ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 3, 2025

AI Interview Bot for Freshers Using Google Gemini and LangChain

[1]M. PRASANNA KUMAR, [2]G. HANUMANTHA RAO, [3]S. SAI SUSHMA

[1,2,3] Asst. Prof., Department of CSE, MAM Women's Engineering College, Narasaraopet, Palnadu, A.P., India.

Abstract-

This paper introduces an AI-driven web application for automated interview evaluation, built using Django and integrated with Google Gemini through LangChain. It streamlines the recruitment process by generating role-specific interview questions and assessing candidate responses in real time, making it especially suitable for fresh graduates and entry-level roles. Candidates register via a secure portal and undergo a four-question interview tailored to their selected job role. Questions are generated dynamically using a large language model (LLM), ensuring relevance and simplicity. Each response is evaluated by the AI, which returns a score (0–5) and a qualification status in structured JSON format. At the end of the session, the system calculates the average score, determines the final result, and automatically emails it to the candidate. The platform includes admin features for managing users and viewing results, along with OTP-based password recovery and profile image support. This scalable, unbiased system offers an efficient solution for bulk hiring and academic assessments by minimizing human effort while maintaining evaluation accuracy.

I. INTRODUCTION

The hiring process is a critical function for any organization, demanding both accuracy and efficiency in identifying suitable candidates. Traditional interviews, while effective, are often time-consuming, resource-intensive, and prone to human bias. With the rise of artificial intelligence (AI), there is a growing opportunity to enhance recruitment methods through automation and intelligent decision-making. This project leverages AI technology to develop an automated interview evaluation system that streamlines candidate assessment while ensuring consistency and fairness. Built using Django as the web framework, the system integrates Google Gemini through the LangChain API to enable dynamic interview interactions. It is specifically designed to cater to freshers and entry level applicants by generating simple, role-specific questions. The candidate's answers are analyzed in real time by the AI, which assigns a score and determines qualification based on predefined criteria. The platform also supports user authentication, OTP based password recovery, and an administrative dashboard for user management and result tracking. By automating the interview process, this system reduces manual effort, speeds up candidate screening, and enhances the overall recruitment experience for both organizations and applicants.

II. LITEARTURE SURVEY

The integration of artificial intelligence in automating job interviews has been a growing area of interest. Romadon et al. [1] explored the effectiveness of traditional NLP techniques such as TF-IDF and Word Embeddings in evaluating candidate responses, establishing a foundation for automated grading systems. Similarly, Pandey et al. [2] introduced an interview bot capable of automatic question generation and answer evaluation, demonstrating the potential of combining natural language processing and machine learning to simulate human-like interviewers.

In understanding the human-computer interaction aspect, Pickard and Roster [3] investigated whether the visual presence of a human face in automated systems influences a candidate's willingness to disclose information, revealing that visual cues significantly affect user behavior during digital interviews. Expanding on grading capabilities, Yusuf and Lhaksmana [4] developed a Support Vector Machine (SVM)-based automated interview assessment system, showcasing promising results in talent acquisition processes.

From a healthcare perspective, Fang et al. [5] demonstrated the utility of NLP techniques for qualitative data classification in cancer patient interviews, while recent studies have leveraged multimodal data such as facial, vocal, linguistic, and cardiovascular signals for biomarker analysis in remote interviews [6]. These works underscore the importance of emotion and sentiment recognition in enhancing interview evaluations.

Machine learning has also been applied to assess scientific reasoning in clinical interviews, as demonstrated by Beggrow et al. [7], who compared ML-evaluated outputs with human performance. Emotion recognition plays a critical role in evaluating soft skills during interviews. Haq et al. [8] focused on speaker-dependent audio-visual emotion recognition,

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 3, 2025

a vital component in assessing candidate affect and engagement.

Datasets play a vital role in training and evaluating automated interview and emotion recognition systems. The Toronto Emotional Speech Set (TESS) [9] provides a wide range of emotional speech samples, useful for speech-based emotion classification. The CREMA-D dataset [10], which includes audio-visual emotional expressions from actors, supports the development of multimodal emotion recognition models. Meanwhile, the International Survey on Emotion Antecedents and Reactions (ISEAR) dataset [11] offers textual records of emotional experiences, which are valuable for emotion recognition in natural language processing tasks. [12] offers a rich medical domain dataset, while SQuAD [13] is widely used for machine comprehension tasks.

Advanced language models like BERT [14] have significantly improved contextual understanding in QA systems, enabling nuanced grading of open-ended responses. These models are often built and deployed using large-scale machine learning platforms like TensorFlow [15], ensuring scalability and real-time application.

Collectively, these studies and resources illustrate a comprehensive shift towards intelligent, scalable, and emotionally-aware automated interview systems. They highlight key components—response grading, emotion recognition, question generation, and natural language understanding—as critical research areas for future developments in AI-based recruitment technologies.

III. METHODOLOGY

The proposed system is developed as a web-based application that automates the interview process using generative AI. The methodology is divided into several key stages: user authentication, interview question generation, answer evaluation, result computation, and administrative control.

User Registration and Login

Users, particularly candidates, begin by registering through a secure web interface. Registration involves providing personal details, including name, email, mobile number, password, and a profile image. After registration, the account remains inactive until approved by the administrator. Once activated, users can log in to access the interview system. An OTP based password reset mechanism is implemented to enhance security and usability.

Interview Initialization

Upon login, candidates initiate the interview by selecting their desired job role. The system stores this information and sends an initial prompt to the Google Gemini large language model (LLM) using LangChain, asking it to generate a simple, beginner friendly question relevant to the selected role. This begins a four-question interview session.

Dynamic Question Generation

The LLM receives context messages that include the job role and previous interactions, and then generates the appropriate question. This dynamic generation ensures questions are context-aware, role specific, and suitable for freshers.

Answer Evaluation

After each candidate response, the system constructs an evaluation prompt that includes the original question and the answer. This is sent to the Gemini model, which returns a structured JSON containing a score (from 0 to 5) and a qualification flag ("yes" or "no"). The score is stored in the database along with the corresponding question and answer.

Result Calculation and Email Notification

Once all four questions are answered, the system calculates the average score for the candidate. If the score meets a predefined threshold (≥ 3), the candidate is marked as "Qualified"; otherwise, they are "Disqualified." An automated email is then sent to the candidate with their result and feedback.

Administrative Dashboard

The admin interface allows for user account management, including activation, deactivation, and deletion of users. It also provides access to all candidate interview results and average scores for monitoring and evaluation.

System Scalability and Deployment

To ensure the system is adaptable for real-world use, scalability and deployment considerations are integrated into the design. The application follows a modular architecture using Django's MVC (Model View-Controller) pattern, allowing individual components—such as user management, AI integration, and result computation—to be updated or expanded independently. The system is designed to handle multiple concurrent interview sessions, with

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 3, 2025

session-specific data such as candidate ID, interview progress, and question history stored in Django's session framework. This prevents data overlap and maintains session integrity

IV. SYSTEM ARCHITECTURE

The system architecture is presented in fig.1.

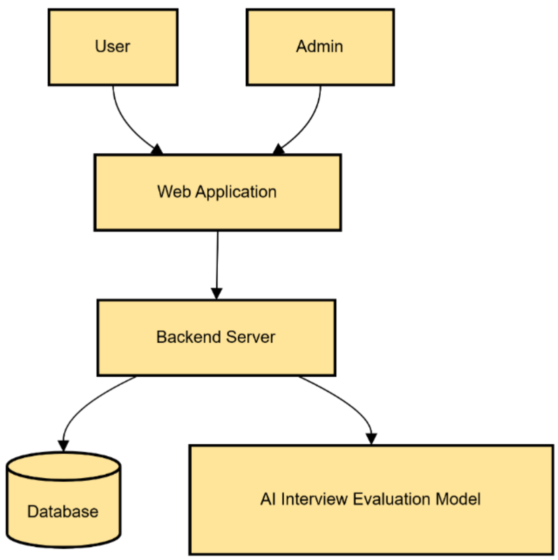

Fig.1. System architecture

The system architecture of the proposed AI-based Interview Evaluation System follows a modular, layered design that ensures flexibility, scalability, and ease of integration with external AI services. At the forefront is the **Presentation Layer**, which serves as the user interface. Developed using Django templates, it allows users—including candidates and administrators—to interact with the system through web pages for registration, login, interview participation, and result viewing. Beneath this lies the **Application Layer**, responsible for handling all business logic. This layer manages user sessions, processes form data, tracks the interview flow, and communicates with both the database and AI components. Central to the project is the **AI Integration Layer**, where the system interfaces with Google Gemini via LangChain. This layer handles two main tasks: dynamic generation of role-specific interview questions and evaluation of candidate answers. The AI provides structured responses in JSON format, which are parsed and used to calculate interview scores and qualification status. The **Database Layer** ensures persistent storage of user credentials, candidate profiles, interview questions, answers, and scores using Django's ORM. Supporting modules include an **Email and OTP Module** that handles password resets and result notifications through email. The entire system is designed to support concurrent interview sessions using Django's session management, ensuring isolated user experiences. It can be deployed on cloud platforms like Heroku or AWS for scalable access. This architecture makes the system well-suited for realworld use in recruitment processes, especially for bulk assessments or academic evaluations, while maintaining automation, accuracy, anduser convenience.

V. IMPLEMENTATION

The implementation of the proposed AI-based Interview Evaluation System is executed using a combination of web technologies, cloud-based AI services, and a structured backend framework. The system is built with Django, a powerful Python web framework that supports modular development, secure authentication, and seamless integration with databases and APIs. The core functionality revolves around automating interview question generation and answer evaluation using Google Gemini through LangChain, a framework that simplifies interaction with large language models (LLMs). The major components of the system implementation are described below

User Registration and Authentication

The system provides a secure registration process where users (candidates) can create accounts by entering their name, email, mobile number, password, and uploading a profile image. Django's FileSystemStorage handles media uploads, storing profile pictures in a designated folder. Once registered, users remain in an inactive state until approved by an administrator.

A secure login system is implemented to allow only verified users to access the platform. Upon successful login, user session data is stored, including login time, user ID, and profile image. An OTP-based password reset module is also developed. If a candidate forgets their password, they can request an OTP via email. This OTP is verified before allowing the user to set a new password, ensuring a secure account recovery process.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 3, 2025

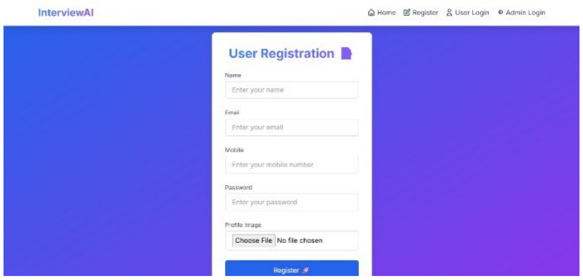

Fig. 2. User Registration.

Interview Initialization and Role-Based Setup

Once logged in, the candidate can begin the interview by selecting a job role or entering a job description. This input forms the basis for generating customized interview questions. The system initializes a session for the candidate and prepares a prompt for the AI model: *"Ask a simple and easy-to-understand interview question suitable for a fresher applying for the role: [job description]"*. This dynamic setup ensures that the interview is personalized and contextually relevant to the user's interest.

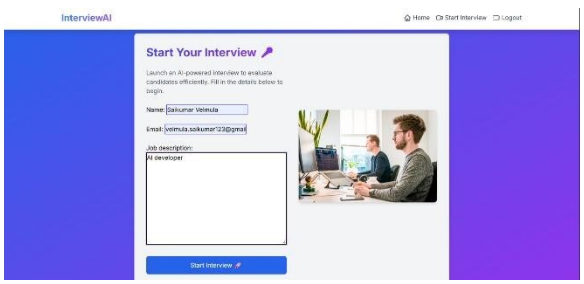

Fig. 3. Interview Initialization and Role-Based Setup

Dynamic Question Generation Using Google Gemini

The system integrates with Google Gemini via the LangChain framework. LangChain simplifies communication with LLMs and allows structured prompting using a conversation-like format. For each question, the system sends previous conversation history along with role-based context to Gemini. The model responds with a unique, role-specific question that is stored in the session and displayed to the candidate.

Each session maintains a sequence of messages—both questions and candidate answers—to guide the LLM in generating follow-up questions. This allows the interview to feel coherent and continuous, mimicking a real-life interviewer scenario.

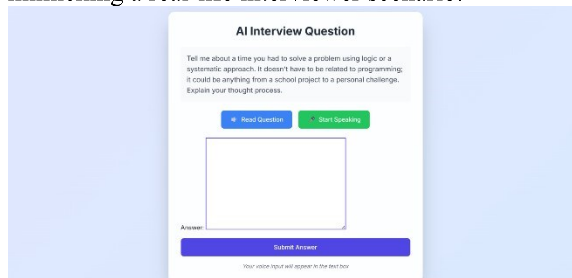

Fig. 4. Dynamic Question Generation

Answer Collection and AI-Based Evaluation

After receiving a question, the candidate inputs their answer through a web form. This answer, along with the question, is sent as part of a prompt to the Gemini model for evaluation. The evaluation prompt is structured as follows:

Evaluate the following candidate answer to the interview question. Question: [question] Answer: [answer] Respond in JSON format like:

{"score": <0-5>, "qualified": "yes" or "no"} Gemini processes this and returns a structured JSON response. The system uses regex as a fallback method to extract JSON in case of any formatting issues. The parsed score and qualification status are stored in the database using Django ORM, along with the corresponding question and answer.

Interview Completion and Result Computation

The interview consists of four questions. After all responses are collected and evaluated, the system calculates the average score. If the average is 3.0 or above, the candidate is marked as "Qualified"; otherwise, they are labeled as "Disqualified." A personalized result email is sent to the candidate, including their name, role, and average score, along with a congratulatory or motivational message depending on the outcome. This is handled through Django's built-in send_mail functionality.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 3, 2025

Fig.5. Interview Completion and Result Computation

Administrative Control Panel An administrator interface is included to manage users and oversee system operations. Admins can:

View all registered users

Activate, deactivate, or delete accounts

Monitor individual interview results Access a summary table of all candidates, including their average scores and final status

This panel ensures that system access is moderated and that interview data is transparently available for review.

VI. CONCLUSION

The AI-based Interview Evaluation System presented in this paper offers a modern, efficient, and scalable solution for automating the initial stages of recruitment and skill assessment. By leveraging the power of Google Gemini through LangChain, the system is capable of generating dynamic, role-specific interview questions and evaluating candidate responses in real time with minimal human intervention. This not only streamlines the interview process but also ensures consistency, fairness, and objectivity in candidate assessment.

The system is built using Django, a robust and secure web framework that supports modular development, user session handling, and seamless database integration. Additional features such as user registration with admin approval,OTP-based password recovery, result-based email notifications, and a comprehensive admin dashboard contribute to a complete and user-friendly platform.With its ability to personalize interviews based on job roles, automate evaluation using structured prompts, and manage candidates efficiently, this solution can significantly reduce the burden on HR teams and educational evaluators. Moreover, the system's design supports future scalability and integration with other AI services, making it adaptable for broader applications in remote hiring and e-learning environments.

In conclusion, this project demonstrates how artificial intelligence can enhance traditional processes by making them faster, smarter, and more accessible—marking a step forward in the digital transformation of recruitment and assessment systems.

REFERENCES

[1] A. W. Romadon, K. M. Lhaksmana, I. Kurniawan, and D. Richasdy, "Analyzing TF-IDF and Word Embedding for Implementing Automation in Job Interview Grading," in *Proc. 2020 8th Int. Conf. Inf. Commun. Technol. (ICoICT)*, Yogyakarta, Indonesia, 2020, pp. 1–4.

[2] R. Pandey, D. Chaudhari, S. Bhawani, O. Pawar, and S. Barve, "Interview Bot with Automatic Question Generation and Answer Evaluation," in *Proc. 2023 9th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Coimbatore, India, 2023, pp. 1279–1286.

[3] M. D. Pickard and C. A. Roster, "Using computer automated systems to conduct personal interviews: Does the mere presence of a human face inhibit disclosure?," *Comput. Human Behav.*, vol. 105, Art. no. 106197, 2020.

[4] M. Yusuf and K. M. Lhaksmana, "An Automated Interview Grading System in Talent Recruitment using SVM," in *Proc. 2020 3rd Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Yogyakarta, Indonesia, 2020, pp. 34–38.

[5] C. Fang, N. Markuzon, N. Patel, and J.-D. Rueda, "Natural Language Processing for Automated Classification of Qualitative Data From Interviews of Patients With Cancer," *Value Health*, vol. 25, no. 12, pp. 1995–2002, 2022.

[6] Z. Jiang et al., "Multimodal Mental Health Digital Biomarker Analysis From Remote Interviews Using Facial, Vocal, Linguistic, and Cardiovascular Patterns," *IEEE J. Biomed. Health Inform.*, vol. 28, no. 3, pp. 1680–1691, Mar. 2024.

[7] E. P. Beggrow, M. Ha, and R. H. Nehm, "Assessing Scientific Practices Using Machine Learning Methods: How Closely Do They Match Clinical Interview Performance?," *J. Sci. Educ. Technol.*, vol. 23, no. 2, pp. 160–182, 2014.

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 3, 2025

[8] S. Haq, P. J. Jackson, and J. Edge, "Speaker-dependent audio-visual emotion recognition," in *Proc. AVSP*, 2009, pp. 53–58.

[9] N. Neubauer and K. Dupuis, "Toronto Emotional Speech Set (TESS)," Aging and Communication Lab, Univ. Toronto, 2011. [Online]. Available: https://tspace.library.utoronto.ca/handle/1807/24487

[10] H. Cao et al., "CREMA-D: Crowd-sourced Emotional Multimodal Actors Dataset," *IEEE Trans. Affect. Comput.*, vol. 5, no. 4, pp. 377–390, Oct.–Dec. 2014.

[11] K. R. Scherer and H. G. Wallbott, "International Survey on Emotion Antecedents and Reactions (ISEAR) [Data set]," Swiss Center for Affective Sciences, Univ. of Geneva, 1997.

[12] A. Pal, L. K. Umapathi, and M. Sankarasubbu, "Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering," in *Proc. Conf. Health, Inference, and Learning*, PMLR, 2022.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguist. (NAACL)*, Minneapolis, USA, 2019, pp. 4171–4186.

[14] S. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in *Proc. EMNLP*, 2016, pp. 2383–2392.

[15] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in *Proc. 12th USENIX Conf. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.