Technical Summary: MentorX

Team ID: ***Upsk-250439***

Repository: https://github.com/pihu58/MentorX

1. Problem Statement

Large-scale skilling and education ecosystems rely heavily on thousands of mentors, yet evaluating teaching quality remains manual, subjective, and non-scalable. Human evaluators introduce bias, fatigue, and inconsistency, making it difficult to ensure standardized instruction across programs like Mission Upskill India.

MentorX addresses this by providing an **automated, multimodal, objective evaluation system** that analyzes teaching videos across three dimensions:

- **Content Quality (Text)**
- **Delivery & Communication (Audio)**
- **Engagement & Presence (Video)**

The goal is to deliver **real-time, explainable teaching scores** that institutions can trust at scale.

2. Approach & AI Components

MentorX uses an **AI-first, parallel-processing architecture** to convert raw video into quantifiable teaching metrics.We utilize “asyncio” to split the input video into three concurrent processing streams:

A. Semantic Pipeline for Content & Technical Depth

Tech: Whisper ASR + Llama 3–8B (Groq LPU)

- Converts audio into timestamped transcripts.
- Applies a "Pedagogical Expert" evaluator prompt.
- Outputs **deterministic JSON scores** (0–10) for clarity, correctness, structure, and concept coverage.
- Enforces schema validation to avoid LLM hallucinations.

B. Visual Pipeline for Engagement & Nonverbal Immediacy

Tech: MediaPipe (FaceMesh, Pose), OpenCV. Runs fully on CPU with lightweight pose estimation. Extracted features:

- **Eye Contact Ratio:** Landmark-vector alignment with camera axis.
- **Gesture Energy:** Variance in wrist landmark trajectories, indicating expressiveness and confidence.
- **Posture Openness:** Shoulder-width and body-angle heuristics.

C. Acoustic Pipeline for Vocal Delivery & Communication Style

Tech: Librosa + FFmpeg
Features extracted:

- **Pitch Variability:** Standard deviation of F0 to detect monotony.
- **Pacing (BPM):** Onset envelope detection; target optimal speech-BPM window defined empirically.
- **Pause Density:** Silence-threshold detection for clarity and rhythm.

3. System Architecture

The platform utilizes a decoupled microservices architecture to ensure low-latency inference on cloud free-tiers.

- **Frontend:** Next.js 14 application hosted on **Vercel** for real-time data visualization.
- **Backend:** Containerized FastAPI service hosted on **Render (Docker)** handling orchestration.
- **AI Inference:** Hybrid model using local CPU-optimized libraries (MediaPipe, Librosa) and high-performance cloud APIs (Groq LPU).

4. Challenges & Mitigations

1. Low VRAM on cloud instances

Local Llama-3 inference infeasible.
Mitigation: Offloaded to **Groq LPU**; reduced Docker footprint by 4GB.

2. File I/O Locking on Windows (`WinError 32`)

Parallel temp-file creation causing race conditions.
Mitigation: Unique UUID-based temp paths along with thread-safe cleanup with async delays.

3. Race conditions across 3 async pipelines

Mitigation: Dedicated per-job temp directories and await-safe pipeline orchestration.

5. RoadMap