

Mentor Scoring AI – Technical Summary (Hackathon Prototype)

1. Problem Statement

Teachers often lack structured feedback on their teaching sessions. Manual evaluation is slow, inconsistent, and subjective.

This system provides an automated method to evaluate a teacher's recorded video using AI — analyzing body posture, speech clarity, technical content, and delivery.

2. Approach and AI Components

A. Body-Language Analysis (Frontend)

- MediaPipe Pose, FaceMesh, and Hands run directly in the browser.
- Extracts:
 - Eye contact ratio
 - Posture angle
 - Gesture count
- No video data is uploaded for body metrics — only numbers.

B. Speech-to-Text (Backend)

- FFmpeg extracts audio from the uploaded video.
- Whisper.cpp converts audio to:
 - Clean transcript
 - SRT timestamps

C. Audio Metrics

From transcript + timestamps:

- Words per minute
- Pause detection
- Filler word detection
- Stability score
- Pace score

D. Rubric (Static or AI-generated)

- Each teaching role (e.g., Python Teacher) has a rubric.
- If missing, the system uses an LLM (Groq or Ollama) to auto-generate a rubric JSON.

E. LLM Scoring Engine

Input:

- Transcript
- SRT timings
- Audio metrics
- Body-language metrics
- Rubric

LLM provides:

- 0–5 score for each category
- Timestamp-based evidence
- Conceptual gaps
- Final weighted score
- 3-step improvement plan
- Corrected teaching version

F. Evidence Clip Generation

- Using FFmpeg, 3–6 second video clips are extracted from timestamps where issues occur.
- Clips are shown to the mentor as evidence.

G. Feedback Speech (Optional)

- Google TTS generates instructor-style audio feedback.
-

3. Technical Architecture

A. Frontend (Browser)

- HTML + JS single-page UI
- MediaPipe for local body metrics
- Upload video to backend
- Dashboard to view:
 - Scores
 - Evidence clips
 - Metrics
 - Improvement plan

B. Backend (Node.js + Express)

- Routes for:
 - create session

- upload video
- extract audio
- transcript
- save body metrics
- scoring
- Utils:
 - ffmpeg wrapper
 - whisper wrapper
 - audioMetrics
 - pauseDetector
 - rubricGenerator
 - clipExtractor

C. Database (MongoDB)

Stores:

- Session information
- Transcript
- Metrics
- Rubric
- Score report
- Clip paths
- Feedback audio

4. Challenges and Mitigations

Challenge 1: Large video file processing

Mitigation: Local FFmpeg + Whisper.cpp for fast execution; supports chunking.

Challenge 2: Whisper accuracy on noisy audio

Mitigation: Audio preprocessing + transcript cleaning.

Challenge 3: Ensuring LLM outputs valid JSON

Mitigation: Strong system prompt + JSON schema validation.

Challenge 4: Timestamp accuracy

Mitigation: Combined SRT timestamps + pause detection to cross-check segments.

5. Roadmap to Final Build

- Add login / instructor dashboard.
- Add batch evaluation (multiple videos per instructor).
- Move FFmpeg + Whisper jobs to a queue system.
- Deploy on cloud (Render + MongoDB Atlas).
- Add domain-specific rubrics (C, Python, Java, OS, DBMS).
- Add model fine-tuning for better timestamp detection.