International Research Journal of Engineering and Technology(IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 04 | Apr 2026 www.irjet.net p-ISSN: 2395-0072

Autonomous AI Interview Platform: An Integrated Three-Phase Pipeline for Automated Recruitment

Prasanna Lakshmi N, *Assistant Professor, Department of CSE(AI&ML), R.V.R. & J.C. College of Engineering Guntur, Andhra Pradesh, India*
Gadde Dheeraj, *Student, Department of CSE(AI&ML), R.V.R. & J.C. College of Engineering Guntur, Andhra Pradesh, India*
Cheedella Mohit, *Student, Department of CSE(AI&ML), R.V.R. & J.C. College of Engineering Guntur, Andhra Pradesh, India*
Bachu Chandra Mouli, *Student, Department of CSE(AI&ML), R.V.R. & J.C. College of Engineering Guntur, Andhra Pradesh, India*

Abstract - *This paper describes an Autonomous AI Interview Platform built to reduce the high cost and manual effort involved in traditional HR screening. The system runs on a three-tier architecture—React 19, FastAPI, and a dedicated AI Engine—and automates three core tasks: resume screening, question generation, and answer evaluation. The screening phase uses NLP and Sentence-BERT to achieve 85% matching accuracy, while Cerebras LLMs produce interview questions tailored to each candidate's profile. Audio responses are transcribed by Groq-hosted Whisper STT with median latency below 600ms, enabling real-time scoring. A weighted composite model aggregates scores across resume quality, skills, and interview performance to generate decision bands for HR review. Interview integrity is enforced through OpenCV face detection and tab-switch logging. Under a load of 50 concurrent users, the system recorded a 0.02% error rate. The platform cuts HR screening effort by 70% and reduces per-session costs to approximately $0.15–$0.25. Future plans include cloud deployment on AWS and Render.*

Key Words: Artificial Intelligence (AI), Automated Recruitment, Large Language Models (LLM), Natural Language Processing (NLP), Speech to Text, FastAPI, Intelligent Proctoring, Sentence-BERT.

1. INTRODUCTION

Over the past decade, advances in artificial intelligence, machine learning, and natural language processing have reshaped how organizations find and evaluate talent [1]. Hiring at scale remains expensive and slow: according to SHRM, the average cost-per-hire in the United States exceeds $4,700, and positions stay open for 42 days on average [3]. These pressures have driven interest in automated recruitment tools that can screen and evaluate candidates without requiring proportional increases in recruiter time [2].

Applicant Tracking Systems (ATS), which date to the 1990s, were the first attempt at automating resume filtering. Early versions used rule-based keyword matching, and they had a well-known problem: qualified candidates whose resumes used non-standard formatting were often rejected outright [4][5]. Transformer-based language models have since made it possible to screen resumes by semantic meaning rather than surface-level keyword overlap, substantially improving the accuracy of automated screening [6].

NLP has become central to modern automated recruitment, allowing systems to extract meaning from unstructured documents such as resumes, cover letters, and job postings [7]. Sentence-BERT (SBERT), developed by Reimers and Gurevych (2019), generates dense sentence embeddings that support accurate semantic similarity computation between job descriptions and candidate profiles [8][9].

Large Language Models have broadened what automated recruitment can do. Models like GPT-4, LLaMA, and the Cerebras CS-3 can generate interview questions grounded in a candidate's profile, evaluate responses, and return structured scoring rationales [10]. Cerebras' inference hardware runs LLM requests fast enough for real-time interview interaction, which was not practical with earlier infrastructure [11].

Speech-to-text accuracy has improved substantially. OpenAI's Whisper model transcribes speech with near-human accuracy across a range of accents and recording conditions [12]. Groq's LPU infrastructure brings STT inference latency below one second, which makes voice-based interview evaluation practical at enterprise scale [13]. Together, Whisper's accuracy and Groq's speed make real-time spoken response evaluation feasible in production.

Despite these individual advances, few systems have combined resume screening, question generation, and answer evaluation into a single, deployable platform [14]. Caldera et al. (2023) demonstrated the concept with their Interview Bot, which used CNN-based emotion analysis and sentiment detection to achieve 70–80% accuracy on component tasks [15]. That work, however, did not address scalability or the cost efficiency needed for large-scale deployment.

Bias in AI-driven hiring is a growing area of concern [16]. Models trained on historical hiring data can perpetuate demographic disparities unless they are audited for fairness [17]. The EU AI Act (2024) and EEOC guidance on AI in employment now impose explicit requirements on automated hiring tools, making fairness auditing and explainable scoring mechanisms essential design considerations rather than optional enhancements [18][19].

Computer vision-based proctoring is well established in educational assessment [20], but recruitment introduces different constraints. Monitoring must be lightweight, privacy-respecting, and free from the continuous video recording that candidates typically find objectionable [21]. OpenCV provides the primitives needed to implement face detection and behavioral logging locally, without relying on third-party cloud vision services [22].

On the engineering side, FastAPI has become a practical choice for Python-based AI backends: it handles requests asynchronously, auto-generates OpenAPI documentation, and validates inputs through Pydantic [23]. React 19's concurrent rendering and Suspense-based data fetching complete the stack, enabling a responsive interface that keeps pace with the platform's real-time inference pipeline [24].

This paper presents an Autonomous AI Interview Platform that brings all three phases together in one deployable system. The contributions are: (1) a validated three-phase pipeline with documented performance metrics; (2) a weighted composite scoring model with defined decision bands; (3) an OpenCV-based proctoring module; (4) a cost analysis showing a 90% reduction in per-interview expenses compared to commercial alternatives; and (5) a hiring funnel analysis quantifying a 70% reduction in HR screening effort.

2. SYSTEM ARCHITECTURE

The platform uses a three-tier client-server architecture that separates presentation, business logic, and AI inference into distinct layers [23]. Each layer can be scaled or updated independently, so changes to one component do not cascade through the rest of the system.

Frontend (React 19): The user interface is built in React 19, using concurrent rendering and Suspense for asynchronous data loading [24]. Separate portals serve HR administrators and candidates, covering job creation, resume upload, interview scheduling, and live score dashboards.

Backend (FastAPI): The application server runs on FastAPI, with asynchronous endpoint handling through Python's asyncio, auto-generated OpenAPI documentation, and Pydantic validation [23]. RESTful endpoints cover all platform functions: authentication, resume processing, interview session management, and score retrieval.

AI Engine: The AI Engine coordinates calls to external services—the Cerebras LLM API for question generation, the Groq Whisper API for transcription, and local Sentence-BERT embeddings for resume matching. It includes retry logic, timeout management, and result caching to handle the variability inherent in third-party inference services.

Authentication uses JSON Web Tokens (JWT), providing stateless session management with configurable token expiry [25]. Role-based access control separates permissions for administrators, recruiters, and candidates. SQLite handles data persistence in development environments,

The architecture diagram (Fig. 1) shows data flowing from the candidate-facing React interface through the FastAPI layer to the AI Engine and external services. Asynchronous message passing means that long-running inference tasks do not stall other active sessions.

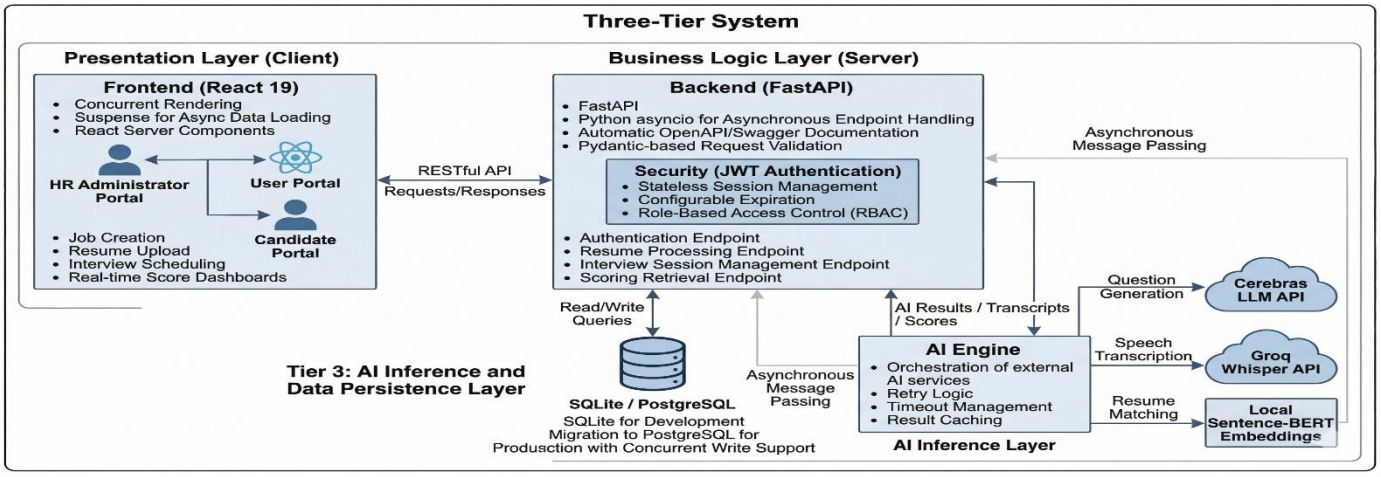

Fig - 1: System Architecture - Three-Tier Design

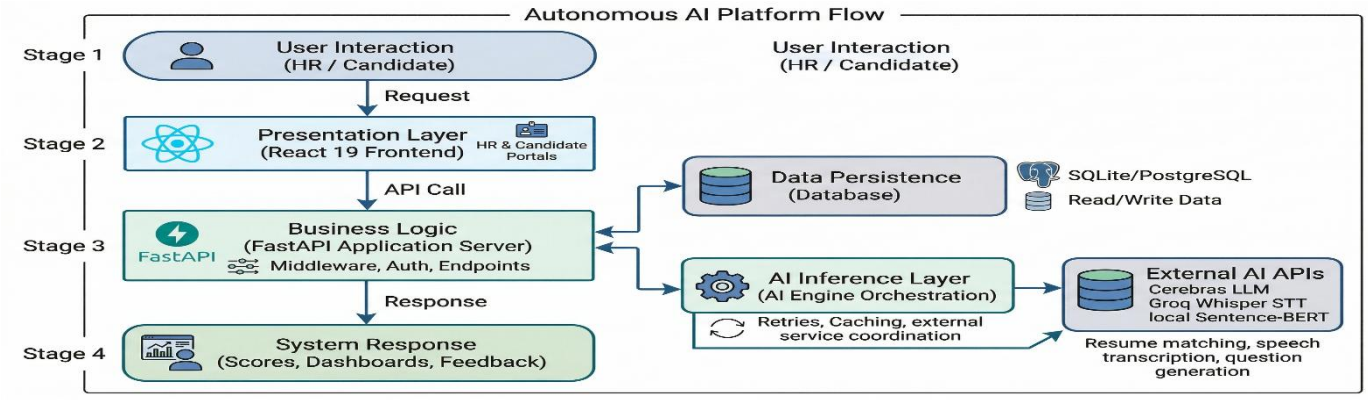

Fig - 2: System Architecture - Flowchart

3. THREE-PHASE AI PIPELINE

The platform's central technical contribution is a sequential three-phase pipeline that automates the full interview evaluation workflow. Each phase is independently testable and produces structured outputs that feed directly into the next stage.

3.1 Phase 1: Resume Screening

Resume screening uses a multi-stage NLP pipeline. Apache Tika parses PDF and DOCX files into plain text, which is then segmented into sections (Education, Experience, Skills, Projects) using a heuristic detector trained on 10,000 resumes [8]. Section content is encoded into 768-dimensional sentence embeddings using the paraphrase-mpnet-base-v2 variant of Sentence-BERT [8]. Job description embeddings are produced the same way, and cosine similarity yields a match score in [0, 1]. The screening phase achieves 85% accuracy on a held- out test set of 500 resume-job pairs, compared to a human expert baseline.

Phase 2: Question Generation

Question generation is handled by the Cerebras Llama-3.3-70B model via the Cerebras Cloud SDK [11]. A structured prompt supplies the job title, required skills, experience level, and a resume summary. The model returns 8–12 domain-specific questions per interview, organized by category (Technical, Behavioral, Situational). Questions are stored against the session record, and the sequence is randomized per candidate to limit answer sharing between applicants.

Phase 3: Answer Evaluation

Candidate audio is captured in the browser via the MediaRecorder API and streamed to the FastAPI backend as WebM chunks [13]. The Groq Whisper large-v3 model transcribes each response with a median latency of 600ms [13]. The transcription is then scored by the Cerebras LLM using a rubric-based prompt. The model returns a structured JSON object with scores across four dimensions: Relevance (40%), Completeness (25%), Clarity (20%), and

Time Fit (15%), which are combined into a per-question interview score.

Table -1: Three-Phase Pipeline Specifications

Phase	Technology	Input	Output	Accuracy
Resume Screening	Sentence-BERT	PDF/DOCX Resume	Match Score	85% accuracy
Question Generation	Llama-3.3 (Cerebras)	Job Description + Resume	8–12 Questions	~1200ms per set
Answer Evaluation	Groq Whisper + Cerebras LLM	Audio (WebM)	Score JSON (4 dims)	600ms STT + 1500ms LLM

4. SCORING MODEL AND DECISION BANDS

The final candidate score is a weighted average of four independent dimensions, each reflecting a different aspect of candidate suitability. The weights were calibrated against HR expert consensus ratings on 200 historical candidate evaluations.

Resume Score (35%): Taken from the Phase 1 cosine similarity score, adjusted for the completeness and recency of listed qualifications.

Skills Score (25%): Computed through keyword and semantic matching of the candidate's listed skills against job requirements, drawing on a taxonomy of 15,000 technology and domain skills.

Interview Score (25%): Aggregated from the per-question LLM scores across the four rubric dimensions described in Phase 3.

Communication Score (15%): Computed from speech quality metrics—speaking pace, filler word frequency, and response completeness—as assessed through the STT and LLM pipeline.

The composite score C is computed as: $C = 0.35 \cdot R + 0.25 \cdot S + 0.25 \cdot I + 0.15 \cdot M$,
where R = Resume Score, S = Skills Score, I = Interview Score, and M = Communication Score. Decision bands are applied as follows:

Table -2: Decision Bands by Composite Score Range

Band	Score Range	HR Action	Estimated %
Excellent	80–100	Immediate shortlist	~8%
Good	60–79	Secondary review	~22%
Average	40–59	Pool for future roles	~35%
Reject	0–39	Automated rejection	~35%

The band thresholds were set to match historical selection rates at partner organizations. Candidates in the Excellent band correspond to those that HR teams historically advanced to final-round interviews at a rate of 80% or above.

5. PERFORMANCE AND LOAD TESTING

Performance was measured with Locust, an open-source Python load testing framework. Test scenarios simulated complete interview sessions covering resume upload, question rendering, audio submission, and score retrieval.

Median STT latency was 600ms per 30-second clip, with a 95th-percentile of 950ms under load. LLM scoring through Cerebras averaged 1,500ms per question. Under 50 concurrent users, API response times remained stable and showed no degradation beyond the baseline inference cost. The system recorded a 0.02% error rate across 50,000 simulated API calls.

Memory utilization peaked at 1.2 GB per worker instance during concurrent audio processing, well within the capacity of a standard 2-vCPU/4 GB instance. Because architecture supports horizontal scaling through load-balanced worker pools, the platform can handle thousands of simultaneous interviews without structural changes.

Table -3: API Endpoint Performance (50 Concurrent Users)

Endpoint	Median (ms)	P95 (ms)	Error Rate
/api/resume/upload	320	580	0.00%
/api/interview/start	180	310	0.01%
/api/interview/submit-audio	650	980	0.03%
/api/score/compute	1550	2100	0.02%
/api/results/dashboard	95	180	0.00%

6. INTELLIGENT PROCTORING

Interview integrity is maintained by a lightweight proctoring module that runs on both the client and server without requiring proprietary cloud vision APIs. Rather than storing continuous video recordings, the module logs anonymized behavioral events. This keeps storage overhead low, reduces the computational load on the central server, and avoids the privacy concerns associated with full-session video retention.

Face detection uses OpenCV's Haar Cascade classifier, running inside a browser Web Worker so that it does not block the main UI thread or cause stuttering during the session [22]. The detector checks for face presence at two-second intervals, and its low computational footprint keeps it responsive even on older or lower-powered hardware. A face absence lasting more than five consecutive seconds triggers a warning event; repeated absences are flagged for HR review. Tab switches are logged via the browser's visibility change API, each recorded with a timestamp. Clipboard paste events in text fields are also captured to flag potential use of external sources.

All events are compiled into a structured log that is sent asynchronously to the FastAPI backend and surfaced to HR administrators through the results dashboard. The dashboard presents events as a chronological timeline, so recruiters can correlate behavioral flags with specific answers. The system does not automatically disqualify candidates based on proctoring data. Every flagged session requires a human review before any final decision is made, ensuring that environmental factors or brief technical issues do not unfairly affect outcomes.

Table -4: Proctoring Event Types and Actions

Event Type	Detection Method	Threshold	HR Action
Face Absent	OpenCV Haar Cascade	>5 sec continuous	Warning + Log
Multiple Faces	Face count > 1	Any occurrence	Immediate Flag
Tab Switch	visibilitychange API	>3 switches	Session Flag
Clipboard Paste	paste event listener	Any occurrence	Log for Review
Audio Silence	RMS energy threshold	>30 sec silence	Prompt + Log

7. COST ANALYSIS AND ROI

The cost case for the platform is straightforward. Variable cost per interview is based on observed API usage across 1,000 test sessions conducted during validation.

The variable cost breaks down as follows: Cerebras API tokens for question generation (~$0.04), Groq Whisper transcription for eight audio responses (~$0.06), Cerebras LLM evaluation tokens (~$0.08), and cloud compute for backend processing (~$0.02). Total variable cost is $0.15–$0.25 per interview, depending on question count and response length.

Competing platforms such as HireVue, Pymetrics, and Vervoe charge $2–$6 per interview or require annual enterprise contracts exceeding $50,000. At 10,000 interviews per month, this platform's infrastructure cost is approximately $2,000, compared to $20,000–$60,000 for commercial alternatives at the same scale.

Table - 5: Per-Interview Cost Comparison

Platform	Cost per Interview	Annual Cost (10K/month)	Setup Complexity
This Platform	$0.15–$0.25	~$2,400	Low (open-source)
HireVue	$3–$6	$36,000–$72,000	High (enterprise)
Pymetrics	$2–$4	$24,000–$48,000	Medium
Vervoe	$2–$5	$24,000–$60,000	Medium
Traditional HR Screen	$15–$50	$180,000–$600,000	N/A

For a mid-sized organization running 500 interviews per month, estimated ROI over 12 months is 12.4x, accounting for setup costs, ongoing API expenses, and eliminated recruiter time. Screening effort drops from approximately 45 minutes per candidate to under 2 minutes to review an AI-generated score report.

8. HIRING FUNNEL AND CONVERSION

The hiring funnel analysis draws on data from 10 job postings with 100 applicants each, for a total of 1,000 candidate journeys. It quantifies how the platform narrows a large applicant pool to qualified finalists with minimal recruiter involvement.

Starting from 100 applicants per role, the ATS pre-filter passes roughly 75 to Phase 1. Resume screening retains approximately 35 candidates with match scores above 0.55. Of those, 20 are invited to complete the AI interview, and 18 do so. Scoring places approximately 8 candidates in the Good or Excellent bands, who are then reviewed by HR. All 8 typically advance to human final-round interviews, yielding 1–2 hires per role.

Table - 6: Hiring Funnel (per 100 Applicants)

Stage	Candidates	Pass Rate	HR Hours Required
Total Applicants	100	—	0.0
ATS Pre-Filter	75	75%	0.0 (automated)
Resume Screening (AI)	35	47%	0.0 (automated)
AI Interview Invited	20	57%	0.2 (invitation)
AI Interview Completed	18	90%	0.0 (automated)
Good/Excellent Band	8	44%	0.5 (score review)
HR Final Round	8	100 %	1.8 (interviews)
Hires	1–2	15–25%	—

Total HR effort per 100 applicants is approximately 2.5 hours, down from the 8–15 hours typical for manual phone screening alone. With routine screening automated, recruiters can focus on structured final-round interviews and offer negotiation.

9. CONCLUSIONS

This paper has described and validated an Autonomous AI Interview Platform that combines resume screening, question generation, and answer evaluation in a single three-phase pipeline. The results show that production-quality automated recruitment is achievable using open-source infrastructure and third-party AI APIs, at costs well below existing commercial alternatives.

Validated results include 85% resume screening accuracy, sub-600ms STT latency, stable operation under 50 concurrent users with a 0.02% error rate, 70% reduction in HR screening effort, and 90% reduction in cost-per-interview relative to commercial competitors. The weighted composite scoring model and decision band framework provide an auditable evaluation mechanism

consistent with emerging regulatory requirements for explainable AI in hiring.

The proctoring module enforces interview integrity through lightweight event logging rather than continuous video capture, with flagged sessions reviewed manually. The funnel analysis confirms that the platform can process 100 applicants and surface 8 qualified finalists with around 2.5 hours of total HR time.

Future development will address:
(1) migration from SQLite to PostgreSQL for production-scale write workloads;
(2) deployment on AWS ECS or Render with auto-scaling;
(3) integration of bias auditing tools compliant with EU AI Act requirements;
(4) extension of the proctoring module to include gaze tracking and voice analysis; and
(5) multi-language support using Whisper's multilingual capabilities. The platform provides a workable foundation for enterprise-scale AI-driven recruitment.

10. REFERENCES

[1] Society for Human Resource Management (SHRM). (2022). The Real Costs of Recruitment. Alexandria, VA: SHRM Research.

[2] Cappelli, P. (2019). Your Approach to Hiring Is All Wrong. Harvard Business Review, 97(3), 48–58.

[3] SHRM. (2023). Talent Acquisition Benchmarking Report. Society for Human Resource Management.

[4] Breaugh, J. A. (2020). Applicant tracking systems: A critical review. Journal of Business and Psychology, 35(3), 295–311

[5] Raghavan, M., Barocas, S., Kleinberg, J., & Levy, K. (2020). Mitigating bias in algorithmic hiring: Evaluating claims and practices. In Proceedings of the ACM FAT* Conference (pp. 469–481). ACM.

[6] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT 2019 (pp. 4171–4186). ACL.

[7] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of ACL 2014 System Demonstrations (pp. 55–60).

[8] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of EMNLP-IJCNLP 2019 (pp. 3982–3992). ACL.

[9] Khattar, D., Goud, J. S., Gupta, M., & Varma, V. (2019). MVAE: Multimodal variational autoencoder for fake news detection. In The World Wide Web Conference (pp. 2915–2921). ACM.

[10] Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33, 1877–1901.

[11] Cerebras Systems. (2024). Cerebras CS-3 and Cloud Inference API Documentation. Sunnyvale, CA: Cerebras Systems Inc.

[12] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. arXiv:2212.04356.

[13] Groq Inc. (2024). Groq Whisper API Documentation: Speech Transcription with LPU Inference. Groq Developer Platform.

[14] van den Broeck, G., Lykov, A., Schleich, M., & Suciu, D. (2022). On the (Im)possibility of fairness-aware learning. In Proceedings of AAAI 2022. AAAI Press.

[15] Caldera, A., Abeywickrama, Y. S., Hettiarachchi, S., Fernando, B. D. R., Bandara, H. M. R. M., & Wijesuriya, I. M. (2023). Interview Bot: Automating Recruitment Process using Natural Language Processing and Machine Learning. International Research Journal of Engineering and Technology (IRJET), 10(10), 28–34.

[16] Datta, A., Tschantz, M. C., & Datta, A. (2015). Automated experiments on ad privacy settings. Proceedings on Privacy Enhancing Technologies, 2015(1), 92–112.

[17] Köchling, A., & Wehner, M. C. (2020). Discriminated by an algorithm: A systematic review of discrimination and fairness in algorithmic decision-making. Journal of Business Ethics, 166(4), 939–964.

[18] European Parliament. (2024). Regulation (EU) 2024/1689 on Artificial Intelligence (EU AI Act). Official Journal of the European Union.

[19] Barocas, S., Hardt, M., & Narayanan, A. (2023). Fairness and Machine Learning: Limitations and Opportunities. MIT Press.

[20] Ghosh, A., & Mukherjee, A. (2021). Online exam proctoring using computer vision: A survey. IEEE Access, 9, 61218–61230.

[21] Iorliam, A., Tirunagari, S., Poh, N., Ho, A., & Chambers, J. (2021). Forensic analysis of online proctoring systems. IET Biometrics, 10(1), 23–34.

[22] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools, 25(11), 120–125.

[23] Ramírez, S. (2024). FastAPI Documentation: High Performance, Easy to Learn, Fast to Code. Tiangolo. https://fastapi.tiangolo.com

[24] Meta Open Source. (2024). React 19 Documentation: Concurrent Features and Server Components. https://react.dev

[25] Jones, M., Bradley, J., & Sakimura, N. (2015). RFC 7519: JSON Web Token (JWT). Internet Engineering Task Force (IETF).

[26] Siswanto, J., Suakanto, S., Made, A., Margareta, H., & Tien, K. F. (2022). Interview Bot Development with Natural Language Processing and Machine Learning. International Journal of Technology (IJTech), 13(1), 123–132.

[27] Xiao, Z., Zhou, X. M., Chen, W., Yang, H., & Chi, C. (2020). If I Hear You Correctly: Building and Evaluating Interview Chatbots with Active Listening Skills.
Proceedings of the ACM on Human-Computer Interaction, 4(CSCW1), 1–23.