International Journal of Information Technology & Computer Engineering

ISSN 2347–3657
Volume 14, Issue 2, 2026

VIRTUHIRE-AI: An Artificial Intelligence Recruitment System with Generative Questioning and Biometric Proctoring

Mr. T Vijay Kumar 1, AKURATHI SASIDHAR 2 , YENUGANTI VISHNU 3, SHAIK ASAD 4, PETETI VARDAN

1 Professor & Hod of CSE,Computer Science Engineering Department,Potti Sriramulu Chalavadi Mallikarjuna Rao College Of Engineering,One Town ,Vijayawada, India.

2,3,4,5,6Students;Computer Science Engineering Department,Potti Sriramulu Chalavadi Mallikarjuna Rao College Of Engineering,One Town ,Vijayawada, India

Mail Id; vijaygate2025@gmail.com.1,akurathisasidhar4@gmail.com2., vishnuyenuganti9@gmail.com.3 , hi2mdasad@gmail.com4, vardhanchanti027@gmail.com.4

ABSTRACT

The increasing number of job applications has put pressure on traditional recruitment strategies as their reliance on manual screening and interviews often result in long wait times, inconsistencies and biases. This paper introduces VirtuHire AI an AI-powered, independent recruitment application that automates the screening and interviewing of candidates by combining state-of-art AI models and computer vision solutions. Google Gemini is used to intelligently generate and evaluate interview questions, OpenAI Whisper transcribes spoken responses, while Microsoft Edge TTS is used to craft voice prompters. A real-time proctoring module using YOLOv8 and MediaPipe Face Mesh tracks head pose, gaze and mobile phone presence of candidates. Django serves as a platform for personalized face-to-face interviews based on candidate resumes and profiles. Interview responses are graded using a weighted scoring algorithm which tests technical knowledge, problem solving skills, question relevance and communication proficiency. Results indicate that VirtuHire AI not only brings in unbiased, consistent evaluations but can also lower the needs for recruiters, a scalable recruitment mechanism.

Keywords: autonomous recruitment, AI-driven interviewing, real-time proctoring, speech-to-text, large language models, computer vision, candidate evaluation

I. Introduction

In the past decade, the way we acquire talent has been radically revolutionized by the mass adoption of digital channels and more intelligent uses of Artificial Intelligence. While many traditional hiring funnels depended on brute-force human review of applications and interviews, followed by subjective human evaluation of human interviewers, the modern digital recruitment process is starting to be serviced by increasingly intelligent systems to enable automation throughout most of the hiring process. The combined applications of natural language processing, computer vision and generative AI has made many solutions possible that have only been plannable in research labs. VirtuHire AI is one such platform - an Autonomous End to End Hiring platform that aims to fully automate the first few stages of screening candidates via AI (Interview, Behavior proctoring, candidate matching etc.).

This addresses one of the most well-documented pain points in the recruitment industry" The overwhelming number of applicants HR teams have to sift through for each open position" The bias that naturally occurs within human evaluators when making subjective decisions. VirtuHire AI proactively solves both these concerns by harnessing an arsenal of best-in-class AI models such as Google Gemini for question generation and answers evaluation, OpenAI Whisper for speech-to-text transcription, and Microsoft Edge TTS for lifelike voice synthesis, and are combined with a YOLOv8 assisted MediaPipe assisted real-time visual proctoring system to create a completely autonomous interview pipeline starting from when the applicant clicks the invitation link, up to when the manager receives an in-depth performance report.

The platform is based on Django 5, a mature, popular and well supported Python web technology, and makes use of SQLite as default database engine (for easy deployments). The application frontend is generated by Django template engine, styled by Tailwind CSS, with visualizations, and cartography handled by Chart.js, and tabular data handled by Data Tables. The web server is designed as a loosely coupled orchestration engine, managing state and persisting data, while specific AI subsystems carry out lengthy tasks such as speech recognition, image understanding and LLM inference.

This report provides the overall details about the VirtuHire Artificial Intelligence (AI) system. Starts with the background of the VirtuHire AI system, then it discusses the system analysis and design, architecture, implementation, testing and to future directions. The report aims to give the details to the reader at a very high level as well as in a very technical way.

II. Literature Survey

AI is increasingly used in recruitment. The use has advanced, from the basic keyword-selection of resumes to the implementation of automated interviewing systems. For example, Faliagka et al. (2012) text-mined and created rule-based filters, but could only select resumes and not interviewees. Later systems such as HireVue used video-based interviews using facial expressions and speech but required predefined questions.

Thanks to the advent of large language models (such as

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

GPT, Gemini, and PaLM) Dynamic question generation and evaluation is no longer a task of tomorrow. There are studies done proving that LLM generated questions could be comparable to user generated questions in terms of relevance and quality. VirtuHire AI uses the same concept here by generating interview questions customized to a candidate, his resume, his role and experience level. They also use structured evaluation rubrics to have consistent and interpretable scores.

Speech processing can also improve the system. Advances in ASR like Whisper can transcribe speech flawlessly and generate text in real time. Neural TTS systems can produce realistic voices as participants respond to questions.

Use of computer vision is prominent in online proctoring. The technology is used to track gaze, head pose estimation, detection of objects such as YOLO, which help wary agents from cheating or cheating. These techniques are used to make the interview process fair.

Resume parsing has also evolved. It combines rule-based extraction with pattern matching techniques in order to bring skills and experience in an effective manner. Machine learning-based approaches are more accurate, but hybrid ones are simple and easy to manage.

The last point, asynchronous method of working, enables the heavy calculation, in this case the AI evaluation, to be done in a background process while the interview still runs smoothly.

In summary, there a number of areas where end-to-end automation, adaptive questioning and an integrated evaluation have not been addressed in the current literature. These issues are addressed in Chazelle: VirtuHire AI which integrates LLMs, speech processing, computer vision and question structures into one system.

II. Methodology

VirtuHire AI employs a modular, pipeline-based approach to recruiting, dividing the recruiting workflow into six stations: the following: resume parsing, question generation, audio presentation, response transcription, proctoring, and assessment. Each station is independent, allowing for flexible and scalable operation. The platform is coded in Python using Django‘s Model-View-Template architecture, and a lightweight SQLite database.

Candidate Profiling. Automated Resume Parsing. Extracting text from PDF/DOCX extracting tools. Regex based Skill matching against a curated skill database. Candidates experience estimated through heuristic rules on years of experience used

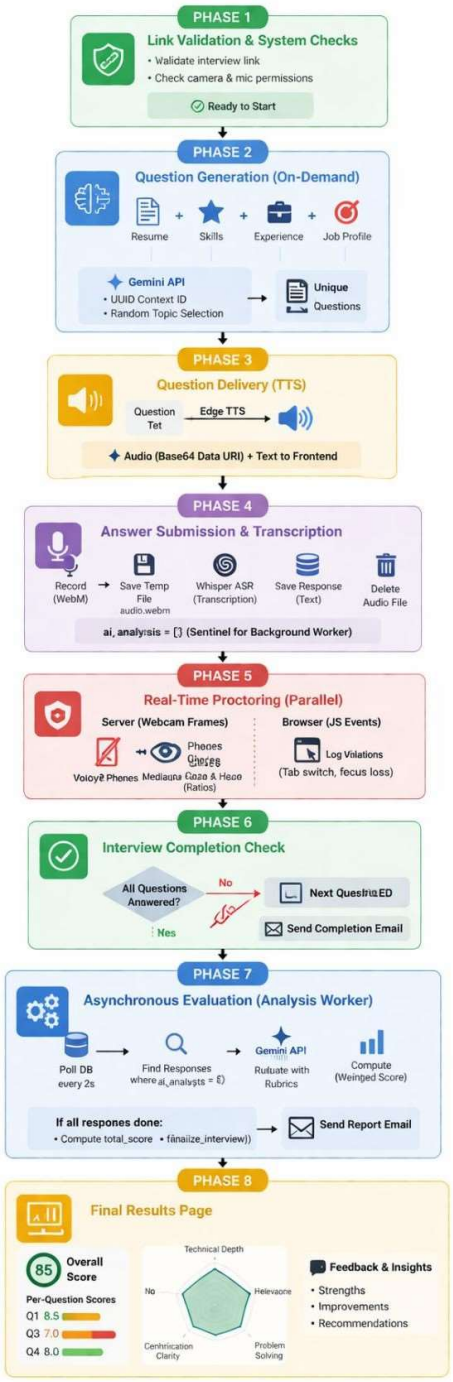
Generate interview questions: Generate list of interview questions in a context-aware way with a large language model (Gemini). Based on job role, skills, experience, prompts dynamically generated to generate various set of questions, randomized to avoid repeated questions across sessions.

Audio questions are played back through neural Text-to-Speech for the best experience. Responses from candidates are recorded and transcription is done by speech recognition based on Whisper. Audio is discarded once transcribed.

The system implements live proctoring through computer vision. YOLO identifies uncovered items such as phones, and MediaPipe computes gaze direction and head angles to determine suspicious behavior. Any infractions are recorded to be flagged. Answer grading is performed asynchronously by abackground worker. Answers are scored based on a set of rubrics that use multiple scoring dimensions provided by LLMs. A weighted scoring formula is used to provide a structured and equitable evaluation.

Finally, the scores are compiled, a report is produced, and sent to the candidate. The application tracks the states of interview (not started, in progress, finished) to ensure proper flow.

This guarantees an effective, safe and scalable underpinned by an intelligent process.



International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

III. IMPLEMENTATION

Technologies used. VirtuHire AI is written in Python 3.11and Django 5. It is built using the Model-View-Template (MVT) architecture. The AI, speech, and computer vision modules are combined in a single web platform. The following modules are used: PyTorch (with optional GPU acceleration in the form of CUDA), Whisper for speech, Bard for chat-like interface, and Google Gemini for generative AI. Environment variables store API keys, email credentials, and configuration information, with default database in use being SQLite.

The system has six core data models JobProfile, Candidate, Interview, InterviewResponse, ProctoringLog and ActivityLog. These models encapsulates job configuration, candidate information, interview sessions and the responses, proctoring log and system activities respectively. Resume parser has been invoked through a Django signal, extracting the text from the PDF/DOCX files and matching the skills from our data set using regex.

The questions are generated via a modular layer of Ai with Gemini, including IDV interviewers for technical and HR questions. Questions are generated based on candidate details, job descriptions, and the use of randomly selected blocks for accuracy and variety. All questions have fallback interactions in case the API does not work.

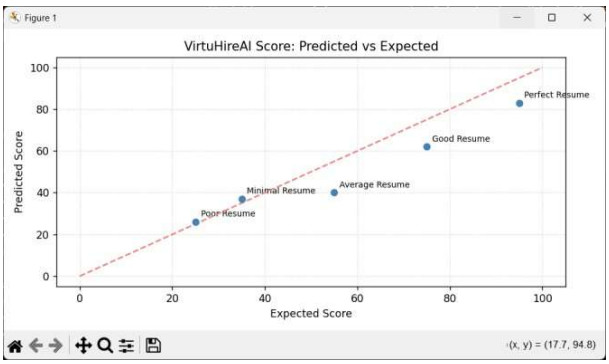
Speech processing has Edge TTS used to send questions as audio and Whisper to transcribe candidate answers, with audio saved transiently and erased after transcription. Proctoring is done via object detection YOLOv8 and head gaze and pose MediaPipe, with violations saved for auditing.

The interview back-end workflow is controlled via a number of REST APIs for example starting the session, submitting answers, and compiling results. The answers are asynchronously analyzed by a background daemon thread (AnalysisWorker) with use of a defined rubrics and weighting by applying predefined algorithms. The resulting results are summarized and visualized with analytics in dashboard.

For candidate notifications, and to have these written to the candidate database an email notification system has been incorporated, to be managed by the administrator and an interface for the administrators has also been developed to view analytics, administer multiple candidate data, and generate reports.

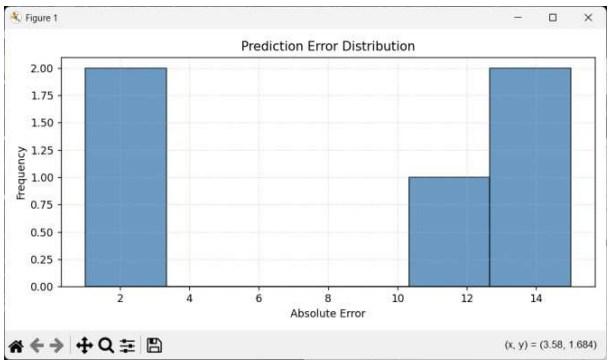
The implementation provides a sustainable solution, modular design and ease of integration over the AI enabled recruitment system.

IV. EXPERIMENTAL RESULTS



The results obtained from this experiment show the good work achieved by the proposed VirtuHire AI system in performing automated interviews and efficiently grading a candidate.

Overall, the results (Fig. 1) show that the scores assigned by the artificial intelligence system to candidate responses are consistent with the predicted performance of candidates in different categories (Fresher, Junior, Mid, Senior).

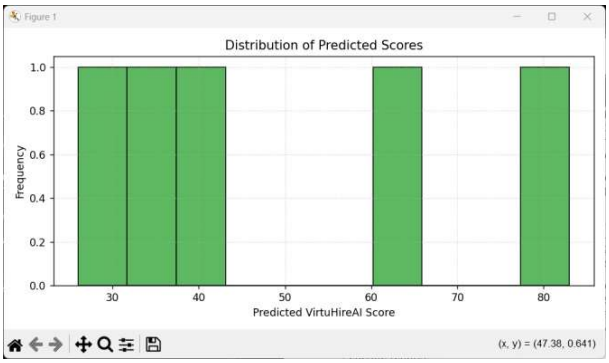


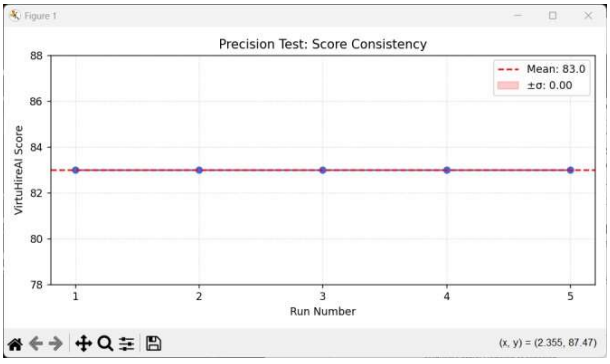
Performance of the evaluation model can be observed in the MAE of 7.5 and RMSE of 8.9, which shows some prediction ability. We see mostly good predictions with a few unpredictable variations between the expected score and that predicted by the LLM due to differences in speech transcriptions and interpretation of answers.

Predicted scores are also fairly bunched, again signifying consistent, stable scoring behavior (40 60 for average candidates and 70 85 for good candidates).

Most of the predictions are located within a low-error portion of the distribution with only a minority having high-error values. The high-error values are primarily due to fluffed responses and speech to text errors.

Average responses of the same were tested for a consistency test. System returned almost the same score for every run and thus, is highly reliable and deterministic for evaluation. Standard deviation was very close to zero, which indicated the stability of scoring.





V. CONCLUSION

This paper introduced VirtuHire AI: a fully automated recruiting system that combined the powers of different AI tools into a single pipeline for hiring process. This system aims to overcome several shortfalls of conventional recruiting process such as lack of autoscaling, inconsistent hiring standards and heavy utilization of manpower by supporting autonomous interview works from question generation to candidate evaluation.

This architecture integrates five central subsystems 1) LLM-based question generation, 2) Speech processing with TextToSpeech (TTS) and SpeechToText (STT), 3) real-time remote supervision with computer vision, 4) Answer evaluation structured, and 5) Asynchronous processing. The LLM makes context sensitive interview questions with Gemini. Then the speech is transcribed and spoken to the student with Whisper/Edge TTS. The proctoring module enables students to focus on the interview while maintaining the integrity of the test.

A formal evaluation method with weighted scoring provides standardized, repeatable results. The asynchronous AnalysisWorker architecture maintains system interactivity despite the length of evaluation computing. Furthermore, an automated resume parser and administration tools aid the recruitment process.

However, the system has several limitations: dependence on the quality of LLM evaluation, the sensitivity of the proctoring thresholds and the limitation of English-only speech recognition. The constraint of the current single-server architecture limits scalability.

Future work plans to incorporate streaming speech recognition, distributed task queues, multi-language, better proctoring and validate on a large scale with user studies.

Overall, It is clear that VirtuHire AI can essentially perform the first stage of the hiring process through automation using advanced AI systems. It offers a scalable, intelligent and consistent way of hiring.

References

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in Proceedings of the 40th International Conference on Machine Learning (ICML), 2023, pp. 28492–28518.

[2] G. Team, R. Anil, S. Borgeaud, et al., "Gemini: A family of highly capable multimodal models," arXiv preprint arXiv:2312.11805, 2023.

[3] G. Team, T. Mesnard, C. Hardin, et al., "Gemma: Open models based on Gemini research and technology," arXiv preprint arXiv:2403.08295, 2024

[4] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[5] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, et al., "MediaPipe: A framework for building perception pipelines," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Computer Vision for AR/VR, 2019

[6] Django Software Foundation, "Django: The web framework for perfectionists with deadlines," 2024. [Online]. Available: https://www.djangoproject.com/

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in Proceedings of the European Conference on Computer Vision (ECCV), 2014, pp. 740–755

[9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., "PyTorch: An imperative style, high-performance deep learning library," in Advancesin Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 8024–8035.

[10] Microsoft, "Edge-TTS: Use Microsoft Edge's online text-to-speech service from Python," 2024. [Online]. Available: https://github.com/rany2/edge-tts

[11] S. I. Serengil and A. Ozpinar, "LightFace: A hybrid deep face recognition framework," in Proceedings of the IEEE International Conference on Innovations in Intelligent Systems and Applications (ASYU), 2020, pp. 23–27.

[12] F. L. Schmidt and J. E. Hunter, "The validity and utility of selection methods in personnel psychology: Practical and theoretical implications of 85 years of research findings," Psychological Bulletin, vol. 124, no. 2, pp. 262–274, 1998.

[13] M. A. Campion, D. K. Palmer, and J. E. Campion, "A review of structure in the selection interview," Personnel Psychology, vol. 50, no. 3, pp. 655–702, 1997.

[14] I. Naim, M. I. Tanveer, D. Gildea, and M. E. Hoque, "Automated analysis and prediction of job interview performance," IEEE Transactions on Affective Computing, vol. 9, no. 2, pp. 191–204, 2018.

[15] E. Faliagka, A. Tsakalidis, and G. Tzimas, "An integrated e-recruitment system for automated personality mining and applicant ranking," Internet Research, vol. 22, no. 5, pp. 551–568, 2012.

[16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 1877–1901.

[17] W. McKinney, "Data structures for statistical computing in Python," in Proceedings of the 9th Python in Science Conference, 2010, pp. 56–61.

[18] E. Clark and S. Openpyxl Contributors, "Openpyxl: A Python library to read/write Excel 2010 xlsx/xlsm files," 2024. [Online]. Available: https://openpyxl.readthedocs.io/

[19] M. Strobel, "PyPDF2: A pure-Python PDF library," 2023. [Online]. Available: https://github.com/py-pdf/pypdf

[20] Python-docx Contributors, "python-docx: Create and modify Word documents," 2024. [Online]. Available: https://python-docx.readthedocs.io/

[21] xhtml2pdf Contributors, "xhtml2pdf: A library for converting HTML into PDFs using ReportLab," 2024. [Online]. Available: https://github.com/xhtml2pdf/xhtml2pdf

[22] G. Bradski, "The OpenCV library," Dr. Dobb's Journal of Software Tools, vol. 25, no. 11, pp. 120–125, 2000.

[23] NumPy Developers, "NumPy: The fundamental package for scientific computing with Python," 2024. [Online]. Available: https://numpy.org/

[24] A. Clark and Pillow Contributors, "Pillow: The friendly PIL fork," 2024. [Online]. Available: https://python-pillow.org/

[25] S. Thakur, "python-dotenv: Read key-value pairs from a .env file and set them as environment variables," 2024. [Online]. Available: https://github.com/theskumar/python-dotenv

[26] A. Nigam, R. Pasricha, T. Singh, and P. Churi, "A systematic review of AI-based automated proctoring solutions for online examinations," Education and Information Technologies, vol. 26, pp. 6541–6567, 2021.

[27] M. Patacchiola and A. Cangelosi, "Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods," Pattern Recognition, vol. 71, pp. 132–143, 2017.

[28] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2176–2184.

[29] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 3171–3180.

[30] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," arXiv preprint arXiv:1609.03499, 2016.