PART 1: MAIN PROJECT DOCUMENTATION (Full Overview)

Kioserra is a **multi-tenant digital commerce platform** tailored for **Kenyan and African micro, small and medium enterprises (MSMEs)**. It provides small vendors with an all-in-one solution to:

- Manage inventory

- Sell online

- Offer products on credit (vendor-led)

- Collect payments via M-Pesa

- Build trust with repeat customers

Vendors get a full **admin dashboard**, their own **storefront (like vendora.ke/store/janes-kiosk)**, and tools for managing sales, staff, and orders. Customers browse products, place orders, and get updates via WhatsApp and email.

🔥 Problem It Solves

Most small shops in Kenya:

- Have no inventory system

- Operate entirely offline

- Lose track of customer credit

- Lack visibility or digital presence

- Struggle with cash flow & sales tracking

🎯 Who It's For

- **Small shop owners (kiosks, groceries, fashion stores, agro dealers)**

- **Hustlers and youth entrepreneurs**

- **NGOs distributing goods**

- **Micro-financers seeking vendor partnerships**

- **Rural communities needing credit tools**

💡 Key Features (High Level)

Feature	Description
Vendor Dashboard	Manage store, orders, stock, credit, and staff
Storefront	Unique link customers can shop from and pay via M-Pesa
Credit Management	Let loyal customers buy now, pay later (managed by vendor)
Super Admin Console	Monitor vendors, usage, payments, ban abuse
Inventory Tracking	Real-time view of available products and their sales flow
WhatsApp Orders	Notify vendors and customers via WhatsApp instantly
Subscription Model	KES 1500/year (or $10 USD); no free trial
Customer Flow	Users browse public store, sign in to order
SEO Friendly	Each vendor store has meta tags + schema markup
Offline Support	App caches data for low-connectivity users

💸 Monetization Model

- **Vendor subscription**: KES 1,500/year (~$10)

- **Optional credit tools with interest cut (future)**

- **MPesa transaction fees covered by vendors**

📈 MVP Launch Timeline

- **Start**: June 30

- **Go Live MVP**: End of July

- **Scale aggressively**: August → December

🏆 Vision

Become the **"Shopify for Africa's small businesses"**, with integrated mobile commerce, peer-managed credit, and localized financial tools—built for where the world is going, not just where it is.

FRONTEND DEVELOPER DOCUMENTATION – KIOSERRA

This section is **fully tailored for you** as the Flutter developer building Kioserra. We're going to treat this like a production-grade SaaS platform that will scale to 100,000+ users in a year. **Everything you build needs to be sleek, modern, minimal, performant, and mobile-first.** Let's go deep.

🎯 OBJECTIVE

Build a **pixel-perfect, responsive Flutter app** (for mobile & possibly PWA/web later) that allows:

- Vendors to register, log in, run their store, track sales, and manage customers.

- Customers to browse stores, order products, and interact via WhatsApp.

- Admins to manage platform activity.

🛠️ TECH STACK

Tool	Purpose
Flutter	Core front-end framework
Firebase Auth / Supabase Auth	Authentication
Supabase	Realtime DB + RLS policies

Riverpod / BLoC	State management (your choice)
Hive / Isar	Local data cache (for offline sync)
Dio / http	Network layer
Lottie	Animations
GoRouter / AutoRoute	Routing
Intl	i18n (for future expansion)
Flutter SVG	Logos, icons
CachedNetworkImage	Performance image loading
MPesa Daraja API	Vendor payment subscriptions
Google Fonts	Brand typography

🎨 THEME & UI SPECS

Property	Value
Primary Color	#0D47A1 (Royal Blue)
Accent Color	#FF6F00 (Vibrant Orange)
Background	#F9F9F9 (Light Gray)
Font	Poppins (clean, modern, local-vibe)
Icons	Fluent / Material Icons + Custom SVGs
CTA Buttons	Rounded, gradient, floating when needed

📱 SCREENS TO BUILD (Full)

✅ A. Public Views (Customer-facing)

1. **Landing Page**

   ○ Hero CTA: "Set up your shop. Go digital in minutes."

   ○ Features section (why use us)

   ○ Pricing preview

- "Contact Us" CTA

- "Start Selling" → Auth redirect

2. **Storefront (/store/:slug)**

- Store name, description, logo

- Product grid (with categories filter)

- "Order on WhatsApp" → auth modal if not logged in

- Product page with images, price, short details

- SEO tags and shareable link

✅ **B. Auth Screens**

1. **Login**

- Email + password

- Google Sign-in

- Forgot password

2. **Register**

- Role (Vendor or Customer)

- Email, name, phone, password

- Accept T&Cs

3. **Password Reset**

- Email-based flow

- Confirmation banner

✅ **C. Vendor Dashboard**

1. **Home**

   - Metrics: Daily sales, orders, stock alerts

   - "Your trial ends in X days"

2. **Products**

   - Add/Edit/Delete Product

   - Image picker

   - Price, stock, categories

3. **Orders**

   - Customer name, order details, total

   - "Mark as delivered"

   - "Chat on WhatsApp"

4. **Customers**

   - Add new customer

   - Assign credit manually

   - Credit balance and payment tracking

5. **Settings**

   - Store name, logo, description

   - Payment status

   - Share store link

✅ **D. Super Admin Dashboard (PWA / Web App only)**

- Overview of vendors

- Trial expiry warnings

- Manual suspensions

- Email reports

✅ E. Extras

- **Error Screens** (404, connection lost, empty states)

- **Loading & skeleton UI**

- **Dark Mode (Optional MVP+)**

- **Push Notifications (Later)**

🔌 API & MOCK DATA NEEDED

You'll be working with:

- Supabase API for users, vendors, orders, payments

- MPesa STK Push flow via backend

- WhatsApp links via button generation

- Product + store image URLs (stored in Supabase Storage)

Create mock data with:

- 3 vendors, 10 products each

- Orders (2–5 per vendor)

- At least 1 credit customer per store

📊 PERFORMANCE GOALS

Metric	Target
Cold Start Time	< 3s

Image Load	Lazy + cached
Animations	Lottie: 60fps
Responsive Layout	Full on all phones down to 320px

💣 GOTCHAS TO AVOID

- Vendor A must NEVER see Vendor B's dashboard (RLS enforced)

- Don't store user sessions insecurely

- Offline mode must persist unsynced orders

- Ensure STK Push feedback is handled gracefully

🧩 FINAL TODOs FOR FRONTEND

- Define all routes + route guards

- Build static mockups before API hookup

- Integrate auth

- Add onboarding state logic

- Test every flow: login, trial, expired vendor, new customer order

- Test with Safari, Chrome, Edge (mobile & desktop)

UI DESIGNER DOCUMENTATION – KIOSERRA

This section is **for the UI/UX Designer**—we're building the most premium, sleek, and mobile-first digital shop & credit ecosystem for Kenya and Africa. The mission? Blend modern elegance with African utility. The final product must feel as intuitive as TikTok, and as impactful as M-Pesa.

🎯 OBJECTIVE

Design a fully responsive and adaptive UI system for **Kioserra**, focusing on:

- High usability and conversion for **vendors** (small biz owners)

- Effortless shopping flow for **customers** (urban + rural)

- Scalable, recognizable **branding**

- A front-end system ready for Flutter implementation

🎨 BRAND IDENTITY

Element	Value
Primary Color	`#0D47A1` (Royal Blue – trust, digital)
Accent Color	`#FF6F00` (Bright Orange – attention, CTA color)
Secondary	`#F4F4F4` (Background neutral)
Error/Alert	`#D32F2F` (Red for warnings, errors)
Success	`#388E3C` (Green for confirmations)
Fonts	`Poppins` (Web-safe, crisp, local-friendly)
Logo Style	Wordmark with subtle icon (Kiosk + Digital + Finance blend)
Theme	Light by default, dark optional MVP+

🖼️ ASSET STYLE & DESIGN RULES

- **Images from Unsplash** (business, people, products, local shops)

- **Icons**: Use `Feather`, `Fluent`, or custom SVGs (outline style, minimal)

- **Animations**: Use LottieFiles for loaders, empty states, success, etc.

- **Shadows**: Soft elevation (`box-shadow: 0px 4px 12px rgba(0,0,0,0.05)`)

- **Button Style**: Rounded 12px, bold, with hover effects for PWA/web

- **Input Fields**: Floating labels, error states, real-time validation

- **Cards**: Elevation 2dp, use for product tiles and order summaries

---

🧩 SCREEN-BY-SCREEN UI FLOWS

💡 1. Landing Page (`/`)

- Hero section with CTA: "Go Digital. Sell Smarter."

- Features:

    - Sell online instantly

    - Accept orders on WhatsApp

    - Credit system for your customers

    - Annual plan: Ksh 1500 only

- Screenshots or illustrations mock

- Contact: WhatsApp, Email, Phone

- Footer: Legal, Support, Socials

CTA Buttons:
`Start Now` → Register as Vendor
`Browse Stores` → /stores (public directory)

👤 2. Authentication Screens

Register Page

- Role picker (Customer | Vendor)

- Minimal form with validation

- Password strength indicator

- T&C checkbox

- Google Sign-in button

Login Page

- Email, password

- Forgot password

- OAuth options

Reset Password

- Email field, clean confirmation flow

🛍️ 3. Public Storefront (`/store/:slug`)

- Store banner: Logo, name, tagline

- Filterable product grid (by category or tags)

- Each product: Image, name, price, quick view

- WhatsApp Order Button → Login gate

- Mobile-first card UI

- Dynamic URL for sharing

🛒 4. Customer Authenticated Flow

- View your orders

- Chat with store via WhatsApp

- Track payment history (if on credit)

- Ability to rate store (future)

🧑‍💼 5. Vendor Dashboard (`/vendor`)

Sidebar Nav:

- Home
- Products
- Orders
- Customers
- Settings
- Subscription (MPesa modal)
- Logout

Home

- Metrics: Today's sales, orders, stock alerts
- Credit balance summary
- "Your trial ends in X days" banner
- Share Store CTA

Products

- Grid + List toggle view
- Add/Edit product modals
- Image uploads (compressed)
- Category tags

Orders

- Order list with status tags (Pending, Complete)

- "Chat on WhatsApp" button

- Filter by date/customer

Customers

- Table view: name, credit balance, last purchase

- Add new customer manually

- Track repayments

Settings

- Store info: name, description, logo, slug

- Payment/subscription settings

🧠 6. Super Admin (Web Only)

- Vendor list: Name, active status, trial remaining

- Manual action modals (Extend trial, Suspend)

- Subscription records + Payments

- System-wide alerts & data

💡 MICRO UX ELEMENTS

Element	Description
Empty states	"No orders yet" with icon and CTA
Skeleton loaders	Every section with loading state
Snackbar notifications	Top-right corner, success/warning
Modal confirmations	"Are you sure you want to delete this product?"

Transitions	Page transitions with `Fade + Slide`
Toasts	For lightweight alerts
Backdrop blur	On modals and pop-ups for aesthetic

📱 RESPONSIVENESS RULES

Screen Width	Layout
< 600px	1-column (mobile)
600px–1024px	2-column grid (tablet)
> 1024px	Full dashboard (web)

🧪 DESIGN DELIVERABLES

- Full Figma file (with all states, flows, and screens)

- UI Style Guide sheet (colors, fonts, icons, buttons)

- Reusable component library (for Flutter handoff)

- Landing screen mockup in mobile + desktop

- Store demo mockups

- Offline states, loading states, failed payment screens

- Dark Mode file (Optional MVP+)

🎁 BONUS UX NOTES

- Onboarding needs to feel like Canva + WhatsApp

- Use Kenyan cultural nuance in icons, images, and language

- Compress images at upload but maintain clarity

- All CTAs should use action verbs: "Start Selling", "Place Order", "Add Product"

- Use microanimations for feedback: Button presses, success, erro

BACKEND DEVELOPER DOCUMENTATION – KIOSERRA

This section is for the **Backend Developer** building out **Kioserra's engine**—from secure authentication to vendor stores, credit systems, orders, payments (M-Pesa STK), and dynamic subscriptions. The stack leans heavily on **Supabase**, **Node.js/Express (if needed)**, and **Daraja (M-Pesa)** for real-time financial flows.

We're building a **multi-tenant**, **role-based**, **real-time** SaaS product. Every line of logic must scale, stay secure, and **survive Kenya's peak hours**.

🧱 BACKEND STACK OVERVIEW

Tool/Platform	Purpose
Supabase	Auth, DB, Storage, Functions, RLS
Supabase Edge Functions	Custom backend logic (Node)
PostgreSQL	Structured, relational DB
Daraja API	M-Pesa STK Push & webhooks
Stripe (optional)	Card payments later (non-Kenya)
SendGrid/Mailgun	Transactional email (future)
Firebase Analytics (optional)	App-wide event logging
Redis/Queue (future)	Async credit checks, limits

🔐 AUTHENTICATION & USERS

Roles

- **Super Admin** – manages everything (DB access, vendors, payments)

- **Vendor Admin** – owns store, manages products, orders, customers

- **Vendor Staff** – limited access to dashboard (future)

- **Customer** – browses & places orders, can get product credit

Sign-up Flow

- Email/Password & Google OAuth

- On signup:

  - Set role claim

  - Create `profiles` record

  - For vendors → redirect to onboarding

  - For customers → redirect to `/stores`

Table: `users`

Column	Type
id (UUID)	Primary
email	Text
role	Enum
created_at	Timestamp
last_login	Timestamp

Use Supabase Auth with RLS policies by role + UID.

🏪 VENDOR STORES

Table: `stores`

Column	Type
id	UUID (PK)
name	Text

slug	Text (unique)
logo_url	Text
owner_id	UUID (FK → users.id)
trial_expiry	Timestamp
status	Enum: ['active', 'suspended']
whatsapp	Phone (e164)

Business rule: A vendor can only manage their own store.

Store Onboarding Flow

- After signup → `/vendor/onboarding`

- Input name, logo, description, type of store

- System generates slug

- Checks for uniqueness

- Saves trial_expiry = now() + 7 days

- Returns unique URL `/store/:slug`

📦 PRODUCTS

Table: `products`

Column	Type
id	UUID
name	Text
price	Decimal
image_url	Text
description	Text
store_id	UUID (FK)

category	Text
available	Boolean
created_at	Timestamp

Full CRUD via vendor dashboard.
RLS: only store owner can modify products.

📥 ORDERS

Table: `orders`

Column	Type
id	UUID
store_id	UUID
customer_id	UUID
product_ids	Array(UUID)
total_price	Decimal
payment_method	Enum
status	Enum
created_at	Timestamp

Flow

- When a customer orders:

  - Auth required

  - Creates an `order` record

  - Kicks off a WhatsApp message

  - Order status starts as `pending`

👤 CUSTOMERS (linked to vendors)

Table: `store_customers`

Column	Type
id	UUID
store_id	UUID
customer_email	Text
name	Text
credit_limit	Decimal
credit_used	Decimal
repayment_date	Timestamp

💳 SUBSCRIPTIONS

Table: `subscriptions`

Column	Type
id	UUID
store_id	UUID
start_date	Timestamp
end_date	Timestamp
amount_paid	Decimal
payment_method	Text

Payment Flow

- On login, check `trial_expiry < now()`

- Show modal if expired

- Collect phone number

- Trigger M-Pesa STK via Supabase Edge Function

- Listen to webhook

- Mark subscription as active

📞 MPESA INTEGRATION (Daraja)

Flow

1. Collect phone number

2. Send request to Daraja STK

3. User approves

4. Webhook confirms

5. Match phone to `store`

6. Save transaction & activate subscription

Tables

- `payments`

- `subscriptions`

Use Supabase Functions to receive and verify STK payment webhooks.

🔐 RLS POLICIES EXAMPLES

sql
CopyEdit
```sql
-- Store access
CREATE POLICY "Vendor Only"
ON stores FOR SELECT USING (auth.uid() = owner_id);
```

```
-- Product CRUD
CREATE POLICY "Own Product"
ON products FOR ALL
USING (
  EXISTS (
    SELECT 1 FROM stores
    WHERE stores.id = products.store_id
    AND stores.owner_id = auth.uid()
  )
);

-- Orders: Customer sees only theirs
CREATE POLICY "Own Orders"
ON orders FOR SELECT USING (auth.uid() = customer_id);
```

🌐 SEO + ANALYTICS (Optional)

- Add metadata fields to `stores`

- Store visits count

- Future: add Firebase or PostHog analytics per store

📈 VENDOR METRICS (Admin Dashboard)

- Table: `vendor_metrics`

- Track:

    - Product count

    - Order count

    - Last login

    - Active vs Suspended

🔐 SECURITY + AUDIT

- Supabase + JWT-based role verification

- Enforce HTTPS (via deployment)

- Store logs of all payments

- Rate-limit M-Pesa retries

- Add DB triggers for audit (optional)

💡 FUTURE PHASES

Feature	Timeline
Vendor Staff Roles	MVP+ Week 2
Product Lending	MVP+ Week 3
SMS Order Confirmations	MVP+ Week 4
Payment via Cards	Month 2
API for POS integration	Month 3
Vendor Credit Scoring	Q4

1. Full Launch Timeline (Client-Facing)

Week	Dates	Milestones
1	Jun 30 – Jul 6	• Finalize branding, app structure, product scope • Deliver wireframes and UI samples • Confirm project plan with all stakeholders
2	Jul 7 – Jul 13	• Build complete UI/UX frontend (Flutter) • Begin backend dev (Supabase/Auth/DB) • Implement responsive design

3	Jul 14 – Jul 20	• Complete all business logic (orders, payments, subscriptions) • Integrate M-Pesa STK + Auth flows • Admin/vendor dashboards v1
4	Jul 21 – Jul 27	• Full QA & bug fixing sprint • Add multilingual support • Beta tests with 10+ real vendors
5	Jul 28 – Aug 3	• Final polish: SEO, mobile testing, WhatsApp ordering • Deploy alpha to Netlify or Firebase • Start social/email launch pre-campaign
6	Aug 4 – Aug 10	• Official public launch • Start paid marketing + onboarding blitz • Setup support/monitoring pipelines

2. Backend Dev Timeline

Week	Backend Tasks
1	• Set up Supabase instance • Define DB schema (Users, Stores, Orders, Products)
2	• Implement Auth (Email, Google) • Role-based access control policies
3	• Create payment integration: M-Pesa STK Push + webhook listener
4	• Build secure API layer (Bolt or Next.js API routes) • Subscription enforcement logic
5	• Integrate dashboard analytics (revenue, traffic, trial status) • Ensure data backups and RLS policies
6	• Deploy final backend • Monitoring + fallback policies (cron, backups)

3. UI/UX Design & Frontend Dev Timeline (You)

Week	Frontend Tasks
1	• Create visual brand system (colors, fonts, component styles) • Build Figma or Flutterflow high-fidelity designs

2	• Implement Landing Page, Auth screens, Store Front (Flutter)
3	• Vendor dashboard: Product CRUD, Orders, Store Settings • Customer flow: Sign-up → Store → WhatsApp order
4	• Finish Admin dashboard UI, trial modal, payment screen
5	• Integrate animations, polish UI, load state indicators, offline UX
6	• Run usability tests • Apply feedback from beta users

---

4. User Journeys & App Flows (UX Designer/Dev Team)

Flow	Description
Onboarding	User lands → Chooses Vendor/Customer → Signs up → For Vendors: Setup Store in 3 Steps
Vendor	Dashboard → Add Products → Share Store → Monitor Orders & Payments
Customer	Visit Store → Browse Products → Place Order via WhatsApp (Login required)
Admin	Manage vendors → View subscriptions → Add trial extensions → Handle payments
Payments	Vendor logs in → Trial expired? Show paywall modal → M-Pesa STK push → Update subscription
Order System	Customer places order → Vendor receives via WhatsApp → Confirms manually or auto