In-Person Meeting Notes App

Overview

Build a mobile app that records in-person meetings, continues recording in background, and delivers AI-generated transcripts via push notifications.

Example user flow:

```
"I'm about to start a client meeting. I tap record, put my phone in my pocket, and forget about it. 30 minutes later I stop the recording. A few minutes later I get a notification — my transcript and summary are ready."
```

What the app should do:

1. Record audio with one tap
2. Continue recording when app is backgrounded or screen locked
3. Upload audio when recording stops
4. Process audio → transcript + summary (backend)
5. Send push notification when ready
6. Deep link from notification to view the meeting

Required Components

You must implement the following. We're testing architecture, not polish.

Component	Purpose
Custom config plugin	Enable background audio recording on iOS and Android
Background recording	Audio capture that survives app backgrounding
Expo Router	File-based routing with deep link support
Supabase integration	Auth, storage, database with RLS
Push notifications	Notify when transcript ready, deep link to meeting
Python backend	Process audio, generate transcript/summary, trigger notification

Technical Requirements

Stack:

Requirement	Details
Framework	Expo SDK 54
Routing	Expo Router
Backend	Python (FastAPI)
Database	Supabase (Postgres + Storage)
Notifications	Expo Push Notifications

Config plugin must configure:

iOS:

- UIBackgroundModes → 'audio' in Info.plist
- AVAudioSession category for recording
- NSMicrophoneUsageDescription

Android:

- RECORD_AUDIO and FOREGROUND_SERVICE permissions
- Foreground service type for microphone
- Notification channel for foreground service

Project structure:

```
/app
  /(tabs) — tab navigation
    index.tsx — home/recording screen
    meetings.tsx — meetings list
  /meeting/[id].tsx — meeting detail
/plugins — custom config plugin
/backend — Python API
README.md — setup + design decisions
```

Backend endpoint:

```
POST /process-meeting
Input: { audio_url, meeting_id, push_token }
Process: Download → Transcribe (mock OK) → Summarize → Update DB → Send
notification
```

Deliverables

1. Public GitHub repository
2. README with: how to run locally, architecture decisions (1 page max), what you'd improve with more time
3. Screen recording showing: start recording → background app → stop → receive notification → view transcript

Evaluation Criteria

Criteria	Weight
Config plugin — Correctly configures native projects for background audio. Shows understanding of iOS/Android requirements.	25%
Background recording — Recording continues reliably when app backgrounded. Handles interruptions.	25%
Architecture — Clean separation of concerns. Easy to understand and extend.	20%
Notifications + deep linking — Push notification works, tapping opens correct meeting.	15%
Code quality — TypeScript, proper error handling, readable code.	10%
Product thinking — Would this actually work for someone recording meetings?	5%

Submission

What to include in your email:

- GitHub repo link
- Screen recording link (Loom, Google Drive, or in repo)

Good luck! We're excited to see what you build.