HELWAN UNIVERSITY
Faculty of Computing and Artificial Intelligence
Artificial Intelligence Department

AI-Interviewer

A graduation project dissertation by:

Esraa Mohammed Abdelhadi	20210154
Farida Khaled Ali	20210675
Mohammed Tarek Omar	20210802
Muhammad Yasser Abdelmoaty	20210848
Madiha Saeid Farouq	20210889
Hania Ruby Mahmoud	20211029

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computing & Artificial Intelligence at the Artificial Intelligence Department, the Faculty of Computing & Artificial Intelligence, Helwan University

Supervised by:

Dr. Soha Ahmed Ehssan

June 2025

جامعة حلوان
كلية الحاسبات والذكاء الاصطناعي
قسم الذكاء الاصطناعي

نظام إجراء المقابلات باستخدام

الذكاء الاصطناعي

رسالة مشروع تخرج مقدمة من:

إسراء محمد عبدالهادي	٢٠٢١٠١٥٤
فريدة خالد علي	٢٠٢١٠٦٧٥
محمد طارق عمر	٢٠٢١٠٨٠٢
محمد ياسر عبدالمعطي	٢٠٢١٠٨٤٨
مديحه سعيد فاروق	٢٠٢١٠٨٨٩
هانيا روبي محمود	٢٠٢١١٠٢٩

رسالة مقدمة ضمن متطلبات الحصول على درجة البكالوريوس في الحاسبات والذكاء الاصطناعي، بقسم الذكاء الاصطناعي، كلية الحاسبات والذكاء الاصطناعي، جامعة حلوان

تحت إشراف

د. سهى أحمد إحسان

يونيو ٢٠٢٥

«وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ»

(سورة هود ،الآية ٨٨)

« My success is not but through Allah. Upon him I have relied, and to Him I return(88)»

(Surah Hud, Ayah 88)

Abstract

This project addresses the persistent challenges in the hiring process, particularly the inefficiencies, human bias, and subjectivity inherent in traditional interviews. Recruiters often face decision fatigue, inconsistent evaluations, and limited time to properly assess candidates, while job seekers frequently lack access to realistic mock interview experiences, hindering their ability to prepare effectively. These issues called for a scalable, automated, and fair solution to support both recruiters and applicants throughout the evaluation process.

To tackle this, the primary objective of the system was to develop a fully automated, AI-powered interview platform capable of simulating human-like interactions and delivering objective, multimodal assessments. The goal was not only to reduce the burden on recruiters but also to provide meaningful feedback to candidates and improve overall hiring quality.

The methodology involved integrating several cutting-edge AI components into a unified pipeline. At the core, a fine-tuned LLaMA 3.3 (70B) model, enhanced through Retrieval-Augmented Generation (RAG), was used for context-aware, non-repetitive question generation tailored to the job description and candidate CV. The interview session incorporated modules for real-time speech-to-text (Whisper), text-to-speech (Bark, GTTS) delivery, facial emotion recognition, and vocal emotion analysis. These inputs were then processed by a judgment agent based on the Mistral model, which evaluated both the AI-generated answers and candidate responses, assigning qualitative scores and offering reasoning and improvement suggestions to support transparency and fairness.

As a result, the system successfully delivered high-quality, context-relevant interview sessions, with detailed analysis of candidate behavior and performance. The generated reports included emotion trends, engagement levels, and structured feedback, culminating in an AI-supported hire/reject recommendation. The platform met its core objectives by enabling efficient, fair, and explainable assessments, while laying the groundwork for integration with external hiring ecosystems and continued expansion.

Keywords: *Artificial Intelligence, Emotion Analysis, Interview Automation, Judgment Agent, Multimodal Assessment, Question Generation*

ملخص

يعالج هذا المشروع التحديات المستمرة في عملية التوظيف، وخاصة ما يتعلق بعدم الكفاءة، والتحيز البشري، والذاتية المتأصلة في المقابلات التقليدية. غالبًا ما يواجه مسؤولو التوظيف إرهاقًا في اتخاذ القرار، وتقييمات غير متسقة، ووقتًا محدودًا لتقييم المرشحين بشكل فعّال، بينما يفتقر الباحثون عن عمل إلى فرص تدريب حقيقية على المقابلات، مما يؤثر سلبًا على أدائهم واستعدادهم.

استهدف النظام المقترح معالجة هذه المشكلات من خلال تطوير منصة مقابلات آلية بالكامل ومدعومة بالذكاء الاصطناعي، قادرة على محاكاة التفاعل البشري وتقديم تقييمات موضوعية متعددة الوسائط. لا يهدف هذا النظام إلى تخفيف العبء عن مسؤولي التوظيف فحسب، بل يسعى أيضًا إلى تقديم تعليقات فعّالة للمرشحين وتحسين جودة قرارات التوظيف بشكل عام.

اعتمدت منهجية التنفيذ على دمج مجموعة من أحدث تقنيات الذكاء الاصطناعي في إطار عمل موحّد. في جوهر النظام، تم استخدام نموذج LLaMA 3.3 (70B) بعد ضبطه، وتم تعزيزه بتقنية الاسترجاع المعزز بالتوليد (RAG) لتوليد أسئلة مخصصة وغير متكررة بناءً على وصف الوظيفة والسيرة الذاتية للمرشح. أثناء جلسة المقابلة، يتم استخدام وحدات لتحويل الكلام إلى نص(Whisper) ، وتوليد الكلام من النص، وتحليل المشاعر من ملامح الوجه والصوت.تُعالج هذه البيانات بواسطة أداة تقييم مبنية على نموذج Mistral ، يقوم بتقييم الإجابات المثالية المولدة من الذكاء الاصطناعي وإجابات المرشح الفعلية، مع إعطاء درجات وصفية وتوضيح أسبابها، واقتراح تحسينات لضمان الشفافية وتجنّب التحيز.

نتج عن ذلك نظام قادر على إجراء مقابلات عالية الجودة ومرتبطة بالسياق، مع تحليل دقيق لسلوك المرشحين وأدائهم. تتضمن التقارير النهائية أنماط المشاعر، ومستوى التفاعل، وتغذية راجعة منظمة، مما يساهم في إصدار قرار مدعوم بالذكاء الاصطناعي بشأن قبول أو رفض المرشح. وقد حققت المنصة أهدافها الأساسية من حيث الكفاءة والعدالة والشفافية، كما وضعت الأساس لدمجها مع منصات التوظيف الخارجية والتوسع المستقبلي .

الكلمات المفتاحية : *الذكاء الاصطناعي، تحليل المشاعر، آلية المقابلات، أداة التقييم، التقييم متعدد الوسائط، توليد الأسئلة*

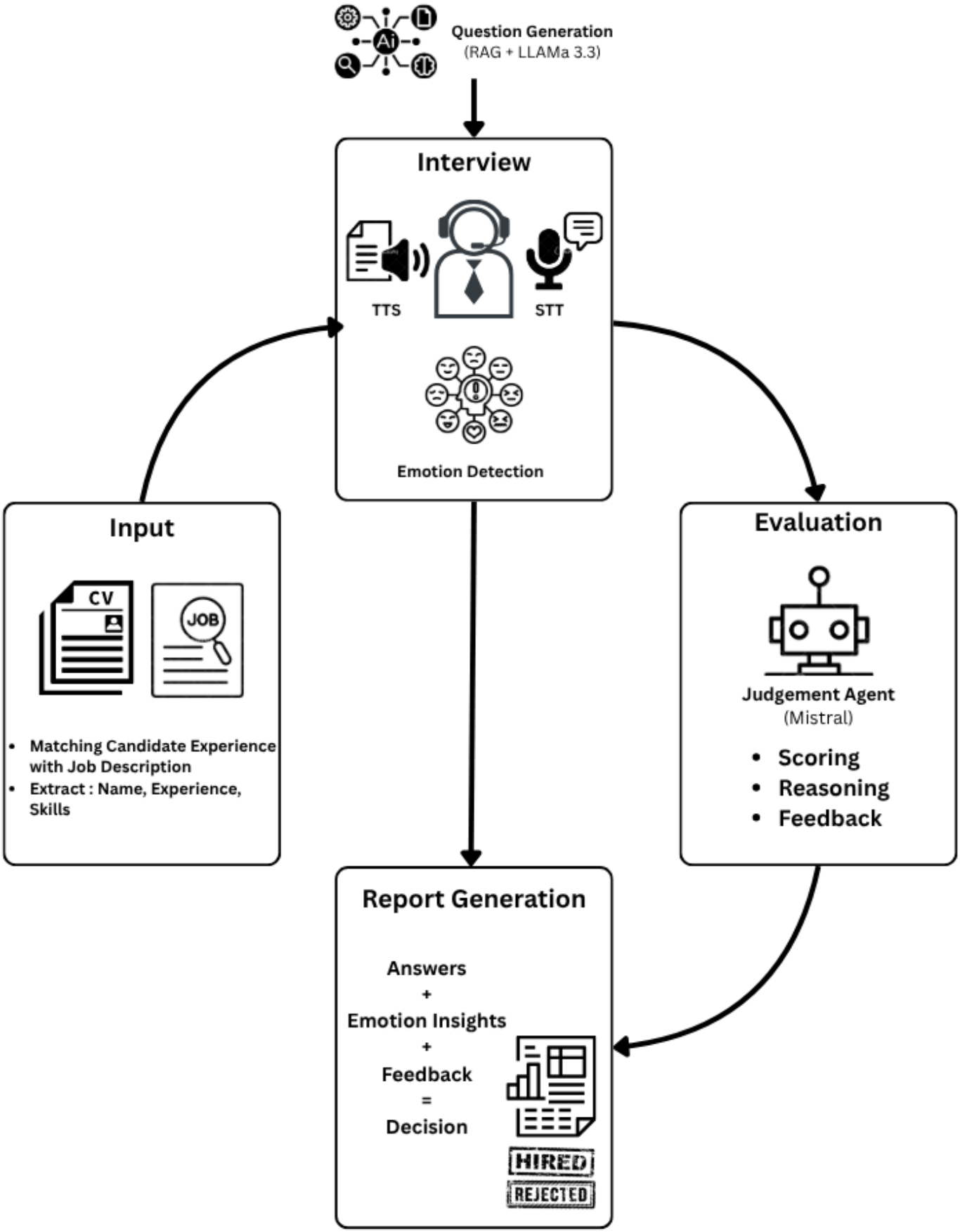


Figure 1: Graphical Abstract

Acknowledgment

We would like to express our sincere gratitude and appreciation to Dr. Soha Ahmed**,** Assistant Professor at the Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt, for her invaluable support, insightful guidance, and continuous encouragement throughout the development of this graduation project. Her expertise, dedication, and constructive feedback were instrumental in shaping the direction of our work and overcoming challenges along the way.

We also extend our heartfelt thanks to everyone who contributed to this project—friends, colleagues, and all those who provided assistance, advice, or motivation during its various stages.

We hope that this work represents a meaningful step forward in our academic and professional journey.

Project Team

Content

Chapter 1 : Introduction

1.1 Background and Overview of the Topic

In today's highly competitive job market, the recruitment process plays a pivotal role in shaping organizational performance. However, traditional interview procedures often suffer from several limitations, including human bias, time inefficiency, and inconsistent evaluation criteria. Recruiters are frequently overwhelmed by the volume of applications and the manual effort required to assess each candidate fairly.

On the other side, many job seekers struggle with limited opportunities for real-world interview practice, which negatively impacts their performance during actual evaluations. This project introduces a fully automated, AI-driven interview assessment platform that aims to redefine the way interviews are conducted. The system simulates a realistic, intelligent interviewer capable of interacting with candidates and providing unbiased, data-informed assessments based on a wide array of inputs, including speech, facial expressions, and textual responses.

By leveraging state-of-the-art AI technologies, such as fine-tuned large language models, emotion recognition systems, and multimodal processing techniques, the platform enables a scalable and objective evaluation framework. The system is designed to cater to a wide range of users—from recruiters in large corporations and startups to job seekers across diverse industries. It generates interview questions based on the candidate's CV and the job description, analyses responses, and delivers detailed post-interview reports. These reports highlight answer quality, emotional patterns, and overall engagement, offering a transparent "hire or reject" recommendation grounded in explainable AI reasoning.

Through this initiative, the project aims not only to enhance hiring efficiency but also to democratize access to structured interview preparation for job seekers. Ultimately, this AI-powered interviewer provides a fair, consistent, and intelligent alternative to conventional recruitment methodologies.

1.2 Problem Statement

The recruitment landscape is facing increasing pressure to become more efficient, scalable, and unbiased. Traditional interview processes are time-consuming, subjective, and heavily dependent on human judgment, which may vary significantly from one interviewer to another. Recruiters often deal with many applications, making it difficult to maintain consistency and fairness in evaluations. Furthermore, human fatigue and implicit bias can unintentionally affect hiring decisions, leading to suboptimal candidate selection and reduced diversity in the workplace.

On the candidate side, many job seekers lack access to realistic interview simulations and constructive feedback, which significantly impacts their preparedness and performance. The absence of a structured mechanism to evaluate soft skills, emotional intelligence, and behavioral cues further complicates the hiring process. Existing AI solutions, when used in isolation (e.g., resume screening or automated question banks), often fail to capture the full scope of a candidate's potential.

Thus, there is a clear need for a comprehensive, automated system that can simulate human-like interviews, analyze multiple input modalities (text, speech, and facial expressions), and provide fair and explainable assessments. Such a solution must not only reduce the resource burden on recruiters but also support candidates with actionable feedback and equitable opportunities.

1.3 Research Question & Objectives

This research project seeks to explore the following key questions:

1. Can a multimodal AI system simulate a realistic, unbiased, and effective job interview experience?
2. To what extent can the integration of emotion analysis and AI-based judgment contribute to fair and transparent candidate evaluations?
3. Does the automation of the interview process enhance recruitment efficiency without compromising the quality and depth of assessment?

To address the above questions, this project sets out to achieve the following objectives:

1. To develop a fully automated, AI-powered interview assessment system that incorporates multimodal data—text, speech, and facial expressions—for comprehensive candidate evaluation.
2. To design a dynamic question generation module using fine-tuned LLaMA models and Retrieval-Augmented Generation (RAG) techniques that tailor questions to specific job descriptions and candidate profiles.
3. To implement speech-to-text, text-to-speech, and emotion analysis modules capable of capturing and interpreting candidates' verbal and non-verbal behaviors during interviews.
4. To integrate a judgment agent that evaluates both AI-generated ideal answers and candidate responses, assigning qualitative scores and justifications to support fair assessment.
5. To generate detailed post-interview reports that include scoring breakdowns, emotional trends, engagement metrics, and transparent hire/reject recommendations.
6. To support job seekers by providing an immersive interview simulation environment and constructive feedback that enhances their readiness for real-world interviews.

1.4 Research Hypotheses

To guide the development and evaluation of the proposed AI-powered interviewing system, the following hypotheses have been formulated:

H1: The integration of large language models (LLMs) with retrieval-augmented generation (RAG) will enable the system to generate more relevant, diverse, and contextually appropriate interview questions than static or prompt-only models.

H2: Multimodal emotion analysis—combining facial and vocal cues—will provide a more accurate understanding of candidate emotional states compared to single-modality analysis.

H3: An AI-based judgment agent, when properly trained and evaluated, can assign interview scores with a level of fairness and consistency comparable to or exceeding that of human interviewers.

H4: Automating the interview process through AI will significantly reduce recruiter workload and improve time-to-hire without degrading the quality of candidate assessments.

H5: Providing candidates with structured feedback and detailed post-interview reports will enhance their self-awareness and readiness for future interviews.

Chapter 2 : Related Work and Literature Review

2.1 Literature Survey

This section provides a thematic breakdown of the key research domains underpinning our system design.

Multimodal Input Processing

- Studies such as [1] emphasize the role of combining audio, visual, and textual data for richer and more human-like machine perception. Methods range from convolutional neural networks (CNNs) for facial recognition to recurrent neural networks (RNNs) and Transformer-based models for speech and text interpretation. Recent advances incorporate cross-modal attention mechanisms, enabling models to jointly reason over different input types for improved decision-making. This multimodal fusion has proven effective in domains like emotion analysis, e-learning, and behavioral assessment.

Emotion and Behavior Analysis

- Emotion detection plays a vital role in human-computer interaction, especially in high-stakes settings like interviews. Literature such as [2] demonstrates that combining micro-expressions with vocal tone significantly improves evaluation accuracy. However, many systems lack real-time emotion tracking or personalized behavioral feedback, which limits the usability of emotional data in actionable decision-making. Furthermore, systems rarely provide transparent explanations or adapt their interpretation based on candidate-specific emotional baselines, resulting in inconsistent evaluation standards.

Automated Question Generation

- Question generation has evolved using large pre-trained language models (e.g., BERT, GPT) capable of creating semantically rich and context-aware prompts. However, most implementations rely on static question sets that fail to respond to candidate behavior or the semantic trajectory of the interview. Our system introduces dynamic difficulty adjustment and job-specific tailoring through a Retrieval-Augmented Generation (RAG)-enhanced LLaMA 3.3 model. This ensures not only linguistic richness and relevance but also adaptability that mirrors human interviewers' situational awareness.

Judgment and Scoring Mechanisms

- Traditional scoring systems are often rule-based or rely on keyword detection, leading to simplistic and frequently biased evaluations [3]. Contemporary studies advocate the use of attention-based architectures, ensemble methods, and human-aligned scoring rubrics. Our system expands on these ideas by integrating a judgment agent based on a fine-tuned Mistral model. This agent evaluates both ideal AI-generated responses and candidate replies using a rubric-informed, transparent reasoning process. It outputs qualitative scores, improvement suggestions, and justification layers—enhancing both accountability and fairness.

Integrated Training and Feedback Platforms

- The majority of current platforms cater exclusively to recruiters, with minimal focus on candidate development. Studies increasingly suggest the importance of systems that offer both parties meaningful support throughout the hiring journey. Our platform is dual-purpose: it enables recruiters to automate fair evaluations while simultaneously offering job seekers a training environment with instant, constructive feedback. This two-sided approach aligns with emerging views on equitable hiring—treating recruitment as a collaborative, iterative, and developmental process.

2.2 Analysis of the Related Work

The rapidly evolving landscape of AI-driven interview systems reflects a significant push to overcome the inherent limitations of traditional recruitment, such as human bias, time inefficiency, and subjective evaluations. Despite notable progress in automating various facets of the hiring process, most existing solutions still grapple with critical deficiencies that compromise their reliability, scalability, and fairness. Our proposed AI Interviewer platform is specifically designed to address and bridge these crucial gaps.

For instance, systems like the one described in "Development of an AI-Based Interview System for Remote Hiring" [4] primarily leverage speech and text analysis to assess candidate responses. While these systems represent a foundational step toward automation, their monomodal nature severely limits their effectiveness. They consistently fail to integrate crucial non-verbal cues, such as facial expressions and vocal emotional nuances, which are indispensable for a comprehensive human-like assessment. Moreover, their feedback mechanisms are often superficial, offering minimal actionable insight or transparency in score justification—elements crucial for fair and developmental feedback.

Another prominent example, presented in "Leveraging Multimodal Behavioral Analytics for Automated Job Interview Performance Assessment and Feedback" [5], adopts a stronger multimodal approach by integrating video, audio, and text. However, even these systems typically lack real-time question adaptability based on a candidate's ongoing performance. This limitation prevents dynamic adjustment of interview flow, hinders deeper personalized feedback, and often ends without a definitive or explainable hire/reject recommendation.

Critically, a common limitation across these platforms is their unidirectional design: they primarily serve recruiters while neglecting the preparation and feedback needs of job seekers. This gap leaves candidates without access to realistic mock interviews or actionable insights that could enhance their future performance, creating an imbalance in the recruitment ecosystem.

How Our System Bridges the Gaps

- Our AI Interviewer platform distinguishes itself by offering a balanced and comprehensive solution. It not only automates the evaluation process but also fosters transparency, fairness, and development for both recruiters and candidates. Key innovations include:
- Robust Multimodal Assessment: We integrate speech-to-text (Whisper), facial expression recognition, and vocal tone analysis to create a layered understanding of candidate performance.
- Context-Aware Scoring Agent: Using a fine-tuned Mistral model, our judgment agent delivers qualitative assessments with clear justification, avoiding shallow sentiment analysis or keyword reliance.

- Adaptive Question Generation: A RAG-enhanced LLaMA 3.3 model dynamically tailors questions to job roles, candidate CVs, and in-interview performance metrics.
- AI-Supported Final Decisions: Each session concludes with a structured report and a hire/reject recommendation supported by analytics and rationale.
- Candidate Empowerment Tools: The system offers mock interviews, personalized feedback, and performance tips—transforming the interview from a filter to a learning opportunity.
- Two-Sided Platform Design: Unlike existing one-directional systems, our platform is built to support both the assessor and the assessed, encouraging fairness, inclusivity, and continuous improvement.

2.3: Comparative Summary of Existing Systems

To further clarify the positioning of our proposed system within the current technological landscape, the following table summarizes the strengths and limitations of several AI-based interview platforms. It highlights where each platform excels and falls short, and how our AI Interviewer addresses specific gaps.

System / **Feature**	**Remote Hiring System** [5]	**Multimodal Analytics System [1]**	**Google Interview Warmup [6]**	**HireVue [7]**	**AI Interviewer (Ours)**
Input Modalities	Audio + Text only	Audio + Video + Text	Audio + Text	Video + Audio + (formerly facial)	Audio + Text + Facial video + Vocal cues
Emotion / Behaviour Analysis	Not supported	Basic non-personalized	Not supported	Voice tone + some visual, recently removed facial analysis	Real-time multimodal emotion analysis
Question Generation	Static set	Static set	Expert-curated static questions	Pre-recorded interview set	Dynamic RAG-based tailored to role & performance
Scoring Method	Keyword-based	Partial	No scoring	Automated scoring with limited transparency	Qualitative scoring with explanation via Mistral
Transparency in Scoring	Low	Medium	None	Improved “glass-box” transparency[8]	High (scores + reasons + improvement suggestions)
Hire/Reject Recommendation	No	No	No	Yes AI-Assisted	Yes—with structured reasoning
Candidate Feedback	Superficial	Limited	General insights (e.g., repetition)	General, not personalized	Instinctive, personalized, improvement-oriented
Candidate-focused Training	No	Limited	Yes (practice interviews)	No	Yes—mock interview experience plus feedback loops
Customization / Adaptivity	Low	Medium	No	Medium (role-based presets)	High (RAG + performance-adaptive Q-generation)

Conclusion

In summary, the current literature and existing AI interview systems provide strong foundational components but lack cohesion, adaptability, and fairness when viewed holistically. Most systems underperform in dynamic reasoning, multimodal integration, and user-centric feedback design. Our AI Interviewer builds on these fragmented insights to present a next-generation, ethically aware, and technically robust solution that not only meets the functional requirements of recruitment but also champions the human elements of empathy, growth, and transparency.

This alignment with broader trends in inclusive hiring, AI fairness, and digital workforce transformation ensures that our system is not only technologically innovative but also socially and ethically responsive—marking a significant step forward in AI-powered recruitment.

Chapter 3: System Design and Architecture

This chapter outlines the high-level design and architectural structure of the AI Interviewer system. It introduces the core system objectives, presents a top-level view of how data flows across components, and describes the modular architecture built to support real-time, multimodal interview assessments. The design prioritizes scalability, integration flexibility, and user realism, leveraging state-of-the-art AI models across natural language processing (NLP), speech processing, and emotion analysis.

3.1 System Overview and Design Objectives

The AI Interviewer is designed to conduct context-aware, multimodal interview sessions that adapt in real time to candidate responses. The system combines language models, emotion recognition, and scoring components to replicate the logic and flow of human-led interviews. It aims to provide fair, engaging, and analytically rich evaluations for a wide variety of job roles and candidate backgrounds.

Key objectives include:

1. Near real-time speech and video handling
2. Adaptive difficulty based on performance
3. Emotionally aware interaction
4. Support for both human-readable and machine-readable output

The architecture is modular, allowing for future extensibility and deployment across different platforms.

3.2 High-Level System Block Diagram

The following diagram illustrates the high-level architecture and end-to-end data flow of the platform. It captures all major components and their interactions during the four main operational phases: interview setup, live session execution, post-interview analysis, and final report generation.

The architecture integrates user interfaces, backend logic, database storage, and several AI subsystems including question generation, transcription, emotional analysis, and evaluation. Each phase of the interview process is mapped to a clear flow of data and control between the respective services.

The platform supports two user roles — recruiters and candidates — and enables asynchronous as well as real-time communication with AI components. Central data coordination is handled by the backend application, while AI models operate in dedicated pipelines that process input and generate contextual output in real time.

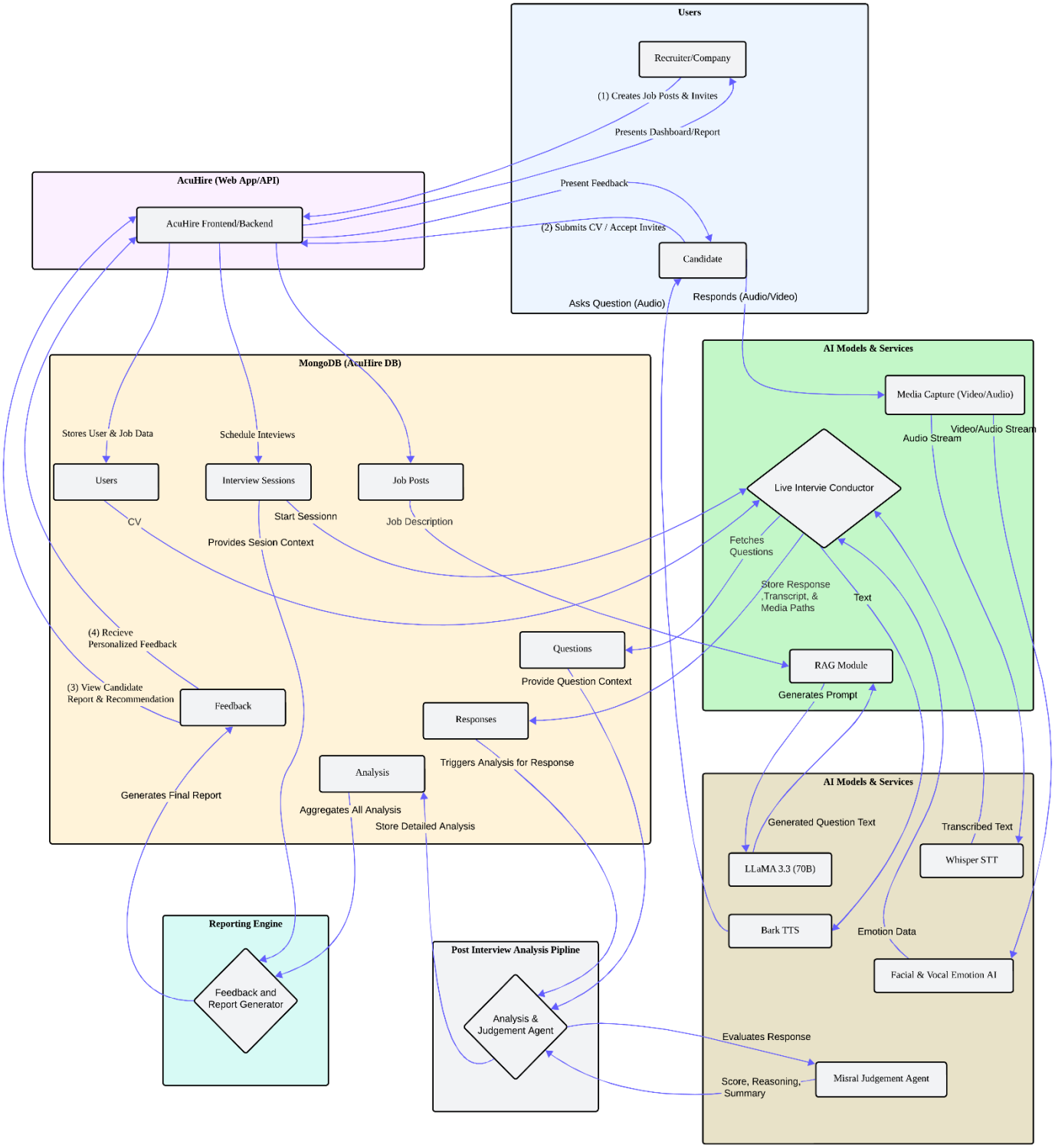


Figure 1: Block Diagram

As shown, the system architecture is modular and event driven. It uses REST APIs and WebSockets for communication, MongoDB for persistent storage, and advanced transformer-based models (such as LLaMA 3.3, Mistral, and Whisper) for core AI functionalities. The architecture is designed for scalability and real-time responsiveness, particularly during live interview sessions.

3.3 Core System Components

3.3.1 Question Generation Engine (RAG + LLaMA)

This module dynamically generates context-aware questions using a Retrieval-Augmented Generation (RAG) pipeline backed by LLaMA 3.3 (70B). It considers the job description and candidate's CV to tailor each prompt. This ensures contextual relevance and diversity in interview questions.

3.3.2 Speech-to-Text Module (Whisper)

The Whisper model enables real-time voice input transcription. It supports natural speech variations and ensures accurate conversion of spoken responses into text, even under less-than-ideal audio conditions.

3.3.3 Text-to-Speech Module (Bark)

Bark synthesizes human-like speech for delivering interview questions. It supports expressive and emotionally nuanced audio output, contributing to a natural and engaging interaction experience.

3.3.4 Voice Emotion Detection

Using a fine-tuned wav2vec 2.0 model, this component identifies emotional cues in spoken responses. The detected emotions (e.g., Happy, Sad, Neutral) provide insight into behavioral patterns.

3.3.5 Facial Emotion Analysis

This module analyzes video frames using the DeepFace framework to detect facial expressions. Frame sampling is applied to optimize performance. Emotion vectors are aggregated to infer dominant facial states over the session.

3.3.6 Judgment and Scoring Agent

The system evaluates candidate responses by comparing them to model-generated reference answers. LLaMA 3.3 is used to produce ideal responses, while Mistral-7B-Instruct scores the user’s answer across four key criteria: relevance, correctness, clarity, and completeness.

To simulate real-world interviewing more closely, the system also incorporates voice and facial emotional signals using a controlled fusion mechanism. These emotion scores are combined with the answer quality score to calculate an effective confidence metric, which contributes to adaptive difficulty scaling. Emotional signals are handled independently and combined after content evaluation to reduce the risk of bias.

3.3.7 Report Generation Module

At the end of the session, a final report is generated summarizing the candidate's technical performance and emotional patterns. This report is constructed using LLMs based on stored evaluations and media analysis.

3.4 Summary of Technologies and Frameworks

Natural Language Processing and Question Generation

LLaMA 3.3 (70B) is used to generate interview questions and reference answers. It was selected based on its contextual precision, low hallucination rate, and robustness in prompt-following tasks across domains.

Mistral-7B-Instruct v0.3 performs response scoring, explanation generation, and feedback synthesis. It was preferred for its balanced performance on evaluation consistency and minimal generation artifacts.

Speech and Audio Processing

Whisper (base) is responsible for converting spoken responses to text in real time. It demonstrates resilience to ambient noise, accent variability, and informal phrasing.

wav2vec 2.0, fine-tuned on emotion-labeled datasets, classifies the emotional state of the speaker. It supports temporal analysis of tone and vocal intensity without requiring transcript input.

Computer Vision and Facial Emotion Recognition

OpenCV handles webcam capture and frame processing, while *DeepFace* performs emotion detection on sampled frames. DeepFace internally applies face alignment, normalization, and expression classification across standard emotion categories.

Text-to-Speech

Bark generates expressive interviewer speech, including non-verbal elements such as pauses and laughter. Its multilingual capabilities and prosodic variety contribute to a more natural interaction experience.

Retrieval and Prompt Contextualization

Qdrant, a vector similarity engine, retrieves role-relevant question–answer pairs and CV text segments. This supports domain-specific grounding for generated questions using a hybrid retrieval-augmented generation (RAG) framework.

Sentence Transformers are employed to embed input job roles and compare them semantically against indexed roles in the database.

Chapter 4: Methodologies

This chapter details the technical procedures, design choices, and evaluation strategies used to build the AI Interviewer system. It follows a modular structure that mirrors the system's key functional components and development workflow.

4.1 Development Process and Component Selection Criteria

Each major component in the system was selected after comparative evaluation based on performance, latency, generalization ability, and suitability for real-time applications.

1. LLaMA 3.3 (70B)

   Used for:

   - Generating interview questions
   - Producing ideal answers for evaluation

   Rationale:

   - LLaMA 3.3 (70B) was selected after extensive testing against LLaMA 2, Flan-T5, and Mistral models It demonstrated superior contextual understanding, higher coherence in long prompt contexts, and reduced hallucination rates. It followed domain-specific prompts more effectively and maintained response quality even with varied prompt structures, making it ideal for professional and adaptive interview environments.

2. Mistral-7B-Instruct v0.3

   Used for:

   - Evaluating candidate responses
   - Scoring based on correctness, relevance, clarity, and completeness
   - Generating reasoning and improvement suggestions

   Rationale:

   - After experimenting with multiple instruction-tuned models, Mistral-7B-Instruct v0.3 produced the most consistent and reliable evaluations. Other models showed higher hallucination risk, especially when scoring ambiguous or partial answers. Mistral generated:
   - Stable, interpretable scoring labels
   - Logical explanations without inventing facts
   - Concise, domain-consistent feedback
   - Its lightweight size and instruction-following capabilities also made it ideal for relatively low-latency deployments.

3. Whisper (base)

   Used for:

   - Real-time speech-to-text transcription

   Rationale:

   - Whisper was selected for its state-of-the-art transcription accuracy, especially in noisy environments and across diverse accents. The model is robust in multi-language scenarios, generalizes well to conversational tone, and handles code-switching.
   - Robust performance with multiple accents and languages.
   - Tolerance to mic and background variability.

- Open-source availability and easy customization.
- Compared to other STT options (e.g., Vosk), Whisper delivered the most accurate and fluid transcriptions during conversational speech, critical for interviews.

4. Bark

   Used for:

   - Expressive, lifelike text-to-speech output

   Rationale:

   - Among evaluated models (e.g. XTTS, Tacotron from Coqui), the Bark model by Suno was selected for its superior voice naturalness and versatility. Compared to other options, Bark produced more expressive, human-like speech with varied intonation and emotion.
   - Key benefits include:
   - Support for emotionally expressive and context-aware speech generation.
   - Capability to synthesize both speech and non-verbal cues (e.g., pauses, laughter) for enhanced realism.
   - Multilingual support and adaptability to different accents and tones.
   - High-quality voice synthesis is essential in this context, as overly robotic or monotonous speech may impair user engagement and degrade the realism of the interview experience.

5. wav2vec 2.0

   Used for:

   - Speech emotion recognition

   Rationale:

   - Fine-tuned on labeled emotional speech datasets (e.g., RAVDESS, TESS, EMO-DB), wav2vec 2.0 provides accurate emotion classification from live audio. It allows the system to detect tone shifts and emotional cues, which are factored into the final interview report.
   - Finetuning:
   - The model was finetuned using huggingface's trainer with weights to address the class imbalance in the accumulation of datasets.

6. DeepFace

   Used for:

   - Facial emotion detection

   Rationale:

   - DeepFace integrates multiple backends (e.g., VGG-Face, Facenet, OpenFace) and supports robust inference under varied lighting and expression conditions. Used in conjunction with OpenCV, it allows:
   - Frame sampling at runtime
   - Emotion classification per frame
   - Aggregation of facial emotion trends over the session
   - By combining DeepFace and wav2vec outputs, the system builds a multimodal emotional profile to complement content-based evaluation.

4.2 Fine-tuning and Training Datasets

Voice Emotion Recognition

Fine-tuned wav2vec 2.0 on:

1. RAVDESS
2. TESS
3. CREMA-D
4. EMO-DB

RAG:

Dataset Overview

To support job-specific and contextually relevant question generation, a custom dataset was compiled from multiple online and offline sources, including job interview preparation websites, books and educational materials. The dataset was manually cleaned and structured to support finetuning and retrieval.

Dataset Statistics

- Total Questions: 3,809
- Unique Job Roles: 26
- Format: JSON records, each with:
  - question
  - answer
  - Job role

Sample Distribution:

Job Role	*Question Count*
Data Science	920
Computer Vision	123
Growth Marketer	111
SEO Specialist	109
Retail Sales Associate	109

4.3 Retrieval-Augmented Generation Pipeline for Question Generation

The system uses Retrieval-Augmented Generation (RAG) to improve contextual relevance in interview questions. It incorporates the retrieval step at the beginning of each interview session:

1. The system queries Qdrant to retrieve top k matches from both the Q&A dataset.
2. If no exact match is found, it uses sentence embeddings + cosine similarity to find the top 3 similar roles.
3. Filters out duplicate questions and returns a clean, context-rich pool of domain-specific questions and answers.

4. A random sample of this context (e.g., 4 Q&A pairs) is passed to the LLM to ground the generated
5. question.
6. This hybrid strategy ensures diversity and relevance, even when job role inputs are misspelled or uncommon.

This helps maintain domain-specificity and realism in interviews even when data for the exact role is sparse.

4.4 Audio and Video Preprocessing Audio

Wav2vec

For voice emotion recognition, we fine-tuned wav2vec 2.0 using the following preprocessing steps:

1. **Resampling**: All audio files were resampled to 22,050 Hz using Librosa to standardize input across datasets.
2. **Feature Extraction**: We extracted MFCCs (Mel-frequency cepstral coefficients) with n_mfcc=40, as they are effective in capturing speech characteristics for emotion classification.
3. **Temporal Normalization**: Each MFCC matrix was padded or truncated to a fixed length of 174-time steps to ensure consistent input dimensions across training samples.
4. **Label Encoding**: Emotion labels were encoded using LabelEncoder and transformed into categorical format using one-hot encoding.

This preprocessing was used to train a classifier on top of wav2vec features for inference during interviews.

Whisper

For real-time transcription of spoken answers, the system uses Whisper (base). To ensure input compatibility and transcription accuracy, the following preprocessing steps are applied before passing audio to the model:

1. **Mono Channel Conversion**: If the recorded audio is stereo, it is downmixed to mono by averaging the channels.
2. **Resampling:** Audio is resampled from its original frequency (typically 48,000 Hz) to 16,000 Hz, which is the optimal input frequency for Whisper.
3. **Waveform Flattening:** The resampled waveform tensor is squeezed into a 1D NumPy array for compatibility with the Whisper processor.
4. **Tokenization and Feature Extraction:** The processed audio is then passed through Whisper's feature extractor (processor) to produce the model-ready input tensors, which are moved to the appropriate device (e.g., GPU) for inference.

This preprocessing ensures Whisper receives clean, normalized audio in the correct format and sampling rate, enabling accurate transcription even under varied hardware or microphone setups.

DeepFace

Facial emotion analysis was implemented using DeepFace, which operates on webcam-captured video during the interview. The following steps were applied:

1. **Frame Sampling:** To reduce computational load while preserving temporal emotion trends, only every 15th frame was processed.
2. **DeepFace applied internal preprocessing**: face detection, alignment, resizing, normalization.

This lightweight, periodic sampling strategy enabled the system to function reliably in real-world hardware settings without requiring GPU acceleration during inference.

4.5 Scoring Logic and Qualitative Evaluation Metrics

The system uses a multi-dimensional scoring mechanism to evaluate user responses in real time. Each response is assessed based on four primary qualitative criteria:

1. **Relevance**: How directly the response addresses the interview question.
2. **Technical Correctness**: Accuracy and appropriateness of the content provided.
3. **Clarity and Depth**: Coherence, articulation, and level of detail.
4. **Completeness**: Whether the response adequately covers the expected aspects.

Each criterion contributes to an overall score, which is categorized into one of four qualitative levels:

- Excellent
- Good
- Medium
- Poor

The evaluation is generated using two models in tandem:

1. LLaMA 3.3 generates a set of ideal answers for comparison.
2. Mistral-7B-Instruct evaluates the user's actual response against these reference answers.

In addition to content-based evaluation, the system incorporates emotional cues—such as vocal tone and facial expressions—into the scoring pipeline using a controlled weighting scheme. Emotional analysis informs a confidence estimation layer that adjusts the final score subtly, without overpowering the content evaluation. This approach allows the system to detect inconsistencies between verbal and non-verbal communication (e.g., confident speech paired with uncertain facial expressions) and reflect them in feedback.

All scores are accompanied by a natural-language explanation, offering suggestions for improvement. These results are stored and used to dynamically adjust the difficulty and focus of upcoming questions, providing a more personalized and adaptive interview experience.

4.6 Fairness Assurance and Bias Mitigation Strategies

Fairness and bias mitigation are integral to the design of the AI Interviewer system. The evaluation framework includes multiple safeguards to promote equitable treatment across candidates, regardless of background, language, or emotional expression:

- **Separated Modal Influence**: Emotional signals (voice and facial) are analyzed independently from linguistic content, with their influence on scoring regulated via explicit weighting. This ensures that emotional expressiveness—whether culturally or individually variable—does not disproportionately affect the final evaluation.
- **Weighted Emotion Integration**: While emotional signals contribute to scoring and follow-up question selection, their influence is secondary to response content. A hybrid confidence adjustment mechanism evaluates the alignment between emotional consistency and answer quality, offering nuanced scoring without bias.
- **Reference Answer Isolation:** Ideal responses are generated prior to user interaction, ensuring that evaluation is blind to user-specific traits and grounded solely in the question's intended context.
- **Adaptive Questioning:** The system dynamically adjusts the difficulty and specificity of questions based on prior performance rather than superficial traits (e.g., accent, speaking speed, or facial demeanor).
- **Semantic Role Generalization**: If role-specific training data is limited, the system selects related roles through semantic similarity rather than hardcoded categories, maintaining fairness in domain adaptation.
- **Transparency and Explainability**: Every response evaluation is paired with human-readable justifications. This promotes accountability and allows users or reviewers to understand the reasoning behind each score.

These strategies collectively ensure that the system balances content, behaviour, and adaptability while minimizing risk of systemic bias—resulting in a more inclusive and equitable interview process.

Chapter 5 : System Analysis

System analysis is a critical phase in the software development lifecycle, as it lays the foundation for understanding the structure, behavior, and requirements of the system. This chapter provides a comprehensive analysis of the proposed AI-powered interview platform. It begins with an overview of the system, outlining its primary functions and goals. Following this, the chapter defines the functional and non-functional requirements that the system must fulfill to meet user expectations and ensure performance, security, and scalability. Subsequently, it presents the system's block diagram to illustrate the major components and their interactions. The chapter concludes with a set of UML diagrams that offer a formal, visual representation of system components, user interactions, data flow, and behavior under different scenarios.

5.1 Overview

The proposed AI Interviewer system is designed as a modular, web-based application that leverages artificial intelligence to conduct automated job interviews and provide multimodal candidate assessments. The platform is built with scalability, extensibility, and ease of deployment in mind, allowing it to serve both recruiters and job seekers in a seamless and responsive manner.

The system architecture follows a full-stack JavaScript model using Next.js (App Router) for both the frontend interface and backend API routing. The backend logic is implemented using TypeScript and Node.js, ensuring type safety and robust request handling. All interview session data, user profiles, and evaluation results are stored in a MongoDB database, selected for its flexibility with unstructured and semi-structured data formats common in AI and user-generated content.

User authentication is managed using bcrypt-based password hashing and JWT (JSON Web Tokens) for session management. Although NextAuth can be optionally integrated for OAuth support, the current implementation emphasizes manual, secure credential handling for full control over authentication workflows.

The system exposes its backend functionalities via RESTful APIs, allowing clear separation between the UI and processing layers. These APIs handle tasks such as user registration, login, session management, question generation, answer submission, and result retrieval. The architecture enables easy integration with external services, such as hiring platforms, cloud storage, or additional ML pipelines.

The frontend, developed using React within the Next.js App Router architecture, provides a dynamic and accessible user interface for both job seekers and recruiters. It supports real-time feedback visualization, structured report viewing, and multimedia interaction (e.g., webcam access, microphone input). The design prioritizes responsiveness, accessibility, and extensibility for future enhancements, such as multi-language support or disability-friendly features.

In summary, the system is a cohesive blend of modern web technologies and AI capabilities, designed to ensure scalability, maintainability, and a seamless user experience across the entire hiring process.

5.2 Functional Requirements

The functional requirements define the core behaviors and operations that the AI Interviewer system must support. These requirements are categorized into key functional domains, each addressing a specific phase of the interview lifecycle—from job posting and candidate setup to AI-driven evaluation and reporting.

FR-1: System Setup and Configuration

- FR-1.1: The system shall allow recruiters to create and manage job postings, including uploading detailed job descriptions.
- FR-1.2: The system shall enable candidates to apply for jobs by creating profiles and uploading their CVs or resumes.
- FR-1.3: The system must extract and parse key entities (e.g., skills, work experience, educational background) from both the job description and the candidate's CV to be used as contextual inputs for the AI components.

FR-2: Dynamic Question Generation Engine

- FR-2.1: The system shall utilize a fine-tuned LLaMA 3.3 (70B) model integrated with a Retrieval-Augmented Generation (RAG) pipeline to dynamically generate interview questions.
- FR-2.2: The generated questions must be tailored to the specific job role, seniority level, and the candidate's background.
- FR-2.3: The engine must ensure that questions are non-repetitive within the same interview session.
- FR-2.4: For each generated question, the system shall also produce a corresponding "ideal answer" to serve as a benchmark during evaluation.

FR-3: Live Interview Session

- FR-3.1: The system shall generate a secure, unique link for each candidate to access their interview session.
- FR-3.2: Questions must be presented using a clear and natural-sounding Text-to-Speech (TTS) voice.
- FR-3.3: The system must record the candidate's video and audio responses in real time using their webcam and microphone.
- FR-3.4: The interface shall include intuitive controls for starting the interview, navigating between questions, and submitting the session.

FR-4: Real-time Multimodal Data Processing

- FR-4.1: The system shall transcribe the candidate's spoken responses using a fine-tuned Whisper model with minimal latency.
- FR-4.2: Facial emotion recognition must be applied to the video stream to detect expressions such as stress, confidence, or confusion.
- FR-4.3: Vocal emotion analysis shall be performed on the audio stream to assess cues like tone, pitch, and pacing.
- FR-4.4: All data streams (transcription, audio, video, emotion metrics) must be synchronized and timestamped for coherent analysis.

FR-5: AI-Powered Evaluation (Judgment Agent)

- FR-5.1: A Mistral-based judgment agent shall compare the candidate's transcribed response with the AI-generated ideal answer.
- FR-5.2: The agent must assign a qualitative score (e.g., Excellent, Good, Medium, Poor) for each question.
- FR-5.3: A concise, written rationale shall accompany every score to ensure explainability and transparency.
- FR-5.4: The agent shall also suggest personalized improvement tips for each answer, which can be included in the feedback report.

FR-6: Reporting and Feedback

- FR-6.1: Once the interview concludes, the system shall automatically compile a detailed report for the recruiter.
- FR-6.2: The recruiter's report shall contain:
    - An overall candidate performance score.
    - A breakdown of question-wise scores with rationales.
    - A visual timeline or summary of emotion trends and engagement signals.
    - A final AI-assisted hire/reject recommendation.
- FR-6.3: A separate candidate-facing report shall be generated, offering constructive feedback on performance and actionable suggestions for improvement.

5.3 Non-Functional Requirements

The non-functional requirements (NFRs) specify the quality attributes, constraints, and operational criteria that the AI Interviewer system must satisfy to ensure robustness, usability, security, and scalability. These requirements address system behavior beyond core functionalities, underpinning user satisfaction and system sustainability.

NFR-1: Performance

- NFR-1.1 (Latency): The speech-to-text transcription component shall exhibit a maximum delay of 3 seconds to preserve the natural flow of the interview dialogue.
- NFR-1.2 (Response Time): The user interface shall maintain responsiveness at all times, ensuring page loads and transitions are complete within 2 seconds.
- NFR-1.3 (Processing Time): Generation of the final interview report shall not exceed 10 minutes following the conclusion of an interview session.

NFR-2: Reliability & Availability

- NFR-2.1 (Uptime): The platform is required to achieve an operational availability of at least 99.5% to minimize disruptions during scheduled interviews.
- NFR-2.2 (Fault Tolerance): The system shall tolerate transient interruptions such as brief network disconnects, enabling candidates to resume interview sessions seamlessly where applicable.
- NFR-2.3 (Data Integrity): All recorded media (video and audio) and analytical data (scores, evaluation results) must be stored with safeguards to prevent corruption or data loss.

NFR-3: Usability

- NFR-3.1 (Candidate Experience): The candidate-facing interface must be intuitive and minimalist, requiring no prior technical training. A pre-interview hardware diagnostic (microphone and camera check) shall be integrated.
- NFR-3.2 (Recruiter Experience): The recruiter dashboard shall facilitate efficient management of job postings, application reviews, and access to candidate reports through a clear and streamlined UI.

NFR-4: Security and Privacy

- NFR-4.1 (Data Encryption): All sensitive data, including personally identifiable information (PII), CVs, and video recordings, must be encrypted during transmission using TLS 1.3 and at rest using AES-256 encryption standards.
- NFR-4.2 (Access Control): Role-Based Access Control (RBAC) mechanisms shall be implemented to ensure that only authorized recruiters can access candidate data relevant to their respective job postings.
- NFR-4.3 (Data Privacy): The system must comply with prevailing data privacy regulations such as GDPR, incorporating mechanisms to support data deletion requests and consent management.

NFR-5: Scalability

- NFR-5.1 (Concurrent Sessions): The architecture must support at least 100 concurrent interview sessions without degradation in system performance or user experience.
- NFR-5.2 (Independent Scaling): AI inference services (e.g., LLaMA, Mistral models) shall be deployed on scalable infrastructure independent of the core web application, to dynamically adjust computational resources based on load.

NFR-6: Fairness & Ethics (Explainable AI)

- NFR-6.1 (Bias Mitigation): The AI models responsible for question generation and candidate evaluation shall undergo rigorous auditing and testing to minimize bias related to gender, ethnicity, accent, or other demographic variables.
- NFR-6.2 (Transparency): The judgment agent shall provide clear, human-readable scoring rationales directly linked to candidate responses, enhancing transparency and fostering trust in AI evaluations.

NFR-7: Maintainability & Extensibility

- NFR-7.1 (Modularity): The system architecture shall adopt a modular design (e.g., microservices) enabling independent updates or replacement of components—such as emotion analysis modules—without impacting overall system functionality.
- NFR-7.2 (External Integration): APIs shall be designed to support future integration with external Applicant Tracking Systems (ATS) and other HR platforms, facilitating ecosystem interoperability and extensibility

5.4 UML Diagrams

5.4.1 Use Case Diagram

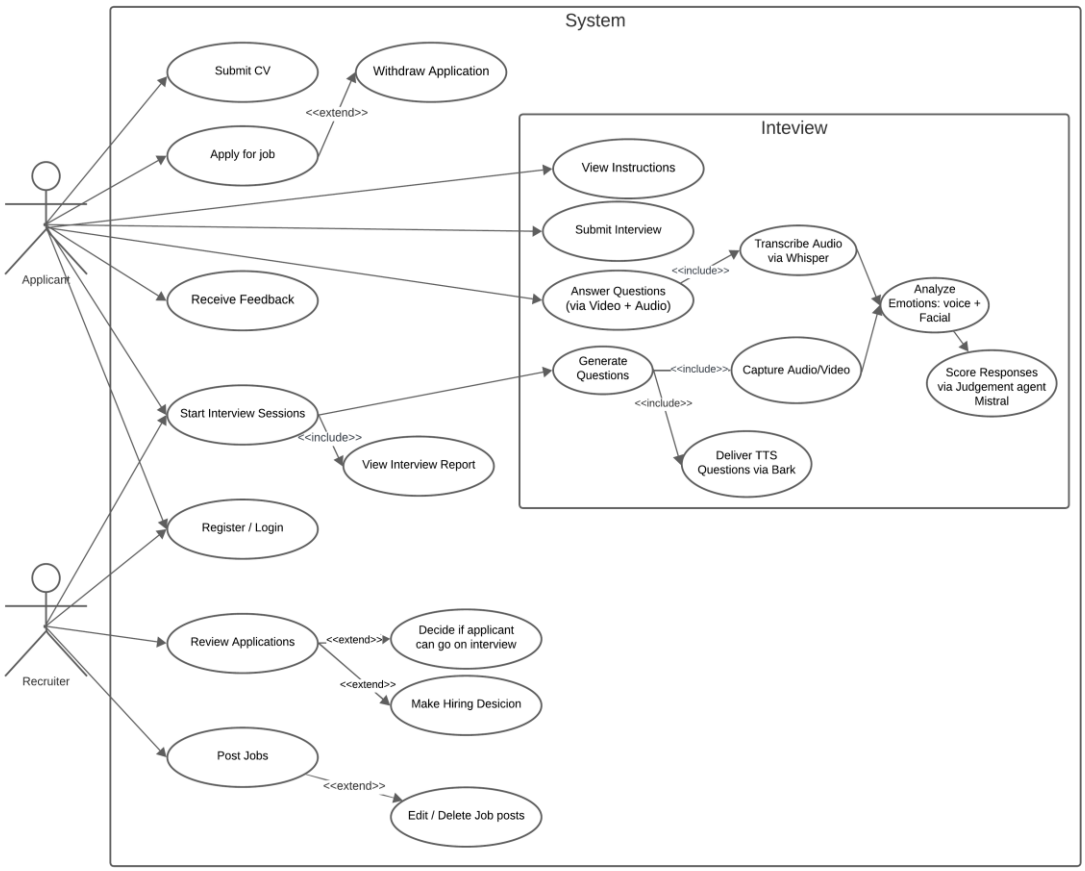
The Use Case Diagram below represents the primary interactions between the system and its external actors, focusing on the functionalities available to both Applicants and Recruiters. The system is conceptually divided into two main functional areas: the Core Recruitment System and the Interview Subsystem, which encapsulates AI-driven assessment processes.

Actors:

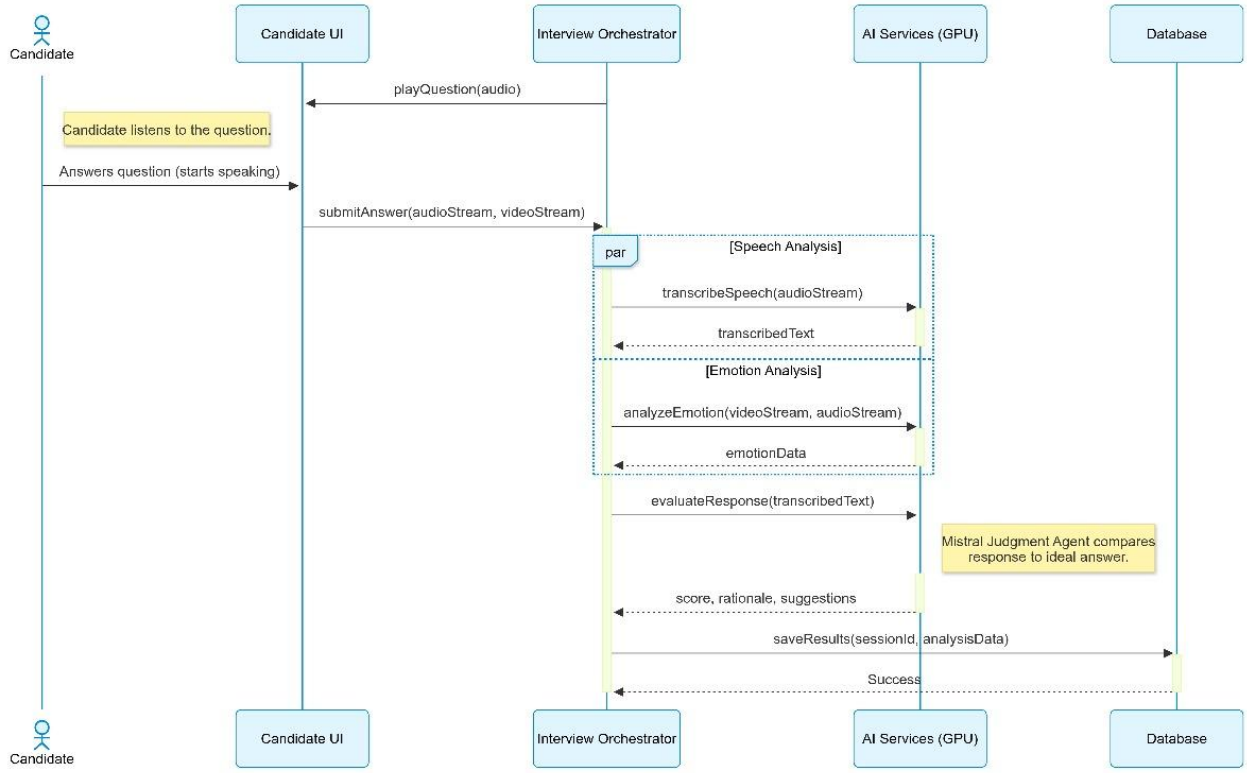
- Applicant: Engages with the system to register, submit applications, attend interviews, and receive feedback.
- Recruiter: Manages job postings, reviews candidate applications, monitors interview outcomes, and makes hiring decisions.

Diagram Structure & Rationale:

- The diagram is intentionally modular:
- Main system Use Cases covers applicant and recruiter actions.
- Interview subsystem captures internal AI automation, which supports the main interview workflow.
- This structure allows for scalability, as the AI modules (e.g., transcription, emotion analysis) can be upgraded or swapped independently.
- The flow provides clear visibility into how a user progresses from application submission to final evaluation.



5.4.1 Sequence Diagram

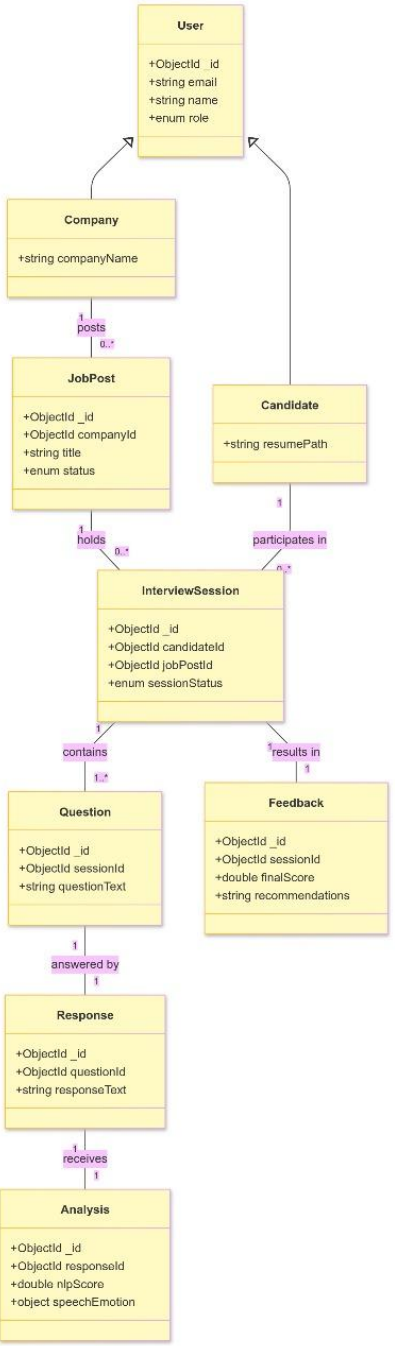


This diagram illustrates the end-to-end process that occurs after a candidate receives a question in the AI Interviewer system:

1. Candidate Interaction:
    - The candidate listens to the question through the UI and begins answering. Audio and video streams are captured in real time.
2. Parallel Processing:
    - Upon receiving the response, the Interview Orchestrator initiates two parallel processes:
        - Speech Analysis: The audio stream is transcribed using a speech-to-text model (e.g., Whisper).
        - Emotion Analysis: Facial and vocal cues are analyzed using dedicated emotion recognition models (e.g., DeepFace and wav2vec).
3. Response Evaluation:
    - The transcribed response is passed to the Mistral Judgment Agent, which compares it to an ideal answer generated earlier by LLaMA 3.3.
    - The system combines the content evaluation with emotion analysis signals to compute a weighted score, rationale, and suggestions for improvement.
4. Persistence and Feedback:
    - The results are stored in the database and sent back to the Candidate UI.
    - These results are also used to adjust the difficulty or focus of future questions in the session.

This workflow ensures that responses are assessed fairly and holistically by incorporating both content quality and emotional delivery.

5.4.1 Class Diagram



The class diagram presented above models the core entities and relationships involved in the automated AI-based interview system. It reflects a domain-centric abstraction for managing users, interviews, job postings, and evaluation data. Below is a detailed description of the involved classes and their interactions:

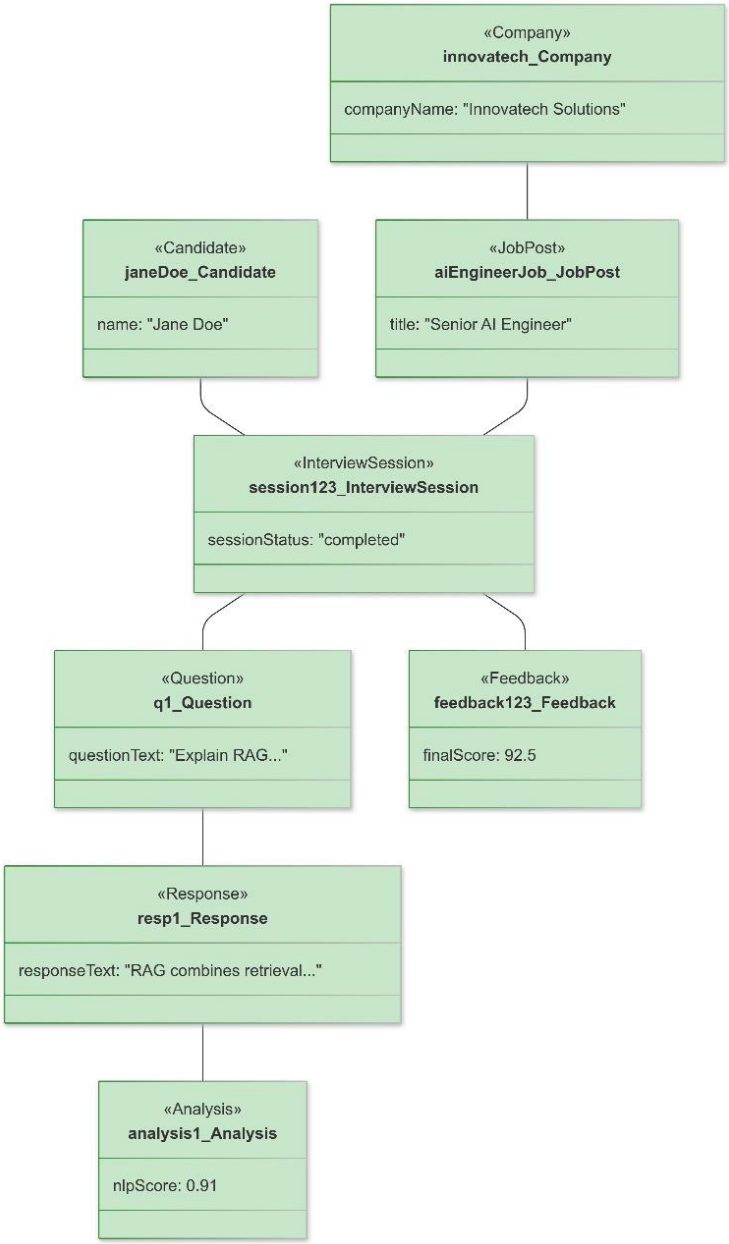
1. **User**
   - Represents the primary entity in the system, encompassing both administrative users (e.g., company recruiters) and applicants. Each user is defined by an identifier, email, name, and role (enumeration). Role-based logic governs access and functionality across the platform.
2. **Company**
   - Denotes an organization associated with one or more users. It holds a one-to-many relationship with JobPost, where each company can create multiple job postings.
3. **Job Post**
   - Represents a job opportunity created by a company. It includes a unique identifier, a reference to the posting company, a descriptive title, and a status indicator (e.g., active, inactive). Each job post may be linked to multiple InterviewSession instances.
4. Candidate
   - A specialized extension of a User, including additional metadata such as the path to the submitted resume. Each candidate may participate in multiple interview sessions associated with various job posts.
5. Interview Session
   - Acts as the central unit of an interview instance, associating a candidate with a specific job post. It includes a status flag (e.g., pending, completed) and serves as the container for all questions, responses, and final feedback. A session contains multiple Question objects and results in one Feedback object.
6. Question
   - Model's individual questions asked during the interview session. Each question is associated with one session and contains a text field representing the question content.
7. Response
   - Captures the candidate's spoken or written answer to a specific question. It is directly linked to one Question and stores the transcribed response text.

8. Analysis
    - Represents the analytical layer applied to candidate responses. Each response receives a single analysis object containing quantitative metrics such as natural language processing scores and multimodal emotion scores derived from facial and vocal inputs.
9. Feedback
    - Summarizes the outcome of an interview session. It includes a final score and qualitative recommendations generated by the evaluation pipeline.

5.4.1 Object Diagram

The object diagram provides a snapshot instance of how actual data objects relate to each other at runtime within the system. The following scenario outlines a typical use case:

1. A user with recruiter role is associated with a company named **“Innovtech”**.
2. The company has posted a job titled **“AI Engineer”**.
3. A candidate applies to the job and initiates an interview session.
4. During the session:
5. A question such as *“Explain RAG…”* is posed.
6. The candidate responds with a transcribed answer.
7. The system analyzes the response using both natural language and emotional metrics.
8. Based on the analysis:
9. A final feedback report is generated with a normalized score and tailored suggestions.



This object diagram illustrates real-world instantiations of the abstract classes, showcasing the lifecycle of an AI-assisted interview from job posting to final feedback.

Chapter 6: System Implementation

This chapter presents the technical implementation of the AI Interviewer system. The system was developed locally using Python and integrates multiple machine learning components and tools for real-time question generation, speech and emotion processing and response evaluation. The implementation focuses on modularity, maintainability, and compatibility with commonly available local hardware.

6.1 System Overview and Development Environment

The system was designed for local deployment and tested on machines equipped with consumer-grade CPUs and GPUs. The full-stack architecture comprises: A Python-based backend for handling AI components. Development tools and environments included: Visual Studio Code for development Kaggle, Google Colab Notebooks for model training and prototyping. GitHub for version control and collaboration

6.2 Technology Stack

Language Models: LLaMA 3.3 (70B) used for generating interview questions and ideal answers Mistral-7B-Instruct used for scoring user responses, generating reasoning, and suggestions

Speech Processing: Whisper (base) model for real-time speech-to-text transcription Bark model from Suno for text-to-speech synthesis

Emotion Recognition: Fine-tuned wav2vec 2.0 for vocal emotion classification DeepFace (with OpenCV) for facial emotion recognition

Retrieval Engine: Qdrant as a vector similarity engine for role-based Q&A context retrieval

6.3 Component-Level Implementation

6.3.1 Question Generator Loop

The core logic of the interview system resides in the interview_loop() function. This loop handles prompt building, question generation, response capture, answer evaluation, and dynamic difficulty adjustment. Role-specific context is retrieved using retrieve_interview_data() The build_interview_prompt() function constructs a formatted string with job role, seniority, retrieved context, last user answer, and previous score The question is generated using groq_llm.predict() with LLaMA 3.3 Difficulty adjustments are made based on previous response scores

6.3.2 Audio Processing Modules

Speech-to-Text (Whisper): Audio is captured from the user's microphone Converted to mono, resampled to 16kHz using Torchaudio’s Resample function Processed into Whisper-compatible input via the HuggingFace processor

Text-to-Speech (Bark): Each generated question is passed to Bark for vocal synthesis Bark automatically includes expressive prosody and non-verbal vocalizations such as pauses or emphasis

6.3.3 Emotion Recognition Modules

Voice Emotion (wav2vec 2.0): Trained using MFCC features extracted with Librosa (n_mfcc=40) MFCC matrices are padded or truncated to a fixed temporal size Final training set includes samples from RAVDESS, TESS, CREMA-D, and EMO-DB During inference, audio is preprocessed similarly and passed to the trained model

Facial Emotion (DeepFace): Video is captured using OpenCV from the webcam Every 15th frame is analyzed for dominant emotion The frame is passed to DeepFace's emotion analysis backend Emotions are aggregated over the session to compute an overall emotional trend

6.3.4 Scoring Agent

First, an ideal answer is generated for each interview question using LLaMA 3.3. The user's actual response is then evaluated by Mistral-7B-Instruct, which assesses the answer along four key qualitative dimensions: • Relevance • Correctness • Clarity • Completeness • Emotion

Each evaluated response is assigned a categorical score (Excellent, Good, Medium, Poor), along with a natural-language explanation and suggested areas for improvement. This evaluation is stored in the session and used to guide the question generator. In addition to content-based assessment, the system integrates emotional signals—vocal tone and facial expression—into the scoring pipeline to estimate the candidate's effective confidence. The emotional classes (e.g., "happy", "neutral", "nervous") are first mapped to normalized scores using a fixed emotion_map. The system then computes an effective confidence score using a weighted fusion: effective_confidence = (0.5 * answer_score + 0.22 * voice_score + 0.18 * face_score + 0.1 * control_bonus)

Where:

- **answer_score** is a float mapped from the categorical grade (e.g., 1.0 for Excellent)
- **voice_score** and **face_score** are derived from detected emotional states
- **control_bonus** is added when content confidence exceeds emotional expression, rewarding emotional regulation

This metric is not directly shown to the user but plays a critical role in the adaptive logic, influencing the difficulty level of upcoming questions.

6.3.5 Pseudocode and Interface Summary

1. ***HuggingFace Tokenizer and Judge Pipeline (Mistral)***
    - Purpose: Loads the Mistral-7B-Instruct model for local scoring of answers.
    - judge_pipeline:
        - Input: prompt (string) containing both user answer and reference answer
        - Steps:
            - Generates a natural-language evaluation (e.g., Score, Reasoning, Improvement Suggestion)
        - Output: list of generated token strings (evaluations)

2. ***LocalEmbeddings Class***

- Purpose:
    - Generates dense embeddings for semantic similarity using SentenceTransformer.
- embed_query(text)
    - Input: text (string)
    - Output: embedding vector (list of floats)
- embed_documents(documents)
    - Input: list of strings
    - Output: list of embedding vectors

3. ***CohereReranker Class***
    - Purpose:
        - Improves context quality by reranking top relevant documents from Qdrant.
        - compress_documents(documents, query)
    - Input:
        - documents: list of Document objects with `.page_content`
        - query: input string (e.g., job role or question)
    - Steps:
        - Use Cohere's rerank API to get relevance scores
        - Return top 5 most relevant documents
    - Output: top 5 document objects (reranked list)

4. ***EvaluationScore:*** Enum mapping of 4 scoring levels (Poor, Medium, Good, Excellent)

5. ***load_data_from_json(file_path) function***
    - Purpose: Load a JSON dataset of interview questions and answers, grouped by job role.
    - Input: file_path (str) – Path to JSON file
    - Steps:
    - Open JSON file and parse it.
    - For each item:
    - Extract job role, question, and answer.
    - Group entries into a dictionary {job_role: [QA pairs]}
    - Return grouped data.
    - Output: Dictionary[str, List[Dict]] – job role mapped to list of Q&A pairs

6. *verify_qdrant_collection(collection_name) function*

- Purpose: Check if a Qdrant collection exists.
- Input: collection_name (str)
- Steps:
  - Call Qdrant client to retrieve collection info.
  - If exists, return True.
  - If not found or error, return False.
- Output:
  - Boolean – True if collection exists, False otherwise

7. *store_data_to_qdrant(data, collection_name, batch_size) function*

- Purpose: Embed and upload all Q&A entries to Qdrant.
- Input:
  - data (dict) – {job_role: [Q&A pairs]}
  - collection_name (str)
  - batch_size (int)
- Steps:
  - 1. If Qdrant collection doesn't exist, create it with 384-dim COSINE vectors.
  - 2. For each job role and Q&A pair:
    - Embed the question using SentenceTransformers.
    - Create a PointStruct with vector and payload.
  - 3. Upload in batches using `qdrant_client.upsert()`
  - 4. Log how many were stored and verify with `qdrant_client.count()`.
- Output: Boolean – True if successful

8. *find_similar_roles(user_role, all_roles, top_k=3) function*

- Purpose: Find top-k job roles similar to the user input using cosine similarity.
- Input: user_role (str), all_roles (List[str]), top_k (int)
- Steps:
  - Embed the user_role string.
  - Embed all roles from the list.
  - Compute cosine similarity.
  - Return top-k most similar roles.
- Output: List[str] – Top-k most similar job roles

9. *get_role_questions(job_role) function*

- Purpose: Retrieve all Q&A pairs in Qdrant for a specific job role.
- Input: job_role (str)
- Steps:
  - Apply a scroll filter where payload.job_role == job_role
  - Paginate results (limit=100)
  - Parse payload fields into a list of dictionaries
- Output: List[Dict] – All Q&A pairs for the role

10.retrieve_interview_data(job_role, all_roles) function

- Purpose: Return Q&A data for an exact or similar job role.
- Input: job_role (str) all_roles (List[str])
- Steps:
    - Normalize job_role string
    - Try exact match: get_role_questions(job_role)
    - If empty: a. Use find_similar_roles() to find top 3 similar roles b. Call get_role_questions() on each one c. De-duplicate questions
    - Return merged list of Q&A dicts
- Output: List[Dict] – Full Q&A set for role or similar roles

11.random_context_chunks(retrieved_data, k)

- Purpose: Randomly select k Q&A entries and format them as prompt context.
- Input: retrieved_data (List[Dict]) k (int)
- Steps:
    - Randomly sample k entries
    - Format each as "Q: ... A: ..."
    - Join all with double newlines
- Output: str – Prompt-ready formatted context

12. eval_question_quality(question, job_role, seniority, judge_pipeline)

- Purpose: Evaluates the quality of an interview question using a language model.
- Input: question(str), job_role(str), seniority(str), judge_pipeline(pipeline)
- Steps
    - Check Pipeline Availability:
    - If no pipeline is passed and none exists globally, return an error message.
    - Construct Evaluation Prompt:
    - Include a scoring rubric for the model.
    - Define the job role and seniority.
    - Include examples of Poor, Medium, and Excellent questions with JSON-style evaluations.
    - Append the target question at the end of the prompt.
    - Run the Evaluation:
        - Use judge_pipeline() to generate a response to the prompt.
        - Set temperature=0.1 and do_sample=False for consistency.
        - Apply repetition penalty and a token limit
        - Extract JSON Output:
            - Use regex to locate a JSON block from the model's output.
            - Parse it and validate it contains Score, Reasoning, and Improvements.
            - Return Result:
        - If everything succeeds, return a cleaned dictionary.
        - If something fails (missing keys, parse errors), return a fallback structure with "N/A" or "Error".
- Outputs:Score, reasoning, Improvements

13. generate_reference_answer() function

- Inputs: question, job_role, seniority
- Steps:
    - Construct a simple prompt: `"You are a {seniority} {job_role}.\nQ: {question}\nA:"`.
    - Call `groq_llm.predict()` (LLaMA 3.3 via Groq API) with the prompt.
    - If output is empty, return a fallback message.
    - Clean and return the generated answer.
- Output: answer

14. evaluate_answer function

- Input: question, answer, ref_answer, job_role, seniority, judge_pipeline
- Steps
    - Check if judge_pipeline is passed or fallback to a global variable.
    - Build a structured prompt describing the evaluation task and including:
        - The question
        - Candidate's answer
        - Reference answer
        - Scoring instructions
    - Run the pipeline (judge_pipeline) on the prompt.
    - Extract a valid JSON response from the output with:
        - "Score": qualitative rating
        - "Reasoning": justification
        - "Improvements": actionable suggestions
    - Return parsed results or fallback fields on error.
- Output: Score, Reasoning, Improvements

15. build_interview_prompt() function

- Input: conversation_history, user_response, context, job_role, skills, seniority, difficulty_adjustment
- Steps
    - Determine the difficulty setting description based on difficulty_adjustment.
    - Build a full system prompt:
        - Includes baseline difficulty rules.
        - Uses RAG-based or fallback context.
        - Embeds structured role, skills, and last user input.
        - Appends evaluation feedback from the last turn if available.
        - Format the last 6 exchanges of history into readable format ("Interviewer: ...").
        - Fill out and return the prompt.
- output: Prompt

16. generate_llm_interview_report(interview_state, job_role, seniority)

- Purpose: Generate a structured final report summarizing the candidate's performance based on the recorded interview session.
- Input: interview_state, job_role, seniority
- Steps
    - Iterate over all questions stored in interview_state["questions"].
    - Extract the question, user's answer, score, and reasoning.
    - Format them into a readable transcript using string formatting.
    - Construct a prompt using that transcript and the job role/seniority.
    - Pass the prompt to groq_llm.predict() to generate the summary report.
    - Return the full report as a string.
- output: report

17. interview_loop(max_questions, timeout_seconds, collection_name, judge_pipeline) function

- Purpose: Conduct a full adaptive interview loop, generating questions, collecting responses, evaluating them, and storing session data.
- Input: max_questions, timeout_seconds, collection_name, judge_pipeline
- Steps
    - User Input & Role Setup
        - Collect job_role, seniority, and skills from the user.
        - Extract all job roles from Qdrant and retrieve role-matching Q&A data.
        - Format 4 samples into context using random_context_chunks.
    - Initialize State
        - Create empty conversation_history and interview_state for session tracking.
        - Set difficulty_adjustment = None
    - Loop Over Questions
        - For each round i up to max_questions:
        - Build prompt using build_interview_prompt()
        - Generate a new question using groq_llm.predict()
        - Evaluate the question quality using eval_question_quality()
        - Print and store the question
        - Wait for a user response using wait_for_user_response(timeout)
        - If skipped: mark as skipped and continue
        - Generate ideal answer using generate_reference_answer()
        - Evaluate user answer using evaluate_answer()
        - Store evaluation results and update conversation_history
        - Adjust question difficulty based on current score
    - Finalize Session
        - Store interview end_time
    - Return interview_state (which includes all data for report generation)
- output: interview_state(dict):Full session log with questions, answers, and evaluations.

6.4 Testing and Validation

Testing was performed at multiple levels: Unit Testing: Functions such as prompt building, answer evaluation, and audio processing were tested with controlled inputs Integration Testing: The full interview_loop() was tested with simulated inputs to verify transitions, emotion handling, and scoring

Edge Case Handling: Empty responses were skipped but logged Missing microphone input triggered warnings and fallback logic Webcam unavailability was handled gracefully by continuing with audio only

Chapter 7: Results and Discussion

This chapter presents the experiments conducted to evaluate the performance of the AI Interviewer system. The evaluation focuses on emotional recognition accuracy, quality of question generation, and consistency of the Judgment Agent. However, due to the absence of large-scale user testing, HR expertise, or gold-standard ground truth data, all performance analyses are qualitative or manually interpreted. Therefore, results reflect best-effort approximations based on internal observation and review.

7.1 Experimental Setup

The system was evaluated using a combination of simulated inputs and real-time candidate responses in controlled environments.

Hardware Configuration: Intel Core i7 CPU |16 GB RAM | NVIDIA GTX 1650 GPU

Software and Libraries: Python 3.10, PyTorch 2.0, HuggingFace Transformers 4.36+, OpenCV 4.8, DeepFace 2023

Datasets Used for Emotion Speech Recognition:

- RAVDESS, TESS, CREMA-D, EMO-DB

Interview Question Dataset: A custom JSON-formatted dataset scraped and cleaned from public job interview platforms. It includes 3,809 total questions across 26 unique job roles. Sample Role Distribution:

- Data Science: 920 questions
- Computer Vision: 123 questions
- Growth Marketer: 111 questions
- SEO Specialist: 109 questions
- Retail Sales Associate: 109 questions

The dataset was split into training and test sets for model fine-tuning and used as context in the RAG pipeline for personalized question generation.

7.2 Results and Analysis

7.2.1 Accuracy of Emotion Recognition

The fine-tuned wav2vec 2.0 model achieved approximately 85% classification accuracy across seven emotion classes (happy, sad, angry, fear, disgust, surprise, neutral) based on test set performance.

The DeepFace framework was validated qualitatively. It consistently identified dominant facial emotions across a range of lighting conditions and facial orientations.

7.2.2 Question Relevance and Diversity

Multiple versions of the LLaMA model were tested to evaluate their performance in generating job-specific and coherent interview questions.

LLaMA 2 (7B)

- Generalized well on domain knowledge but showed low diversity in output.
- No hallucinations
- Issue: Repeated identical questions across turns.

Example:

```
Q1: How do you handle conflict in a team?
Q2: Q: How do you handle conflict in a team?
Q3: Q: Q: How do you handle conflict in a team?
```

Finetuning:

- Failed to resolve repetition.
- Even with prompt tuning, the model out duplicated prompts (e.g., “Q: how do you handle conflict in a team?”).

Eval metrics:

- eval_loss: 0.055
- eval_runtime: 309.48s
- samples/sec: 1.37

LLaMA 3.1 (finetuned)

- Overfit the training set.
- Low question diversity: The model almost constantly reused near-identical question templates regardless of the candidate’s role, seniority, or prior responses.

LLaMA 3.3 (70B)

- Selected as the final model.
- Eliminated repetition.
- Delivered adaptive, seniority-aligned questions even with minimal or noisy context.
- Integrated well with RAG to support role-specific question personalization.

Overall Observations:

- Relevance: Strong alignment with job role and seniority in most runs.
- Diversity: Multi-turn sessions yielded unique, progressive questions.
- Specificity: Qdrant-based context injection improved personalization.

7.2.3 Model Behaviour Evaluation Across Interview Dimensions

Evaluation Procedure

To assess the overall behavioral and linguistic quality of the AI interviewer, a series of mock interviews were conducted manually using the system. The resulting transcripts were evaluated by a higher-capability language model (GPT-4) according to a fixed rubric. This rubric covered core aspects of human-like interviewing and rated the model's outputs on a 5-point Likert scale across five key dimensions.

Evaluation Results

Dimension	Score (5)	Comments
Question Relevance	4.9	Highly aligned with backend engineering tasks. Technically deep and situationally accurate.
Contextual Responsiveness	4.7	Follow-ups adapted logically to prior answers. Minor rigidity in transition language.
Evaluation and Scoring	4.8	Scoring was consistent and well-reasoned, though slightly over-generous at times.
Language Coherence	4.6	Formal and precise, but lacked variation and conversational warmth
Behavioural Realism	4.4	Professional tone maintained, but emotionally neutral and robotic at moments.

Aggregate Score and Conclusion of evaluation

The average score across all evaluation axes was **4.68 out of 5**. This suggests the model operates at near-human levels for technical interviewing in backend engineering contexts. The AI interviewer demonstrated high fidelity in replicating real-world technical interviews. Its performance indicates readiness for use in educational simulations, preliminary screening processes, and AI-assisted hiring tools. Future iterations should focus on improving conversational flow and incorporating soft-skills evaluation to enhance realism.

7.3 Summary of Findings

1. The system demonstrates robust multimodal input handling and smooth real-time operation.
2. Emotion detection modules (voice and face) show consistent performance across diverse inputs.
3. Generated questions are generally relevant, varied, and role-appropriate—even under ambiguous input scenarios.
4. Scoring and reasoning outputs appear logically consistent, but lack of HR-validated benchmarks limits claims of evaluative precision.
5. LLaMA 3.3 and Mistral 7B were superior to their predecessors and resolved issues such as repetition and overfitting.

In summary, the system fulfills its intended goals with promising behavior in all tested components. However, external validation by HR professionals remains essential before real-world adoption in recruitment settings.

7.4 Discussion

The experimental results demonstrate that the AI Interviewer system delivers a functional, adaptive interview simulation grounded in both technical performance and behavioral cues. By integrating natural language generation, speech processing, retrieval-based personalization, and emotion recognition, the system provides a structured and explainable interview experience. However, several important technical and design considerations emerge from the evaluation.

Multimodal Fusion and Confidence Estimation

The system combines voice and facial emotion scores with content-based evaluations to generate an effective confidence metric. This fusion approach reflects real-world interviewer behavior, where emotional control and communication style often influence impressions. However, the current implementation relies on fixed emotion-to-score mappings and manually tuned weights. While interpretable, these heuristics have not been empirically validated against real-world recruitment data, which limits their generalizability. Additional research is needed to determine whether such emotional indicators consistently correlate with interview performance across different user demographics and roles.

Question Relevance, Diversity, and Role Adaptation

The LLaMA-based question generator, especially in its final LLaMA 3.3 configuration, demonstrated strong alignment with job roles and seniority levels. Contextual retrieval from Qdrant further improved specificity, with observed benefits in relevance and reduced repetition. The model avoided hallucinations and maintained logical progression in multi-turn interactions.

However, evaluation results indicated occasional rigidity in language transitions and minor context-blind responses in edge cases (e.g., uncommon roles or ambiguous input). The fallback strategy—using similar roles retrieved via semantic similarity—mitigates this partially but does not fully ensure semantic

precision. These outcomes were reflected in the behavior analysis in Section 7.2.3, where the system scored 4.9 in question relevance and 4.7 in contextual responsiveness.

Scoring Fairness and Explainability

The scoring logic—based on comparison to LLaMA-generated reference answers and evaluated by Mistral-7B-Instruct—performed reliably under structured input. Scores were accompanied by reasoning and improvement suggestions, maintaining internal consistency across similar answers and roles.

However, as noted in the behaviour evaluation, scoring was occasionally over-generous. The lack of HR-reviewed benchmarks or demographically labelled data prevents a comprehensive fairness audit. Biases embedded in training data, or differences in cultural expression (especially in emotion-related behaviour), may influence scoring in unintended ways.

Language Coherence and Behavioral Realism

The system maintained formal and precise language throughout interviews. However, it lacked conversational variation and emotional nuance, which can reduce realism during behavioral interviews. The behavior analysis in Section 7.2.3 reflected this: the system scored 4.6 in language coherence and 4.4 in behavioral realism, suggesting room for improvement in tone, empathy simulation, and conversational flexibility. This may be addressed in future work by fine-tuning on more diverse, multi-turn dialogue datasets or incorporating affect-aware generation models.

User Experience and Feedback Interpretation

From a user-cantered design perspective, the system succeeded in delivering structured, clear feedback and adjusting difficulty in real time based on performance. This supports both interviewer simulation and practice-based learning for job seekers. However, some feedback—such as the confidence score—remains internal and not exposed to the user interface.

Improving transparency and interpretability could enhance user trust and learning outcomes. Possible directions include visualizing emotional trends, highlighting behavioural insights, and offering coaching suggestions, especially for candidates who receive lower scores.

Conclusion

Overall, the discussion confirms that the AI Interviewer system offers technically sound and pedagogically valuable simulations. The system meets core design goals, but further validation, user experience improvements, and fairness auditing are necessary before deployment in real-world recruitment or educational environments.

7.5 Limitations and Ethical Considerations

While the integration of emotion recognition enhances realism and depth in candidate evaluation, it introduces important ethical and interpretability challenges. Emotion-based profiling—if not transparently communicated—may be perceived as intrusive or unfair, especially across cultures or neurodiverse users.

Additionally, the absence of expert-reviewed datasets or human-annotated ground truth limits our ability to validate the accuracy and fairness of scoring across diverse scenarios. Without formal bias audits, there remains a risk that the system may unintentionally reinforce subjective patterns or cultural norms encoded in the training data.

To ensure responsible use in real-world applications, any deployment must include:

1. Transparent explanations of how emotion contributes to scoring
2. Optional consent for behavioural analysis
3. Configurable sensitivity thresholds
4. Independent audits of fairness and accuracy

These steps are critical to ensuring that the system supports—not replaces—human judgment in recruitment contexts, and that its decisions remain interpretable, fair, and justifiable.

Chapter 8: Conclusion

The AI-Interviewer project marks a significant step toward transforming conventional hiring methods through the integration of advanced artificial intelligence. With a central aim to address persistent flaws in the traditional recruitment landscape—namely inefficiency, bias, and inconsistency—the system demonstrates how intelligent automation can yield meaningful improvements in both candidate evaluation and recruiter experience.

At the heart of the platform lies a synergistic architecture composed of multiple AI components. A fine-tuned LLaMA 3.3 model with Retrieval-Augmented Generation (RAG) technology powers the dynamic and context-sensitive question generation engine. This ensures that interview prompts are tailored to the specific candidate profile and job requirements, thereby enhancing relevance and depth. The integration of speech-to-text and text-to-speech systems, facial and vocal emotion analyzers, and a judgment agent based on the Mistral model collectively allow the system to interpret and evaluate a candidate's responses across multiple modalities.

These innovations enable not only automation but also a form of *intelligent assessment* that accounts for emotional nuance and behavioral context. The multimodal nature of the platform ensures that important soft skills—often overlooked in automated systems—are recognized and factored into the final decision-making process. By grounding evaluations in explainable AI, the system also ensures transparency, making it easier for candidates and recruiters alike to understand the rationale behind each decision.

The system was rigorously designed to generate comprehensive post-interview reports that offer both quantitative scores and qualitative insights. These reports provide actionable feedback for candidates, helping them understand their strengths and weaknesses, while giving recruiters a high-level overview enriched with emotional and engagement metrics. Such structured insights are a key differentiator, elevating the system beyond simple automation into the domain of decision intelligence.

One of the project's key achievements is its dual-purpose design: it not only supports recruiters by streamlining and objectifying the evaluation process but also serves as an invaluable training tool for job seekers. By offering a realistic and data-informed simulation environment, the system helps candidates refine their communication skills, emotional regulation, and interview strategies—ultimately enhancing their readiness for real-world challenges.

Despite these achievements, the project acknowledges certain limitations that form the foundation for future research and development. Issues such as model generalizability across languages and cultures, deeper temporal emotion tracking, and improved integration with existing recruitment pipelines are important areas to address in subsequent iterations.

In conclusion, the AI-Interviewer system presents a compelling vision for the future of recruitment. It redefines the candidate evaluation process through AI-powered objectivity, multimodal understanding, and intelligent feedback delivery. By bridging the gap between technological capability and human-centric assessment, this work not only solves current inefficiencies in hiring but also paves the way for a fairer, more inclusive, and more effective global employment landscape.

References

[1] A. Agrawal, R. A. George, S. S. Ravi, S. Kamath and A. Kamar, “Leveraging Multimodal Behavioral Analytics for Automated Job Interview Performance Assessment and Feedback,” *arXiv preprint arXiv:2006.07909*, 2020. [Online]. Available: https://arxiv.org/pdf/2006.07909

[2] A. Jadhav, R. Ghodake, K. Muralidharan, G. T. Varma and V. Bharathi J., “AI-Based Multimodal Emotion and Behavior Analysis of Interviewee,” *International Journal of Scientific Research in Engineering and Management (IJSREM)*, vol. 7, no. 5, May 2023. [Online]. Available: https://www.researchgate.net/publication/370653388

[3] L. Shi, W. Zhou, W. Wang, Q. Wu and W. Zhang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” *arXiv preprint arXiv:2301.00375*, 2023. [Online]. Available: https://arxiv.org/abs/2301.00375

[4] D. A. Levashina and M. A. Campion, “How does bias enter the employment interview? Identifying the riskiest applicant characteristics, interviewer characteristics, and sources of potentially biasing information,” *Human Resource Management Review*, vol. 17, no. 2, pp. 120–142, 2007. [Online]. Available: https://doi.org/10.1016/j.hrmr.2007.05.001

[5] B. C. Lee and B. Y. Kim, “Development of an AI-Based Interview System for Remote Hiring,” *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. 12, no. 3, pp. 537–544, Mar. 2021. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/66683706/IJARET_12_03_060-libre.pdf

[6] Google, “Interview Warmup – Practice answering real interview questions,” *Grow with Google*, 2023. [Online]. Available: https://grow.google/interview-warmup

[7] HireVue, “HireVue AI Hiring Platform – Official Site,” 2023. [Online]. Available: https://www.hirevue.com/

[8] HireVue, “How HireVue Created ‘Glass Box’ Transparency for Its AI Application,” *HireVue White Paper*, 2021. [Online]. Available: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1623&context=misqe