Gandhar – Final Task Block: Agent UX

Role: Audio-Video AI + UX Flow

Goal: Build a smooth pipeline where agent responses can be heard and optionally lip-synced with an avatar on-screen. Ensure it's usable via frontend trigger (Rishabh) and extendable by Dnyaneshwari.

Task 1: Agent Audio Response Layer

- Build a TTS pipeline that converts agent replies (from Vedant's API) into audio.
- Format: MP3/OGG, use a free voice (Google, ElevenLabs lite, or coqui).
- Ensure session ID tagging or a temp UUID per reply.
- Save to Supabase (if needed) or stream directly.
- Expose a frontend call or emit event (onAgentReplyAudio()).

Depends on:

- Vedant: final /ask-agent and /memory APIs
- Rishabh: agent frontend trigger

Task 2: Avatar Lip Sync (Prototype Mode)

- Connect agent response audio with avatar mouth movement.
- Use any of:
    - Tachyon.js + R3F mouth animation (basic waveform-to-lip model),
    - Papagayo-style viseme mapping, or
    - Use open-source lip-sync AI (e.g., SadTalker if no heavy compute).
- Prototype a demo with one fixed face/avatar.

Goal: Prove that voice + facial sync is possible. Not full production polish.

Task 3: Assist Dnyaneshwari (Audio UI)

- Expose audio replay controls (play, pause, replay last line).
- Help wire basic UX: avatar appears, says reply, disappears.
- Make sure APIs are reusable and cleanly documented.

Final Deliverables

- TTS layer for all agent replies (callable from frontend)
- Avatar/lip demo with synced speech
- Support doc for Dnyaneshwari and Rishabh (how to call/play voice)