

☒ CropGenius - Product Requirements Document (PRD)

1. ☒ Project Overview

- **Project Goal:** An AI-powered, mobile-first precision agriculture platform for smallholder farmers in Africa.
- **Core Mission:** To provide real-time intelligence on crop disease, weather, and markets, making advanced agronomy accessible to everyone.
- **Tech Stack:**
 - **Frontend:** React 18 (Vite), TypeScript, Tailwind CSS, Framer Motion
 - **Backend:** Supabase (PostgreSQL, Auth, Edge Functions)
 - **APIs:** PlantNet, Google Gemini, OpenWeatherMap, Sentinel Hub

2. ☒ Global Requirements

2.1 Authentication & Authorization

- **Test Accounts:**
 - **Farmer User:** farmer@cropgenius.app / password123
 -

Admin User: admin@cropgenius.app / password123

-

Access Control:

-

The /mission-control and /admin dashboards are accessible **only** to users with the admin role.

-

Unauthenticated users attempting to access any protected route must be redirected to the /login page.

-

Standard users (farmers) attempting to access admin routes must be shown a "Permission Denied" page.

2.2 Global UI Elements

-

Navigation: The main navigation bar must be present on all pages post-login. It must include links to "Dashboard", "Mission Control" (admin only), and a "Logout" button.

-

Error Handling: All API fetch failures must trigger a standardized toast notification at the top-right of the screen with the message: Error: Unable to fetch data. Please check your connection and try again.

-

Loading States: All data-fetching operations must display a centrally-located skeleton loader matching the shape of the content being loaded (e.g., a list of cards, a chart).

3. 📁 Feature Breakdown & UI Specifications

3.1 Onboarding Wizard (/onboarding)

-

User Story: As a new farmer, I want a guided setup process to input my farm details so the platform can provide personalized intelligence.

-

Functional Requirements:

- 1.

The wizard consists of 6 sequential steps. The "Next" button is disabled until the current step's required fields are filled.

- 2.

Step 1 (Farm Vitals): Must use a map interface (e.g., Leaflet) for location input.

- 3.

Step 6 (Genius Plan): Must display a loading spinner for at least 3 seconds (simulating AI processing) before showing the final recommendations.

-

UI & UX Specifications:

-

A progress bar at the top must accurately reflect the user's position (e.g., "Step 1 of 6").

-

All input fields must have a 2px green border when valid and a 2px red border when invalid.

-

Validation error messages must appear directly below the invalid field in red text.

3.2 Mobile Home Dashboard (/dashboard)

-

User Story: As a farmer, I want a one-glance summary of my farm's health and key tasks so I can make quick, informed decisions.

-

Functional Requirements:

1.

The **Health Orb**'s color must directly correspond to the farm's health score (Green: >0.8, Yellow: 0.5-0.8, Red: <0.5).

2.

Clicking the "Quick Scan" button must open the device's camera.

3.

The **Weather Widget** must correctly display the current temperature and icons for the next 5 days.

-

UI & UX Specifications:

-

Health Orb: Must have a continuous, slow pulsing animation. Upon loading, it should animate from 0% to the current health score over 1.5 seconds with an 'ease-out' transition.

-

One-Finger Navigation: The expandable menu must open with a drag-down gesture and provide haptic feedback (if on a supported device).

-

Layout: On screens narrower than 480px, the dashboard must be a single-column layout.

3.3 Crop Disease Detection Flow

-

User Story: As a farmer, I want to upload a photo of a sick plant and get an immediate, actionable diagnosis and treatment plan.

-

Functional Requirements:

1.

User uploads a .jpg or .png image.

2.

A loading modal with the text "Analyzing your crop..." must be displayed while the backend processes the image.

3.

The results page must show:

- Disease Name (string)
- Confidence Score (percentage, e.g., "98.5%")
- Severity (Low/Medium/High)
- A list of actionable treatment recommendations (both organic and inorganic).

•

UI & UX Specifications:

- The uploaded image preview must be displayed clearly at the top of the results page.
- The "Confidence Score" must be color-coded: Green for >95%, Yellow for 85-95%, Red for <85%.
- Each treatment recommendation must be an expandable accordion, revealing more details upon click.

3.4 Admin - Mission Control Dashboard (/mission-control)

- **User Story:** As an admin, I want to manage all users and view aggregated farm data to oversee the platform's operations.

- **Functional Requirements:**

1.
The dashboard must contain a table of all registered users.
2.
Admins must be able to search for users by name or email. The table must filter in real-time as the admin types.
3.
Each user row must have a "Delete" button. Clicking it must trigger a confirmation modal with the text: Are you sure you want to delete this user? This action cannot be undone.
4.
Confirming the deletion should remove the user from the table and show a success toast: User successfully deleted.

- **UI & UX Specifications:**

- The user table must support pagination, displaying 10 users per page.
- The table's columns (Name, Email, Role, Joined Date) must be sortable by clicking the column headers.
- The "Delete" button must be styled with a red background to indicate a destructive action.

use this document to generate highly specific tests, identify deviations with surgical precision, and even suggest code fixes that align perfectly with your defined requirements.