Problem Statement

Build a real-time, open-source, modular avatar system that can listen, think, speak, and live-stream with sub-second latency. The avatar must be fully local or self-hosted with no paid APIs. The design should be generic, configurable, and easily re pursuable (education, support bot, campus guide, helpdesk, etc.) with minimal changes.

Core capabilities

- Real-time audio I/O and streaming: Capture mic input, render synthesized speech/video, and broadcast to viewers with WebRTC or equivalent, targeting sub-second latency. Prefer Janus/Ant Media Server or similar open-source media servers; RTMP/HLS can be provided as fallback.
- Lip-synced talking head: Given either TTS audio or pre-recorded audio, animate a 2D face or 3D head with high-quality lip sync in real time; acceptable open-source options include Wav2Lip or MuseTalk. Support at least 24–30 FPS on a single consumer GPU.
- Offline or self-hostable speech stack: Open-source ASR (e.g., Whisper variants) and open-source TTS capable of low-latency streaming synthesis; must output phonemes/visemes or timestamps usable for lip sync. No paid cloud TTS.
- Reasoning/LLM: Use an open-source chat model (e.g., Llama-family via local inference). Allow plug-and-play to swap models and prompt templates. No paid APIs.
- Frontend avatar rendering: Provide two interchangeable frontends:
- 2D talking-head (image-driven) using lip-sync model output.
- Web 3D avatar (GLB/ReadyPlayerMe spec) with blendshape/viseme mapping driven by phoneme timings.
- Session orchestration: Real-time loop: mic audio → ASR → LLM → TTS (+phonemes) → lip-sync/3D visemes → stream to viewers. Design must support back-pressure and graceful degradation.

Deliverables

- Source code and Docker compose for all services; single command brings up the stack locally with GPU if available.
- Two reference avatars:
- 2D portrait image talking head pipeline.
- 3D GLB avatar pipeline with viseme mapping.
- Demo web app:
- One-to-one conversation view (caller + avatar) and viewer broadcast mode.
- Toggle between ASR→LLM→TTS loop and text-input mode.

- Documentation:
- Setup guides for Ubuntu with NVIDIA GPU, model downloads, and performance tips.
- Architecture diagram and module interfaces to enable reuse in other projects.
- Benchmark report:
- Latency per stage, FPS, VRAM/CPU usage for small/medium models, and scalability notes.

Acceptance criteria

- Fully functional local demo with: live mic input, real-time response, synchronized mouth movements, and WebRTC live playback with sub-second to near-real-time latency.
- No paid or proprietary APIs; all components must run from open-source projects with local inference.
- Easy retargeting: Changing the avatar (new image or GLB) and swapping the LLM or TTS must not require code changes beyond config edits.
- Documented deployment for CPU-only fallback and GPU-accelerated paths, with expected quality differences.

Suggested open-source building blocks (non-binding)

- Streaming: Janus Gateway or Ant Media Server Community for WebRTC; fallback RTMP ingest to server.
- Lip-sync: Wav2Lip, MuseTalk; optional CodeFormer/ESRGAN for quality.
- ASR: Whisper variants (local).
- TTS: Open-source TTS with phoneme/timestamp support or alignment workflow.
- 3D frontend: Three.js with ReadyPlayerMe-style GLB and viseme mapping.

This framing keeps it generic, reusable, and fully open-source, while supporting live streaming, lip-sync, and GPU acceleration for real-time performance.