SyncScribe

Tech Stack

The project leverages a modern technology stack:

- **Frontend:** React (with React Router), Material-UI (MUI), React Hooks, localStorage, `react-big-calendar`, Google Calendar API (OAuth2), `date-fns`, `marked`.
- **Backend:** Node.js with Express.js, Firebase Authentication, Firestore, Google Gemini API.
- **Deployment:** Render for both frontend (static site) and backend (web service).

Key Features Implemented

Core functionalities I delivered include:

- **User Authentication:** Secure Google Sign-In via Firebase Authentication.
- **Real-Time Transcription:** Microphone audio captured in chunks, transcribed rapidly (1-2s) using Google Gemini API.
- **Chat with Transcript:** Interactive AI chat (Google Gemini) using the full meeting transcript.
- **Automated Summaries:** Concise, AI-generated meeting summaries available quickly (2-4s).
- **Google Calendar Integration:** Connection via OAuth2 to fetch events and create meetings from entries. (Note: Uses test user access pending full Google API verification).
- **Data Persistence Sharing:** Secure storage of meeting data in Firestore; shareable summary links.
- **Error Handling:** Robust mechanisms for auth, audio processing, and API connectivity.

Key Libraries for Audio Processing

My approach to in-browser audio handling centered on:

- **Web Audio API:** For capturing microphone input and real-time audio stream processing, including visualization data via `AnalyserNode`.
- **MediaRecorder API:** My primary tool for browser-based audio recording and chunking data into blobs for transcription.
- **ffmpeg.wasm**: Used for in-browser re-encoding of audio blobs to normalize format (e.g., WAV/MP3) and optimize quality for Gemini transcription.

Access Links

The deployed application and its source code can be accessed via the following links:

- **Application:** https://syncscribe.me
- **GitHub Repository:** github.com/harshbhati1/SyncScribe

For testing the Google Calendar integration, please contact the author to be added as a test user, as full Google API verification for the project is pending.

Development Insights and Learnings

This project significantly enhanced my skills in Express.js, React.js, and Material-UI. I achieved impressive speeds for transcriptions (1-2 seconds) and summaries (2-4 seconds) by optimizing audio processing. Overcoming complex audio data challenges, even while traveling, was a key learning experience. The app is on Render's free tier, so local performance is naturally faster. I consistently used best practices like incremental testing and clear commit messages. Debugging was streamlined with tools like React Dev Tools and server logs.

Project Structure

The codebase is organized into distinct client (frontend) and server (backend) directories:

```
SyncScribe/
+-- client/               # React frontend
|   +-- public/           # Static files (index.html, _redirects)
|   +-- src/
|   |   +-- components/   # React components
|   |   +-- contexts/     # React context providers
|   |   +-- services/     # API utilities
|   |   +-- App.js        # Main app and routes
|   |   +-- ...           # Other React files
|   +-- package.json      # Frontend dependencies
|   +-- ...               # Other config files
+-- server/               # Node.js/Express backend
|   +-- src/
|   |   +-- routes/       # Express route handlers
|   |   +-- middlewares/  # Express middlewares
|   |   +-- config/       # Config files (Firebase, Gemini)
|   |   +-- ...           # Other backend logic
|   +-- package.json      # Backend dependencies
|   +-- ...               # Other config files
+-- .gitignore
+-- README.md
```

System Architecture Overview

The application follows a client-server model. The user interacts with the React frontend in their browser. This frontend communicates with the Node.js backend, which in turn interfaces with various external services like Firebase (for authentication and database storage) and Google APIs (for calendar access and AI-driven features). The high-level interaction flow is depicted as:

```
+--------------------+    +--------------------+    +--------------------+
|                   |     |                   |     |                   |
|   User Browser    | <-----> |   Frontend (React)| <-----> |   Backend (Node)   |
|                   |     |                   |     |                   |
+--------------------+    +--------------------+    +--------------------+
      |                       |                             |
      |                       |                             |
      |      Google OAuth     |                             |
      |<--------------------------->|                             |
      |                       |                             |
      |                       |      Firestore/             |
      |                       |<---------------------------->|
      |                       |      Google APIs            |
      |                       |      (Gemini)               |
      |                       |                             |
      |                       |      Google Calendar        |
      |                       |<---------------------------->|
      |                       |                             |
```