconciliation, Refund portal)
[16, 32]
└── /settings (Tax rules, Shipping zones, Currencies) [21]
```

4. Backend SEO Structure (Custom Module)

Modeled after RankMath to automate marketplace visibility 4, 13, 28.

```
/pkg/seo
├── schema_generator.go (Automated JSON-LD for Price/Stock/Ratings)
[4, 12, 13]
├── sitemap_manager.go (Dynamic XML update on product addition) [5,
13, 28]
├── meta_tag_api.go (Editable titles and Open Graph tags) [5, 13, 28]
└── content_analyzer.go (Keyword density and readability scoring) [4,
29]
```

5. Comprehensive CMS Structure

Headless CMS logic for managing non-transactional content 28, 33.

```
/noblemart-cms
├── /blog (Style guides, tips, local promotions) [28, 32, 34]
├── /static-pages (About Us, Terms, FAQs/Knowledge Base) [28, 33]
├── /marketing (Hero carousels, Flash sale banners) [11, 21]
```

```
└── /services (Real Estate listings, Booking calendar logic) [Prompt 
Context]
```

6. Vendor Panel & SEO Structure

The "Business Tool" for sellers to manage their NobleMart stores 11, 35.

```
/noblemart-vendor-panel
├── /inventory (Stock management, "Flash Sale" scheduler) [11, 35]
├── /analytics (Most viewed products, Customer demographics) [11, 35]
├── /orders (Sub-order management: "Pending" to "Ready for Pickup") 
[11, 35]
├── /seo-optimization (Individual listing meta tags and image 
alt-text) [4, 28]
├── /whatsapp-sync (Direct inventory-to-catalog bridge) [25, 26]
└── /dispute-center (Customer inquiry handling, Returns management) 
[19, 32]
```

7. More Structures (Mobile & Logistics)

The tripartite Flutter app ecosystem ensuring performance on budget devices 36-38.

```
/noblemart-mobile-apps
├── /customer-app (Smart search, In-app wallet, Push notifications) 
[11, 18, 39]
├── /vendor-app (Photo-based product upload, Earnings summary) [35]
├── /delivery-app (Google Maps GPS, OTP verification, Route 
optimization) [18, 23, 24]
└── /agent-app (NobleForce: Commission portal for community orders) 
[25, 40]
```

Key Strategic Logic within the Folders:

- **Infrastructure:** Uses **Redis** for high-speed caching of session and cart data to prevent database overload during high traffic 2, 33, 34.
- **Trust Building:** The payment service folder contains an **Escrow Logic** that holds funds until a 7-day return window expires or delivery is confirmed via the logistics OTP module 16, 18, 26.
- **Local Logic:** The checkout folder in the frontend includes custom fields for **landmarks and digital map geo-pins** to overcome Nigerian addressing barriers

This comprehensive folder structure for **NobleMart** is designed to support a high-performance, multi-vendor ecosystem. It follows the recommended **Next.js 16 (Frontend)** and **Go (Golang) (Backend)** stack to ensure the platform remains the "fastest and safest trade engine in Africa" 1-3.

1. Marketplace Frontend Structure (Next.js 16)

This structure focuses on **Server-Side Rendering (SSR)** and **Static Site Generation (SSG)** to ensure peak performance and Jumia-style high-conversion layouts 2, 4, 5.

- **/src/components**: Modular UI components.
- **/ui**: Branded buttons (**#01C2CA**), inputs, and rounded icons matching the **Poppins** aesthetic 6-8.
- **/layout**: Jumia-style sticky headers, bottom navigation for mobile-first views, and mega-menus 6, 7.
- **/marketplace**: **Multi-vendor cart split** visualizations and "Save for Later" modules to cater to speculative shoppers 6, 9, 10.
- **/src/pages**: Optimized routes.
- **/category**: Faceted filtering systems (price, brand, rating) 11.
- **/product**: Dynamic product pages with **WebP/AVIF** image optimization and JSON-LD scripts 11-13.
- **/checkout**: High-conversion 1-2 page flow with **Escrow** payment notices 10, 14.
- **/src/styles**: Tailwind configurations using the **NobleMart palette**: Teal (#01C2CA), Purple (#5C1BE8), and Federal Blue (#020249) 6, 7.

2. Marketplace Backend Structure (Go Microservices)

A microservices architecture built for **massive concurrency** and memory safety, handling thousands of simultaneous transactions 1-3.

- **/cmd**: Entry points for the main application services.
- **/internal**: Core business logic.
- **/catalog**: Real-time inventory synchronization and database "locks" to prevent **overselling** 15-17.
- **/orders**: Automated **sub-order generation** and routing for multi-vendor split logic 9, 18.
- **/payments**: Integration with **Paystack/Flutterwave** featuring an **Escrow Logic** that holds funds until delivery verification 14, 19, 20.
- **/logistics**: Landmark-based addressing, digital map geo-pins, and **OTP verification** systems 20-22.
- **/pkg**: Shared packages, including the custom SEO engine and database drivers for **PostgreSQL** and **Redis** 3, 23, 24.

3. Comprehensive Admin Marketplace Panel Structure

The "Nerve Center" for system-wide control, vendor vetting, and financial oversight 25, 26.

- **/admin/vendors**: Tiered onboarding for reviewing vendor IDs and business permits 26, 27.
- **/admin/commissions**: **Dynamic Commission Engine** to set percentages by category (e.g., 5% electronics, 10% fashion) 26, 27.
- **/admin/finance**: Dashboards for **T+1 automated settlements**, escrow balances, and payout reconciliation 12, 14, 25.
- **/admin/moderation**: Tools to audit product listings for quality control and "fake product" detection 26, 28, 29.

4. Advanced Backend SEO Structure (RankMath-Style)

A custom-built engine designed to automate search visibility through high-quality metadata and schema 12, 27, 30.

- **/seo/schema-generator**: Automatically generates **JSON-LD scripts** for Product, Aggregate Rating, and Breadcrumbs to secure "Rich Snippets" 12, 27, 31.
- **/seo/sitemap-manager**: A dynamic **XML sitemap generator** that updates instantly when new products are added 27, 30, 31.
- **/seo/content-analyzer**: An **AI-driven tool** that scores vendor product descriptions for keyword density and readability 12, 32.
- **/seo/meta-manager**: Global fields for editable title tags, canonical URLs, and Open Graph tags for social sharing 30, 31.

5. Comprehensive CMS Structure (Advanced Headless)

Managed via a headless approach (e.g., Strapi) to power marketing content and educational resources 23, 30.

- **/cms/blog**: Standard blog posts for **style guides, tips, and promotions** to engage the community 24, 30.
- **/cms/knowledge-base**: FAQ and help systems to reduce support load 30, 33.
- **/cms/static-pages**: Management for "About Us," "Terms of Service," and "Safe Shopping" guides 30.
- **/cms/media-library**: Optimized storage for **well-lit, dark-skinned photography** to foster local connection 6, 8, 13.

6. Vendor Panel & SEO Structure

Modeled after **WCFM**, this "Business Tool" empowers sellers to manage their own storefronts and organic reach 26, 34, 35.

- **/vendor/inventory**: Real-time price management and **"Flash Sale" scheduling** 34.
- **/vendor/analytics**: Insights into total sales, most viewed products, and **customer demographics** 34, 35.
- **/vendor/seo-tools**: Individual listing meta-tag editing and image alt-text optimization 12, 26.
- **/vendor/whatsapp-sync**: An API bridge allowing vendors to sync their NobleMart inventory with their **WhatsApp Business Catalog** 36, 37.

- **/vendor/fulfillment**: Order status transitions from "Pending" → "Processing" → "Ready for Pickup" 34, 35.

7. Specialized Listing Structures (More Marketplace Focus)

To expand NobleMart's capabilities beyond physical goods into services and property Prompt Context.

- **/listings/services**: Booking and appointment logic for local professionals.
- **/listings/real-estate**: Property listing modules with integrated map pins and agent contact flows.
- **/listings/booking**: Availability calendars for events or hotel reservations.
- **/listings/auctions**: Bargaining or limited auction features for specific categories 38.

Strategic Technical Note: All components must use **image lazy loading** and **code minification** to ensure performance on budget/mid-range smartphones, which account for over 82% of transactions in the target market 6, 39, 40.

This comprehensive folder structure for **NobleMart** is designed to support a high-performance, multi-vendor ecosystem. It utilizes the recommended **Go (Golang) + PostgreSQL** backend for massive concurrency and **Next.js 16** for an SEO-optimized, blazing-fast frontend 1, 2. This architecture ensures NobleMart becomes the "fastest and safest trade engine in Africa" by addressing the "Lagos reality" of low-bandwidth connections

1. Marketplace Frontend Structure (Next.js 16)

The frontend focuses on **Server-Side Rendering (SSR)** and **Static Site Generation (SSG)** to pre-render pages for SEO and performance 6, 7.

- **/src/components**: Branded UI components using the **Poppins** font family 8, 9.
- **/ui**: Branded buttons (#01C2CA), inputs, and rounded icons matching the friendly brand style 9, 10.
- **/layout**: Jumia-style sticky headers, persistent cart icons, and bottom navigation for mobile views 9.
- **/marketplace**: Logic for **multi-vendor checkout**, cart splitting, and the "Save for Later" feature to cater to speculative shoppers 8, 11, 12.
- **/src/pages**: Optimized routes for the shopper-facing site.
- **/category**: Faceted filtering (price, brand, rating) and grid layouts (4–5 columns on desktop) 9, 13.
- **/product**: Dynamic pages featuring **WebP/AVIF** optimized images and real-time stock updates 13, 14.
- **/checkout**: High-conversion 1-2 page flow collecting **landmarks and digital map geo-pins** for localized delivery 12, 15.
- **/src/styles**: Tailwind CSS configurations utilizing the primary **Robin Egg Blue (#01C2CA)** and **Electric Indigo (#5C1BE8)** palette 9.

2. Marketplace Backend Structure (Go Microservices)

A microservices approach is used to isolate services like catalog, orders, and payments, ensuring the system can handle thousands of simultaneous transactions 1, 6, 16.

- **/cmd**: Main entry points for the various microservices.
- **/internal/services**: Core business logic decoupled for scalability 16.
- **/catalog**: Manages the hierarchy, product variants, and **database locks** to prevent inventory "race conditions" 13, 17.
- **/order**: Handles **automated sub-order generation** and logistics triggers for multi-vendor split baskets 11, 18.
- **/payment**: Integrated with Paystack/Flutterwave; contains the **Escrow Logic** to hold funds until delivery confirmation 18, 19.
- **/logistics**: Logic for route optimization, **OTP verification**, and automated Cash on Delivery (COD) remittance logs 20, 21.

- **/pkg**: Shared packages including database drivers (PostgreSQL/Redis) and the custom SEO engine 1, 22, 23.

3. Comprehensive Admin Marketplace Panel Structure

The "Nerve Center" for NobleMart staff to manage the entire ecosystem and moderate listings 24, 25.

- **/admin/vendors**: Management of **tiered onboarding** and KYC (ID/business permit verification) 25, 26.
- **/admin/commissions**: The **Dynamic Commission Engine** to set rules by category or individual vendor 25, 26.
- **/admin/finance**: Dashboards for T+1 automated settlements, escrow balances, and payout reconciliation 14, 18, 26.
- **/admin/moderation**: Tools to audit listings for quality control and prevent "fake products" 25, 27.

4. Advanced Backend SEO Structure (RankMath-Style)

A custom engine to automate search visibility and generate "Rich Snippets" 14, 26, 28.

- **/seo/schema-generator**: Automatically creates **JSON-LD scripts** for Product, Aggregate Rating, and Breadcrumbs 14, 26, 29.
- **/seo/sitemap-manager**: Dynamic XML sitemap generator that updates instantly when new products are added 26, 28.
- **/seo/meta-manager**: Interface for editing global title tags, meta descriptions, and Open Graph tags for all pages 26, 28.
- **/seo/ai-analyzer**: An integrated tool to score product descriptions for keyword density and readability 14.

5. Comprehensive CMS Structure

Headless CMS logic (like Strapi) for managing marketing content and community engagement 7, 25, 28.

- **/cms/blog**: Standard-setting blog manager for style guides, tips, and local promotions 23, 28, 30.
- **/cms/static-pages**: Management for "About Us," "Safe Shopping" guides, and FAQs/Knowledge Base to reduce support load 28, 30.
- **/cms/banners**: Tools to schedule and manage promotional banners and flash sale graphics 25.
- **/cms/media**: Storage for **well-lit photography of dark-skinned individuals** to foster local connection 8, 10, 31.

6. Vendor Panel & SEO Structure

Modeled after WCFM to provide vendors with a professional "Business Tool" 32, 33.

- **/vendor/inventory**: Real-time stock management and **"Flash Sale" scheduling** 32.

- **/vendor/analytics**: Dashboards showing total sales, most viewed products, and customer demographics 32, 33.
- **/vendor/seo-tools**: Ability for vendors to edit their own store/product meta tags and URL slugs 25, 28.
- **/vendor/whatsapp-sync**: The **WhatsApp-to-Web** bridge allowing vendors to sync inventory with their WhatsApp catalogs 34, 35.
- **/vendor/fulfillment**: Order status tracking from "Pending" → "Ready for Pickup" 32, 33.

7. Specialized Marketplace Structures

To differentiate NobleMart, specific modules address local gaps like trust and rural reach 34, 36.

- **/marketplace/escrow-portal**: Dedicated interface for managing the 24-hour "no-complaint" window before vendor payout 18, 34.
- **/marketplace/nobleforce**: The **Agent Model** portal where local community agents earn commissions for placing orders for others 34, 37.
- **/marketplace/loyalty**: Logic for "NobleMart Plus" memberships, loyalty points, and wallet discounts 17, 30.
- **/marketplace/services**: Special listing modules for services and real-estate products with map integrations 13.