

# COOP Agri — Technical Asset Description

Prepared for Financing Purposes | June 2026

## 1. Overview

**COOP Agri** is a purpose-built agricultural cooperative management platform deployed at [coopagri.org](https://coopagri.org). It consists of two distinct digital assets:

1. **The Public Website** — a bilingual marketing and information site for cooperative members and the public, with integrated online shops.
2. **The Farm Management Application** — a secure, role-controlled web application managing the full agricultural value chain from farm setup through to product sale and financial reporting.

Both assets are live, operational, and actively developed.

## 2. Asset 1 — The Public Website

**URL:** <https://coopagri.org> (unauthenticated)

### Description

A mobile-first, bilingual (English and Portuguese / Mozambique) public-facing website presenting COOP Agri's mission, model, and platform capabilities to prospective members, partners, and investors. The site also hosts two fully operational public webshops accessible without login.

### Key Features

- Fully responsive, mobile-first layout with smooth-scroll navigation
- Bilingual content (EN / pt-MZ) with client-side language switching and persistent preference
- Sections: Hero, Trust Strip, Platform Introduction, In-Practice Demonstrations, Platform Scope, Why It Matters, Community Model, Call to Action, Footer
- Progressive Web App (PWA) manifest — installable on mobile devices as a native-feeling app
- Service worker with offline caching strategy
- SEO-ready with `robots.txt` and structured content hierarchy
- Earth-tone brand identity consistent with an agricultural cooperative
- **Two integrated public webshops** (Farm Shop and Live Animals Shop) accessible from the homepage

### Technical Specification

Item	Detail
Framework	Python Dash 4.2.0 (Flask-backed)
Languages	Python, JavaScript, CSS

i18n	Client-side JS with <code>localStorage</code> persistence
Hosting	gunicorn WSGI (1 worker, 4 threads; worker and thread counts are a single-line config change — the <code>post_fork</code> DB-connection hook ensures safe multi-worker operation at any scale)
PWA	Service worker + Web App Manifest
Style	Custom CSS ( <code>landing.css</code> ), mobile-responsive

## Replacement Cost Estimate

A comparable bilingual PWA marketing site with integrated webshops, professionally developed, would cost **USD 6,000 – 12,000** to reproduce at regional rates, and **USD 18,000 – 30,000** at European/US agency rates.

---

## 3. Asset 2 — The Farm Management Application

**Access:** Authenticated users only; login at <https://coopagri.org/login>

### Description

A comprehensive, multi-farm agricultural operations platform covering the full cooperative value chain. The application manages everything from farm and unit registration, daily production logging, feed management, breeding, health events, slaughter, processing, sales, logistics, workforce, and reporting — all within a single integrated system with role-based access control and full product traceability. It also powers two customer-gated webshops accessible only to registered cooperative customers via the **Mercado portal** ([coopagri.org/mercado](https://coopagri.org/mercado)).

### Architectural Highlights

- **63 relational database models** backed by PostgreSQL
- **~45,000 lines** of application logic across the core application, six specialised callback modules, Mercado customer portal, and supporting services
- **24 distinct application routes** — 7 Mercado customer portal, 4 public/landing, and 13 authenticated staff management modules
- **Additive database migration system** — zero-downtime schema evolution on every restart
- **IoT integration** — live temperature, humidity, and precipitation monitoring via Tuya Cloud API (polling every 5 minutes), with rain sensor cumulative tracking and farm weather history charts
- **Mercado customer portal** (`/mercado`) — a fully operational, standalone Flask Blueprint (~1,200 lines) handling customer registration, login, profile maintenance, and password reset; separate session namespace from staff authentication; bcrypt password hashing; 5-attempt rate limiter with 5-minute lockout; bilingual (PT/EN) with `localStorage` language preference; GPS delivery-location capture at registration and profile edit
- **Two customer-gated webshops** — `/agri` (Farm Shop) and `/animal` (Live Animals) — accessible only after Mercado login; three-modal cart flow with real-time 30-minute cart reservation system that

deducts stock on add-to-cart and restores it on abandonment

- **PWA-capable** — installable on field workers' mobile devices, works on low-bandwidth connections with a background sync API for offline egg recording
  - **Full product traceability** — every animal batch, egg tray, and processed product carries a traceable content ID from origin to sale
  - **Functional feed categorisation** — feed brands are tagged with functional types (Starter Feed, Grower Feed, Finisher Feed, Layer Mash, etc.) that group and label feeds across batch scheduling dropdowns; renaming a brand cascades automatically to all historical records
  - **Server-side authorization** — every data-changing action enforces the caller's role and farm assignment on the server before executing. Direct API and Dash callback routes are gated by role permissions, so access control cannot be bypassed by hiding or forging UI controls
- 

## 3.1 Functional Modules

### Farm Management

- Register and manage multiple farms with GPS coordinates, map display, manager contact details
- Farm-level environmental monitoring: weather history with temperature, humidity, and precipitation charts
- Live weather status badges (Normal / Heat Stress / Extreme Heat) with IoT sensor refresh
- Rain sensor integration with cumulative rainfall delta display per polling window

### Production Units

- Production and Breeding Unit registration per farm with species, breed, capacity, and facility type
- Species-aware workflows: Poultry (Chicken, Duck, Turkey, Quail, Goose), Mammals (Rabbit, Goat, Sheep, Pig, Cow), Aquaculture (Fish, Shrimp)
- Stage-based lifecycle management: Incubating → Hatching → Brooding → Growing → Cage/Coop (poultry); Paired → Nursing → Delivered (mammal)
- Batch transfer between units with full data inheritance (breed, feed requirements, birth date, stage)

### Daily Production Logging

- Egg production recording with weight grading (S/M/L/XL), optimised for rapid mobile entry using `inputmode="decimal"` with auto-format (e.g., 663 → 66.3 g)
- Background sync queue for offline egg recording — retries automatically when connectivity returns
- Daily output logging for meat, produce, and eggs
- Feed consumption logging against scheduled requirements

### Feed & Supplies Management

- Feed stock inventory with purchase records, supplier tracking, and stock adjustments
- **Functional feed type tagging** — each feed brand is tagged with one or more functional categories (Starter Feed, Grower Feed, Finisher Feed, Layer Mash, Pre-Breeder Feed, Breeder Feed, Supplement); batch scheduling dropdowns display feeds as "Grower Feed — Brand Name" for unambiguous selection

- **Brand name cascade** — renaming a supplies type automatically propagates to all batch feed requirements, actual feed logs, purchase records, and distribution records
- Per-batch feed requirements (daily kg per head, total consumption projection)
- Stock overview, consumption projections, daily requirement calculations, transaction history
- Supplies cost reporting

### Breeding Registration

- Active batch cards per breeding unit showing Newborn Registration, Health Events, and Loss Reporting
- Mammal-specific newborn registration panel (triggered by "paired" stage)
- Pairing date, expected delivery date, gestation seed days by species (Rabbit 30d, Goat/Sheep 150d, Pig 114d, Cow 280d)
- Breeding parent registry

### Health & Loss Management

- Health event recording per batch (routine checks, treatments, disease events) with condition, product used, animals affected
- Loss reporting with cause classification (natural, disease, accident, predation, other) per production and breeding batch
- **Dedicated Batch Losses page** — consolidated loss and mortality reporting across all active batches
- Health history view per batch (last 3 events)

### Slaughter Management

- Slaughter record registration linked to breeding unit batches
- View and filter slaughter history

### Product Processing

- **Dedicated Processing module** (`/product-processing`) — converts Produce stock into refined outputs
- Processing batch management with inputs and outputs, processing output recording by product type

### Sales — Staff

- **Farm Shop** — produce and egg sales with customer linkage, unit pricing, quantity, and date
- **Live Animal Sales** — batch-based live animal sales from eligible breeding unit stages with age-class validation and weight-class pricing
- Customer management with contact and location details
- Market rate reference table for pricing guidance
- Sales trend reporting

### Customer Portal — Mercado (`/mercado`)

A standalone, fully operational Flask Blueprint serving cooperative customers. Separate session from the staff application. Routes:

Route	Function
<code>/mercado</code>	Landing dashboard — shows shops when logged in, sign-in/register when not
<code>/mercado/entrar</code>	Customer login with bcrypt verification and rate limiting
<code>/mercado/registrar</code>	Customer registration — name, email, phone, delivery address, GPS location, preferred language; sends welcome email
<code>/mercado/perfil</code>	Profile and detail maintenance — update name, phone, address, GPS, language preference, password; sends profile-update confirmation email
<code>/mercado/esqueci-senha</code>	Password reset request — email enumeration-safe (always redirects)
<code>/mercado/redefinir-senha/&lt;token&gt;</code>	Token-based password reset — 1-hour expiry, secure random token
<code>/mercado/sair</code>	Customer logout and session clear

**Account & input safety:** all customer-supplied values re-rendered into form pages are HTML-escaped to prevent cross-site scripting (XSS); registration rejects any already-registered email — including staff-created customer records that have no password yet — closing the pre-account-takeover gap.

### Transactional Email Service (`email_service.py`)

Zoho SMTP integration routing all customer emails through the `coopagri.org` domain: - **Welcome email** — sent on successful registration; language matches customer's chosen preference (PT or EN) - **Profile update confirmation** — sent whenever customer saves changes to their profile - **Password reset link** — bilingual, 1-hour expiry, includes safety notice if not requested - All emails BCC'd to the cooperative administrator; sent asynchronously (non-blocking background threads) - Inline logo embedding via base64 PNG for email client compatibility

### Webshops

- **Farm Shop** (`/agri`) — registered-customer product grid with per-batch stock selection, product images, weight-class pricing, and full cart flow (browse → details → cart → checkout)
- **Live Animals Shop** (`/animal`) — live animal listings sourced automatically from active breeding unit contents; age-class pricing validation at checkout
- **Cart reservation system** — stock deducted and a 30-minute hold placed the moment an item enters the cart; abandoned carts automatically restored on the next page load. Each reservation is bound to the browser session that created it via a per-session nonce, so a reservation can only be released or checked out by its owner
- **Tamper-resistant checkout** — product names, quantities, and prices are derived entirely server-side at checkout (from the stock record and the pricing table); client-supplied prices and cart groupings are ignored, preventing under-pricing or cross-product manipulation

- **Staff-verified payments** — orders are recorded as *pending verification* rather than auto-marked paid; a staff member must confirm payment before an order is treated as settled
- Bilingual shop interface (EN / pt-MZ) via a dedicated translations API endpoint

### Deliveries & Logistics

- Supplies distribution planning and execution
- Farm delivery management
- Order collection and delivery tracking with exportable history (Excel-compatible)
- **Driver routing** — optimised route planning for delivery drivers with stop management

### Workforce & Equipment

- Worker registration and hour logging with work history
- Equipment inventory, rental management (with rental rates and history), and equipment sales
- Environmental sensor readings (manual and automated via Tuya IoT)

### Reports & Analytics (printable)

- **Egg Production Report** — by farm, period, and grade
- **Meat Production Report** — slaughter volumes and weight
- **Feeding Report** — feed consumed vs. scheduled, variance analysis
- **Sales Trends** — revenue by product category and period
- **Supplies Costs** — expenditure tracking by supplies type
- All reports printable via popup print window (clean output, no app chrome)

### User & Role Management

- Role-based access control with per-page permission assignment
- Farm-scoped access — users see only their assigned farms
- Admin-only access to data management, backups, and user administration

### Data Management (Admin)

- Species and species group configuration
- Breeding parent registry
- Product type configuration with shop availability control (Farm Shop / Live Animals / Not for sale)
- Product images upload and management
- **Supplies type configuration** with functional feed type tagging
- Product categories, product pricing by weight class
- Supplier registry
- Payment methods configuration
- Shop settings management (shop images, opening status, display text)
- Market rate management
- Removal reason configuration
- Database backup and restore (SQL and Excel formats)

## 3.2 Technical Specification

Item	Detail
Framework	Python Dash 4.2.0 + Dash Bootstrap Components
Backend	Flask (embedded in Dash), gunicorn WSGI
Database	PostgreSQL (SQLAlchemy ORM, 63 models)
Deployment	gunicorn: 1 worker, 4 threads, <code>preload_app=True</code> ; <code>post_fork</code> hook safely disposes and re-initialises DB connections per worker — worker and thread counts are adjustable on the fly via a single config line with no code changes required
IoT	Tuya Cloud API (EU region), 5-minute polling; temperature, humidity, rain sensor
PWA	Service worker with versioned cache, Web App Manifest
Auth	Session-based with bcrypt password hashing; server-side role/farm-scoped write authorization enforced on direct API and Dash callback routes (not reliant on UI hiding)
Mobile	<code>inputmode="decimal"</code> optimised entry; background sync API for offline recording
Customer shops	2 customer-gated webshops (Farm Shop + Animal Market) with real-time cart reservation and bilingual UI
Codebase size	~22,400-line core application + 6 specialised callback modules (~15,900 lines) + Mercado portal (~1,200 lines) + shop, database, IoT, email, landing (~5,500 lines) = <b>~45,000 lines total</b>
Database models	63 ORM models
Application routes	24 routes — 7 Mercado customer portal, 4 public/landing, 13 authenticated staff management modules
API endpoints	~15 Flask REST endpoints (egg recording, feed log, daily output, export, IoT refresh, shop translations, backup, driver routing)

Email service	Zoho SMTP — 3 transactional email types, bilingual, async, BCC-to-admin
---------------	---

### 3.3 Replacement Cost Estimate

The following is a conservative professional estimate based on current scope:

Component	Hours (Est.)	Rate Cost
Database design (63 models, migrations)	120	\$667,200
Core application & routing	150	\$669,000
Farm, PU, BU management modules	160	\$669,600
Production logging & feed management	120	\$667,200
Feed type tagging & brand-function linking	20	\$661,200
Sales, deliveries & logistics	120	\$667,200
Driver routing system	40	\$662,400
Mercado customer portal (registration, login, profile, password reset)	80	\$664,800
Transactional email service (Zoho SMTP, bilingual templates, async)	30	\$661,800
Customer-gated webshops (2 shops, cart reservation, i18n)	140	\$668,400
Breeding, slaughter & processing	100	\$666,000
Health, loss, batch losses & workforce	100	\$666,000
Reports & print system	70	\$664,200
IoT / Tuya integration (sensors, rain, weather)	60	\$663,600

PWA, offline sync, mobile optimisation	40	\$602,400
Role-based access control & auth	40	\$602,400
Testing, deployment & configuration	80	\$604,800
<b>Total</b>	<b>1,470</b>	<b>\$88,200</b>

Rate of USD 60/hour reflects a mid-market Southern African development rate. At European/US agency rates (USD 120–150/hour), the equivalent replacement cost would be **USD 176,000 – 221,000**.

## 4. Combined Asset Summary

Asset	Replacement Cost (Regional)	Replacement Cost (International)
Public Website + Shops	USD 6,000 – 12,000	USD 18,000 – 30,000
Farm Management Application	USD 85,000 – 95,000	USD 176,000 – 221,000
<b>Combined</b>	<b>USD 91,000 – 107,000</b>	<b>USD 194,000 – 251,000</b>

## 5. Additional Value Factors

The following factors support a valuation above pure replacement cost:

- **Live and operational** — the platform is in active production use, not a prototype or proof of concept. It manages real farm data, real transactions, and real users.
- **Ongoing development** — the codebase is under active maintenance and feature development, demonstrated by a continuous change history. Between May and June 2026 alone, significant new capabilities were added: customer-gated webshops with Mercado portal, cart reservation, product processing, batch losses, driver routing, and functional feed categorisation.
- **Domain specificity** — the system encodes agricultural cooperative workflows (species-aware breeding, feed deduction scheduling, cooperative traceability, age-class live animal pricing) that are not available in generic farm management software. This specificity represents accumulated domain knowledge with real barriers to reproduction.
- **IoT-ready infrastructure** — live sensor integration (Tuya Cloud) covering temperature, humidity, and precipitation positions the platform for expansion into precision agriculture without rebuilding the data layer.
- **Controlled commerce layer** — the two webshops, gated behind the fully operational Mercado customer portal (registration, login, profile management, password reset, bilingual transactional emails via coopagri.org), transform the platform from an internal management tool into a member-only commerce

system, opening direct-to-member revenue channels while maintaining cooperative membership control.

- **Scalability** — the multi-farm, multi-role architecture supports onboarding additional cooperatives or members with no structural changes required.
- **PWA capability** — field-ready on low-bandwidth mobile connections, reducing the barrier to adoption in rural agricultural settings.
- **Full product traceability** — every unit of produce, from breeding batch to sale, carries a traceable content ID. This is a compliance and quality assurance asset with increasing regulatory relevance.
- **Security & data integrity** — the platform enforces authorization on the server for every state-changing action (role- and farm-scoped), protects customer-facing forms against cross-site scripting, hardens the public webshop checkout against price and cart tampering, and keeps third-party dependencies actively patched against known vulnerabilities. This reduces operational and reputational risk for a system that handles member, financial, and operational data.

---

*Document prepared from codebase and system analysis. Replacement cost estimates are indicative and based on scope analysis, not a formal software audit. An independent technical assessor can verify scope against the live system and source code on request.*