MAT

Project Overview

MAT is an open-source web platform designed to help property owners, facility managers, and technicians in Africa manage and maintain their assets more efficiently. With AI-powered recommendations, timely reminders, and a clean user interface, MAT simplifies the way users track, service, and preserve the value of their equipment like cars, generators, and home appliances.

Backend Integration Overview

Section	Status	Frontend Needs
Authentication	✅ Working	Build JWT-based auth, token refresh, and logout
Assets & Logs	✅ Working	Build asset dashboard, CRUD UI, logs per asset
Reminders	✅ Working	Display & manage reminders, show notifications
AI Services	✅ Working	Show AI-generated maintenance schedules, recs

---

API Access & Setup

Item	Details
Backend URL	`http://localhost:3001/api`
API Docs (Swagger)	`http://localhost:3001/api-docs`
Auth Type	Bearer Token (JWT)
Port	The frontend should connect to port `3001`

Frontend Structure Plan

Framework: React.js
Styling: Tailwind CSS
Routing: React Router v6
Auth: JWT stored in localStorage
Forms: React Hook Form + Yup
State Mgmt: Context API or Redux (if needed)

USER FLOW

1. Landing Page
↓
2. Signup
↓
3. Login
↓
4. Onboarding
↓
5. Dashboard
↓
6. Asset Detail Page
↓
7. Notifications
↓
8. Settings
↓
9. Logout

Pages & Components (Minimum Viable)

1. Login & Signup

- Auth forms with validation
- Store JWT on login
- Refresh token mechanism
- Redirect on login success

2. Dashboard

- Show user assets
- Quick links to: “Add Asset”, “Upcoming Maintenance”
- Display unread reminders/notifications

3. Asset Management

- Add/Edit/View Asset
- View asset logs
- Button: “Generate AI Maintenance Schedule” (connect to AI endpoint)

4. Reminders & Notifications

- Display upcoming reminders
- Mark as read/delete reminder
- View WhatsApp/email toggle per user

5. AI Integration

- Connect to:
  - `/ai/maintenance-recommendation`
  - `/ai/extract-data`
  - `/ai/history-analysis`

- Display AI responses in clean card view

Protected Routes

Use a `PrivateRoute` component to wrap:

- `/dashboard`
- `/assets`
- `/notifications`
- `/settings`

JWT should be sent via: Authorization: Bearer <token>

Testing Checklist

- ✅ Login/Register works & persists
- ✅ All protected routes redirect if not authenticated
- ✅ Assets can be added, updated, and viewed
- ✅ Maintenance logs appear per asset
- ✅ Reminders show + mark-as-read works
- ✅ AI endpoints return valid data to the frontend