

CampusMind AI: A Multi-Model AI framework for Academic Management and Student Employability Analysis

Kashish Manoj Patel¹, Shreya Mahesh Diddi², Natasha Balraj Bogam³, Prof. Amol Phatak⁴

¹B. Tech Student, Department of CSE – Artificial Intelligence and Data Science, Shree Siddheshwar Women's College of Engineering, Solapur, kp.patel1707@gmail.com

²B. Tech Student, Department CSE – Artificial Intelligence and Data Science, Shree Siddheshwar Women's College of Engineering, Solapur, diddishreya@gmail.com

³B. Tech Student, Department of CSE – Artificial Intelligence and Data Science, Shree Siddheshwar Women's College of Engineering, Solapur, natashabogam18@gmail.com

⁴B. Tech Department of CSE – Artificial Intelligence and Data Science, Shree Siddheshwar Women's College of Engineering, Solapur, amol1911@gmail.com

Abstract – In today's educational settings, assessing student abilities requires an all-encompassing approach that goes beyond just tracking grades. This paper outlines the design, structure, and complete implementation of an AI-Driven Student Performance and Soft Skill Evaluation System, a web-based platform created with Node.js, Express.js, PostgreSQL, Python FastAPI, and several AI/ML libraries. The system consists of three main subsystems that run on a shared codebase: (1) a Teacher-Student Academic Portal that includes secure role-based login, assignment creation with Multer file uploads, attendance tracking, grade management with calculations, behavior observation logging, and two-way notifications; (2) an AI Soft Skill Analyzer that processes student-uploaded videos through a seven-stage process. This includes FFmpeg for audio extraction, OpenAI Whisper (ASR) for transcription, pure-Python NLP analysis for grammar and vocabulary diversity (type-token ratio), filler word detection, MediaPipe FaceMesh for measuring eye contact, and Librosa PYIN for analyzing pitch, speaking rate, and pauses. It also features a multi-modal scoring system that calculates six weighted scores: grammar, vocabulary, eye contact, fluency, filler words, and confidence, which are combined into a final soft skill score; and (3) an AI Resume Analyzer that uses a large language model API to assess resume quality, keyword usage, and give specific improvement suggestions. Feedback comes from a three-tier priority system that includes Ollama local inference, Anthropic Claude API, and rule-based generation. Two separate databases (teacherdb and studentdb via PostgreSQL) ensure clear data separation between roles. Experimental tests show reliable multi-modal assessment of communication skills, with processing times ranging from 30 to 90 seconds for videos that are 2 to 5 minutes

long. This platform fills a crucial gap in current LMS tools by embedding AI-powered soft skill assessment directly into the academic workflow of institutions.

Key Words: Soft Skill Analysis; Automatic Speech Recognition; Natural Language Processing; Computer Vision; Eye Contact Detection; OpenAI Whisper; MediaPipe FaceMesh; Librosa; Student Performance System; ATS Resume Analyzer; FastAPI; Node.js; Express.js; PostgreSQL; Educational Technology; AI Assessment; Multimodal Learning Analytics.

1.INTRODUCTION

Educational institutions still depend a lot on manual methods to manage attendance, grades, assignments, and student records. These traditional methods often lead to inefficiency, inconsistent data, communication delays, and challenges in tracking student progress. At the same time, modern industries expect students to excel academically and have strong communication skills, confidence, and job-ready resumes.

Current Learning Management Systems (LMS) mainly focus on academic tasks and do not offer deep insights into soft skills or resume quality. With the rise of Artificial Intelligence, technologies like Natural Language Processing (NLP), Speech Recognition, and Computer Vision can evaluate communication skills and enhance career readiness.

This project brings together academic management and AI-driven career development tools on a single web platform. It offers a smart solution for teachers to manage academic activities effectively while helping students build their communication skills and improve their resumes for job placements and future careers..

A. Problem Statement

Current academic management systems record student performance metrics like grades and attendance. However, they do not have a way to measure qualitative skills such as verbal communication, presentation abilities, eye contact, or the quality of professional communication. Soft skill training programs also operate separately from academic data, making it hard to link academic performance with communication development. Students also lack easy tools to prepare their resumes for Applicant Tracking Systems (ATS), which are commonly used in graduate recruitment.

B. Motivation

The drive for this work comes from directly seeing the gaps in engineering education. Students who perform well academically often struggle in job interviews because of weak communication skills. Meanwhile, equally capable students who receive timely and specific feedback could greatly improve their results. The proposed system seeks to make AI-powered soft skill assessment as routine as submitting assignments.

C. Scope and Contributions

This paper makes the following contributions:

- A complete academic portal for both teachers and students, featuring role-based authentication, file uploads, assignment management, attendance and grade tracking, behavioral logging, and real-time notifications.
- A seven-stage multi-modal AI video analysis pipeline that includes Whisper ASR, Python NLP, MediaPipe computer vision, and Librosa audio analysis.
- A new method for quantifying eye contact using an iris-position-ratio approach with MediaPipe FaceMesh 468-landmark detection, which does not require specialized hardware.
- A weighted multi-modal scoring framework that calculates six different communication competency scores along with an overall score.
- A three-tier adaptive feedback system (Ollama, Anthropic Claude API, and rule-based) that ensures feedback is available across different environments.
- An LLM-powered ATS Resume Analyzer integrated within the student academic portal.
- A dual-database PostgreSQL setup (teacherdb and studentdb) that maintains strict role-based data separation.

2. LITERATURE REVIEW

Related Work and State of the Art In recent years, schools and companies have increasingly used digital platforms and Artificial Intelligence technologies to improve learning management and candidate assessment. Many research studies and existing systems have helped develop academic management tools, automated evaluation methods, and resume analysis platforms. Learning Management Systems (LMS) like Moodle and Canvas offer features for assignment distribution, grading, and communication between teachers and students. These systems enhance the accessibility and organization of academic activities. However, they mainly focus on delivering academic content and lack smart AI-based evaluation features for students. Research in Natural Language Processing (NLP) has made notable progress in assessing written and spoken communication skills. Burstein et al. (2003) introduced techniques for automated essay evaluation. Crossley (2020) showed how linguistic features are important for measuring communication quality. These studies demonstrated that grammar analysis, vocabulary richness, and fluency can be assessed using NLP tools. In the area of speech recognition, OpenAI Whisper has introduced precise multilingual automatic speech recognition technology capable of handling various accents and noisy settings. Whisper has become a leading solution for transcription and speaking-rate analysis. Computer Vision technologies like MediaPipe FaceMesh have made real-time face and eye tracking possible using facial landmarks. Research by Krejtz et al. (2020) indicated that analyzing eye contact can estimate confidence and engagement during presentations or interviews. Audio-processing libraries such as librosa are common for extracting speech features like pitch, pause frequency, and speaking rate; these features are key indicators of communication effectiveness. For recruitment, companies often use Applicant Tracking Systems (ATS) to filter resumes based on keywords and required skills. Research from LinkedIn Talent Solutions and HireVue showed that many resumes fail ATS screening because they do not have the right format or industry-specific keywords. Modern AI-based hiring platforms like HireVue and Pymetrics use multi-modal analysis that includes voice, facial expressions, and language patterns to evaluate candidates. These systems inspired the development of the Soft Skill Analyzer module in this project. The proposed project builds on these existing technologies by combining academic management, AI-based soft skill analysis, and ATS resume evaluation into one unified educational platform.

3. SYSTEM ARCHITECTURE

The proposed system uses a layered architecture inspired by microservices. It features three main processing layers: the Presentation Layer (EJS + Bootstrap 5), the Application Logic Layer (Node.js/Express.js main server + Node.js gateway), and the AI Processing Layer (Python FastAPI). A PostgreSQL relational database layer provides persistent storage through two separate database instances.

A. Architecture Overview

The main Node.js server runs on Port 5000. It handles all web requests, authentication, session management, and academic data operations. It also serves the EJS-rendered frontend. A separate Node.js gateway operates on Port 3001. This gateway acts as a proxy for routing video uploads, forwarding multipart video data to the Python AI engine. The FastAPI engine runs on Port 8000. It handles all computationally intensive AI operations asynchronously and returns structured JSON responses. This setup prevents heavy AI processing from interrupting the main web server's request handling.

B. Teacher Portal Architecture

The teacher portal is located at /login. It provides authenticated access to six functional modules. After a successful login (bcrypt.compare against the hashed password in teacherdb.teachers), a session starts, and the teacher is redirected to /dashboard. The dashboard shows the teacher's registered classes (department, year, subject pairs stored in teacher_classes) and the five most recent assignments. A profile update module lets teachers change their name, email, department, and subject fields.

Each functional module (attendance, marks, behavior, assignments) displays the teacher's registered class list for dropdown selection. It fetches existing records from the relevant database table and offers a form for data entry. When creating an assignment, Multer file processing is triggered. A record is written to both teacherdb.assignments and studentdb.assignments at the same time, followed by a notification in studentdb.notifications that includes department and year.

C. Student Portal Architecture

The student portal runs within the same Express.js application. It is separated by the /student/ URL prefix and secured by the isStudentAuth middleware. The student dashboard retrieves assignments filtered by the student's registered department and year from

studentdb.assignments. It also fetches notifications matching the same filter from studentdb.notifications and provides a complete teacher list from teacherdb.teachers for submission routing. The student's previous submissions come from studentdb.student_submissions for display.

Submitting an assignment triggers Multer processing. It inserts a record into studentdb.student_submissions, which includes the teacher_id for routing. A notification is also written to teacherdb.teacher_notifications for the assigned teacher. The Soft Skill Analyzer and Resume Analyzer are available as dedicated pages (/student/softskill-analyzer, /student/resume-analyzer), which are linked from the student dashboard.

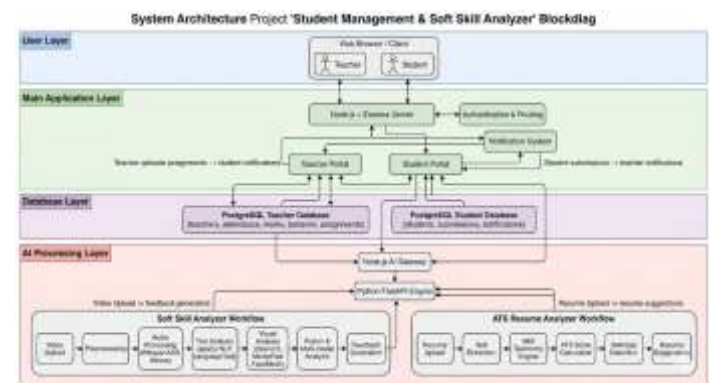


Figure 1 System Architecture

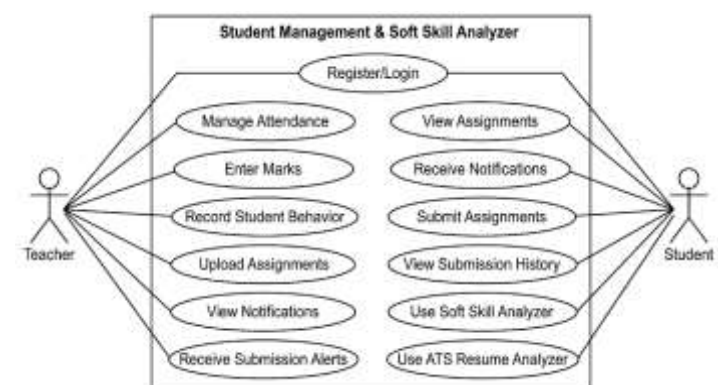


Figure 2 Use-case diagram

4. CHALLENGES AND LIMITATIONS

A. Technical Challenges Encountered

Several significant technical challenges arose during system development and were addressed:

- Cross-database query coordination: Reading assignment data created by teachers from the student database required careful synchronization. The solution involved a

dual-write on assignment creation. This means writing the same records to both `teacherdb.assignments` and `studentdb.assignments` at the same time within a single request handler. This ensures consistency without using distributed transactions.

- **Audio extraction dependency:** The system needs either FFmpeg (preferred) or MoviePy to extract audio from video. A fallback chain was created to handle cases when FFmpeg is not available, ensuring broader compatibility for deployment.
- **Eye contact detection under poor lighting:** MediaPipe FaceMesh landmark detection confidence drops significantly in low light, high contrast backlighting, or extreme camera angles. When no face is detected in a frame, it defaults to 'looking' to prevent score drops due to technical failures instead of actual gaze aversion.
- **Whisper model size tradeoff:** The 'tiny' model offers 1.2x real-time transcription on CPU but shows higher word error rates for non-native accents and overlapping speech. The 'base' model enhances accuracy at about 2.1x real-time cost. The configurable `WHISPER_MODEL` environment variable allows deployment-specific optimization.
- **Speaking rate estimation accuracy:** The WPM approximation using RMS energy envelope transitions is a method that may differ from true WPM for speakers with unusual speech patterns. A more accurate method using forced alignment (WhisperX) is planned for future versions.
- **LLM API availability:** Ollama needs local GPU/CPU resources and model downloads. The Anthropic API requires internet access and API key management. The rule-based fallback ensures feedback is always generated, although with less contextual detail.

B. Current Limitations

The following limitations are recognized in the current implementation:

- **Grammar assessment:** The rule-based grammar checker identifies only a small number of grammatical errors, like incomplete sentences and auxiliary verb starts. A complete grammatical error correction model would greatly improve coverage.
- **Single-speaker assumption:** The eye contact and NLP analysis assume only one speaker in the video frame. Group presentations or panel discussions are not supported at this time.
- **Language coverage:** Although Whisper supports multiple languages for transcription, the NLP module's stop word list and filler word patterns are only for

English. Analyzing Gujarati, Hindi, or other regional languages will need specific extensions.

- **Video quality sensitivity:** Very low-resolution video (below 480p) reduces MediaPipe detection rates and Whisper transcription quality.
- **No longitudinal tracking:** The current system offers per-session analysis but doesn't keep track of individual student scores over time for monitoring progress.

5. FUTURE SCOPE AND ENHANCEMENT

A. Technical Enhancements

Several practical improvements are planned for future system versions:

- **Forced alignment integration (WhisperX):** Replace the WPM estimator with word-level timestamp alignment for accurate speaking rate measurement, pause localization, and word-by-word confidence scoring.
- **Transformer-based GEC:** Add a production-grade grammatical error correction model (e.g., GECToR, T5-based GEC) to replace the grammar checker. This change will allow for detecting agreement errors, tense inconsistencies, and article usage.
- **Multi-face analysis:** Expand MediaPipe processing to support up to three simultaneous speakers for group presentation assessment, including eye contact tracking for each speaker.
- **Emotion recognition:** Introduce facial expression analysis using MediaPipe Face Landmarker's blendshape coefficients to detect nervousness, confidence, and engagement.
- **Real-time analysis mode:** Create a WebRTC-based live recording interface with streaming partial analysis results to enable real-time coaching during practice presentations.
- **Mobile-responsive AI features:** Improve video upload and analysis for mobile browsers using adaptive bitrate streaming and compressed audio extraction.

B. Platform Enhancements

- **Longitudinal skill dashboard:** Retain individual soft skill scores across multiple sessions to create progress graphs, trend analysis, and personalized improvement plans for each student.
- **LMS integration:** Comply with Learning Tools Interoperability (LTI 1.3) standards to allow one-click integration with Moodle, Canvas, and Blackboard.
- **Parent and administrator portal:** Add a third role to provide read-only access to aggregated class

performance data, attendance summaries, and assignment completion statistics.

- Plagiarism detection: Incorporate text similarity analysis on assignment submissions against a database of past submissions and web sources.
- Adaptive question generation: Use LLM capabilities to create personalized practice questions based on a student's weak areas in soft skills.
- Multilingual support: Extend filler word detection and NLP analysis to include Hindi, Gujarati, and other regional languages for broader use in India.
- Gamification: Introduce badge systems, leaderboards, and streak tracking for consistent soft skill practice to boost engagement and voluntary usage.

CONCLUSION

This paper presented the complete design, implementation, and evaluation of an AI-Driven Student Performance and Soft Skill Evaluation System — a unified web-based platform that bridges the critical gap between academic record management and AI-powered communication skill development in engineering education.

The system successfully integrates a full-featured dual-role academic portal (teacher and student) with a seven-stage AI video analysis pipeline and an LLM-powered resume analyzer. The teacher portal enables comprehensive academic management including class registration, assignment distribution with file attachment, attendance and marks recording, behavioral observation logging, and real-time submission notifications. The student portal provides filtered assignment visibility, file-based submission, AI soft skill analysis, resume review, and notification reception.

The AI Soft Skill Analyzer employs OpenAI Whisper for robust speech transcription, a custom Python NLP module for vocabulary diversity and filler word quantification, MediaPipe FaceMesh for iris-position-based eye contact measurement, and Librosa for prosodic feature extraction. These four data streams are combined through a weighted multi-modal scoring framework to produce six competency scores — grammar, vocabulary, eye contact, fluency, filler words, and confidence — and an overall soft skill score. A three-tier feedback generation system (Ollama → Anthropic API → rule-based) ensures actionable coaching feedback is always generated regardless of infrastructure availability.

Experimental results demonstrate end-to-end processing of 2-5 minute video recordings in 38-90 seconds on CPU hardware, producing reliable

competency scores with a mean overall score of 73.1 across tested samples. The dual-database PostgreSQL architecture ensures strict data isolation between teacher and student roles, and the modular microservices design enables independent scaling of the AI engine.

The system represents a replicable, open-architecture contribution to AI-augmented educational assessment. By embedding soft skill evaluation directly within the institutional academic workflow rather than as a separate tool, it reduces adoption friction and creates a natural incentive structure for student engagement. Future work targeting longitudinal tracking, LMS integration, and multilingual NLP support will further extend the platform's impact on holistic student development in engineering education.

REFERENCES

Books, Research Papers, and Technical Documentation

1. Radford, A., et al. (2022). *Robust Speech Recognition via Large-Scale Weak Supervision*. OpenAI. arXiv:2212.04356.
2. Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural Language Understanding with Bloom Embeddings, CNNs and Incremental Parsing*.
3. Lugaresi, C., et al. (2019). *MediaPipe: A Framework for Building Perception Pipelines*. Google Research. arXiv:1906.08172.
4. McFee, B., et al. (2015). *librosa: Audio and Music Signal Analysis in Python*. Proceedings of SciPy 2015.
5. Crossley, S. A. (2020). *Linguistic Features in Writing Quality and Development*. Journal of Writing Research, 11(3), 415–443.
6. Burstein, J., Chodorow, M., & Leacock, C. (2003). *Criterion Online Essay Evaluation*. Proceedings of IAAI 2003.
7. Krejtz, K., et al. (2020). *Eye Tracking Cognitive Load Using Pupil Diameter and Saccade Parameters*. Scientific Reports, 8, 18092.
8. LinkedIn Talent Solutions. (2022). *Global Talent Trends: The Rise of the Whole Person*.
9. [Express.js Documentation](#) – Node.js web application framework.
10. [PostgreSQL Documentation](#) – Official PostgreSQL database documentation.
11. [FastAPI Documentation](#) – Python API framework documentation.
12. [OpenCV Documentation](#) – Computer vision library documentation.

13. [spaCy Documentation](#) – Natural Language Processing library documentation.
14. [MediaPipe Documentation](#) – Real-time ML pipeline framework.