NDIS Management System – Guide

This document explains the structure of the NDIS management system repository, how the frontend and backend interact, and how to run the application locally. It is intended as a guide to understand the codebase and extend it further.

1. Repository Structure

The project is divided into two main parts:

- **frontend/** – React application (UI). Handles routing, pages, and user interaction.
- **backend/** – FastAPI application (API). Handles data, authentication, and database interaction.

2. Frontend (React)

- **src/App.tsx** – Defines application routes (/, /participants, /documents, /sil-homes, /login). Uses ProtectedRoute to ensure only authenticated users can access protected pages.
- **src/index.tsx** – Entry point. Renders App component into the DOM, wrapped with Router and AuthProvider.
- **src/components/common/Layout.tsx** – Main layout including sidebar navigation and top bar. Provides navigation links to different pages and displays the authenticated user.
- **src/pages/*** – Each page (Dashboard, Participants, Documents, SIL Homes, Login) defines a UI screen. Pages like Participants fetch data from backend APIs.
- **src/context/AuthContext.tsx** – Provides authentication state (logged in user, logout function) across the app.

3. Backend (FastAPI)

- **backend/app/main.py** – Entry point for FastAPI. Creates the FastAPI instance, registers routes, and applies middleware.
- **Endpoints**: /participants, /documents, /sil-homes, / (root). These return JSON data to the frontend.
- **CORS Middleware** – Configured to allow frontend (http://localhost:3000) to make API calls to backend (http://127.0.0.1:8000).
- **requirements.txt** – Lists dependencies including fastapi, uvicorn, sqlalchemy, psycopg2-binary, etc.

4. How Data Flows Between Frontend and Backend

When a user navigates to a page like Participants:

- React Router renders Participants.tsx page.
- Participants.tsx uses fetch('http://127.0.0.1:8000/participants') to call backend API.
- FastAPI backend receives the request at /participants, returns JSON data.
- React frontend updates state and displays the list of participants in the UI.

5. Running the Application

Follow these steps to run both frontend and backend locally:

- **Backend**:
- - cd backend
  - python -m venv venv
  - venv\Scripts\Activate.ps1 # (Windows PowerShell)
  - pip install -r requirements.txt
  - uvicorn app.main:app --reload
  - Backend runs at http://127.0.0.1:8000
- **Frontend**:
- - cd frontend
  - npm install
  - npm start
  - Frontend runs at http://localhost:3000

6. Key Takeaways

- Frontend (React) handles UI, routing, and calls backend via fetch.
- Backend (FastAPI) provides REST APIs that return JSON data.
- Authentication is handled by AuthContext and ProtectedRoute on frontend.
- CORS configuration is required to allow frontend to access backend.
- Both frontend and backend must be running for the app to function correctly.