Technical Interview - Admin Dashboard Assignment

1. Context

This technical interview is based on a real production system.
The tasks you will work on are real-life engineering tasks, not theoretical exercises.

You will be joining an existing project that includes:

- A public mobile/web application
- An internal **Admin Dashboard**
- A backend API serving multiple clients

Your responsibility during this interview will focus **exclusively on the Admin Dashboard**.

2. System Overview (High-Level)

Applications & Services

1. Public Mobile / Web App

- **Stack:** Flutter
- **Audience:** End users
- **Consumes:** Backend API

2. Admin Dashboard *(Your focus)*

- **Stack:** React (Bun runtime)
- **Users:**
  - Internal administrators
  - Event providers (organizers)
- **Purpose:**
  - Event management
  - Provider management
  - Platform administration

3. Backend API

- **Stack:** Python (Django), Supabase
- **Responsibilities:**
  - Endpoints for mobile/web app
  - Endpoints for admin dashboard

4. Event Processing Service

- **Stack:** Python, OpenAI API

- **Role:** Event categorization and enrichment

3. Scope of Work

You will be given:

- ✅ Access to the **Admin Dashboard codebase**
- ✅ Access to the **Backend API**
- ❌ No work required on the mobile app

Your work will be **mainly focused on the Admin Dashboard**.

You may ask **any questions at any time**.
During the interview window, we/you expect responses within **~5 minutes**.

4. Tasks (Real-Life Engineering Work)

1. Authentication Integration

Integrate **Firebase Authentication** into the Admin Dashboard:

- Email & password login
- Google Sign-In
- Protected routes
- Proper session handling

2. Codebase Cleanup & Organization

The current codebase is **not well organized**.

Your task:

- Refactor and clean the code
- Improve folder and file structure
- Remove unused or redundant code
- Improve readability and maintainability
- Apply reasonable architectural patterns (hooks, services, utils, etc.)

3. API Integration

Currently, the dashboard is using **Supabase directly**.

Your task:

- Remove direct Supabase usage
- Integrate the **Backend API**
- Create clean API abstraction layers
- Handle loading and basic error states

4. Deployment

Deploy the Admin Dashboard to a **VPS**:

- Production-ready build
- Publicly accessible URL
- Clean environment configuration

Hosting credentials will be provided after task completion.

5. Evaluation Criteria

You will be evaluated on:

- **Proactivity**: asking questions, anticipating issues
- **Task Completion**: working features, not partial work
- **Autonomy**: ability to move forward independently
- **Creativity & Judgment**: code structure, UX decisions, trade-offs

6. Interview Timeline

- **Interview window:**
  - **Today afternoon**
  - **Tomorrow**
- **Review meeting:**
  - **Saturday at 9:00 AM (Ethiopia Time)**

During the interview window:

- Short async communication
- Fast response time (~5 minutes)

7. Final Deliverables

- Deployed Admin Dashboard (URL)
- Updated repository
- Short summary of:
  - Technical decisions
  - Trade-offs
  - Any assumptions made