

# FarmTrack Project Documentation

Version: 1.0 | Generated: February 15, 2026

## 1. Project Overview

FarmTrack is a web-based farming management platform built with React and Firebase. It supports farm setup, crop lifecycle tracking, field monitoring, weather views, user administration, and report export. A backend Express service provides AI assistant responses using Gemini as primary provider with Ollama fallback.

## 2. Technology Stack

- Frontend: React 19, Vite 7, React Router, Recharts, React Leaflet.
- Backend API: Node.js + Express.
- Database and Auth: Firebase Firestore and Firebase Authentication.
- AI Integration: Gemini API (recommended) with optional Ollama local fallback.
- Utilities: PapaParse (CSV), jsPDF/html2canvas (PDF/reporting), ESLint.

## 3. Repository Structure

- src/components: Shared UI shell and assistant floating chat widget.
- src/features/auth: Login, signup, email verification, auth guards.
- src/features/farms: Farm creation and map/location features.
- src/features/crops: Dashboard, crop creation/details, monitoring, history, projection, analytics.
- src/features/reports: Report generation and export.
- src/features/users: User list, detail view, role/activation management.
- src/features/weather: Weather dashboards and alerts.
- server/index.js: Assistant API service.

## 4. Route Map

/: Landing page  
/login: Login page  
/signup: Signup page  
/verify-email: Email verification flow  
/account-pending: Approval pending page  
/app: Protected application shell and dashboard  
/app/farms/new: Add farm  
/app/crops/new: Add crop  
/app/crops/:id: Crop detail  
/app/monitoring: Monitoring hub  
/app/history/:type: History records by type  
/app/projection: Farm projection  
/app/fields/analytics: Field analytics  
/app/reports: Reports and export  
/app/weather: Weather page  
/app/users: Users management (admin UI visibility)  
/app/profile: Profile page

## 5. Data Model Summary

- users/{uid}: profile metadata including isAdmin and isActive flags.
- users/{uid}/fields/{farmId}: farm-level structure.
- users/{uid}/fields/{farmId}/seasons/{seasonId}: season and crop state.
- users/{uid}/notifications/{notificationId}: in-app notifications.
- Some components read both users and Users collections for compatibility.

## 6. Environment Configuration

Create or update .env at project root. Main variables:

GEMINI\_API\_KEY: Gemini API key (if set, Gemini is used)

GEMINI\_MODEL: Gemini model id, default gemini-2.5-flash

OLLAMA\_BASE\_URL: Ollama host for local fallback

OLLAMA\_MODEL: Ollama model name

PORT: Assistant API port (default in this project: 8793)

VITE\_ASSISTANT\_API\_URL: Optional frontend override when API is remote

## 7. Running the Project

- Install dependencies: npm install
- Start frontend only: npm run dev
- Start API only: npm run api
- Start both frontend and API: npm run dev:all
- Build production bundle: npm run build
- Preview production build: npm run preview

## 8. Assistant API Endpoints

GET /api/health: Status, provider, model, and timestamp

POST /api/assistant: Generates AI response from prompt + farm context payload

## 9. Security and Access Notes

- Authentication is required for /app routes.
- UI now gates Add Farm, Add Crop, and Users buttons by isAdmin status.
- Email verification is enforced before entering app shell.
- Signup sets isAdmin false and isActive false by default.
- Use Firestore Security Rules to enforce role permissions server-side.

## 10. Known Operational Notes

- Keep frontend and API running together for full assistant functionality.
- If Gemini is unavailable, API falls back to Ollama when configured.
- If role flags do not appear, verify user document exists in users collection.
- Large JS chunks appear in production build warning; consider code splitting.

## 11. Maintenance Checklist

- Review and rotate API keys periodically.
- Validate Firestore indexes and security rules after schema changes.
- Re-test signup, verification, admin role visibility, and report export after releases.
- Regenerate this PDF after major route, model, or environment changes.