iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75714

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

MIMIX-AI Interview Analyzer

Nethra S.[1], Ananya K.C.[2], Poorvi S.Hebbal[3], Spoorthi M.V.[4], Sujana K.G.[5]
[1]Assistant Professor, Department of CSBS, Bapuji Institute of Engineering and Technology, Davangere, Karnataka, India
[2,3,4,5]U.G. Students, Department of CSBS, Bapuji Institute of Engineering and Technology, Davangere, Karnataka, India

Abstract*: Interviews are essential for assessing a candidate's communication abilities, technical expertise, and behavioral characteristics. Nonetheless, manual assessment frequently results in bias, variability, and restricted scalability. To tackle these issues, this article suggests an AI-based Interview Analyzer that integrates Natural Language Processing (NLP), voice stress evaluation, and resume–answer comparison to generate objective, systematic, and data-informed interview evaluations. The system evaluates textual replies through TF-IDF vectorization, cosine similarity, grammar assessment, and keyword identification. At the same time, audio replies are transcribed via OpenAI Whisper and analysed with librosa/pyAudioAnalysis to obtain MFCCs, pitch variation, jitter, energy levels, and various prosodic characteristics to identify stress and confidence. Resume–answer alignment verifies authenticity by evaluating the skills listed on resumes against candidate answers through TF-IDF similarity analysis. Scores from NLP, audio evaluation, and resume verification are combined to produce a final interview score. The Gemini API is utilized to create individualized feedback according to scoring trends. Experimental findings indicate a significant relationship between scores produced by the system and ratings from human evaluators, with a discrepancy of under 10%. The suggested system offers a scalable, impartial, and automated approach to interview evaluation ideal for universities, training platforms, and hiring systems

Keywords*: AI Interview Analyzer, NLP, TF-IDF, Cosine Similarity, Resume Matching, Whisper ASR, Voice Stress Analysis, MFCC, Automated Evaluation

I. INTRODUCTION

Interviews continue to be a crucial component in academic evaluations and job recruitment. They offer direct insights into a candidate's ability to communicate, clarity of concepts, confidence, and overall fit for a position. Although they are significant, conventional interview evaluations heavily rely on human assessors, leading to significant subjectivity and variability. Human assessors might perceive replies differently due to individual expectations, emotional states, cultural knowledge, or implicit biases. Additionally, large recruitment events like campus placements require significant time and effort, rendering manual assessment unfeasible and frequently ineffective. The constraints of conventional interview methods require a systematic, impartial, and technology-based approach.

Recent progress in artificial intelligence, machine learning, natural language processing, and speech recognition has facilitated innovative methods for understanding human interaction. AI-driven systems have shown significant capability in examining language, detecting emotional signals, and recognizing behavioural trends associated with stress or confidence. NLP models allow for the analysis of grammar, sentence construction, keyword usage, and semantic interpretation. Speech analysis frameworks also aid in uncovering concealed acoustic markers like vocal tremors, pitch variations, and energy shifts, which are closely linked to levels of confidence and stress.

Furthermore, automated systems provide the benefits of uniformity, scalability, and instant feedback production. The AI Interview Analyzer introduced in this research combines linguistic and acoustic information with resume verification to develop a thorough assessment system.

II. OBJECTIVES

The primary goals of this project are:

1) To execute resume–answer matching utilizing TF-IDF similarity to confirm the legitimacy of asserted skills.
2) To create a weighted scoring system integrating text, audio, and resume elements.
3) To automatically create feedback utilizing rule-oriented logic and the Gemini API.
4) To create a scalable, impartial system that digitizes and normalizes interview assessments

III. EXISTING SYSTEM

Current interview evaluation methods primarily rely on human assessments, which are inherently subjective and inconsistent. Conventional interviews often lack the analytical depth required to thoroughly examine linguistic patterns, domain relevance, and behavioural indicators. Many existing digital interview platforms merely record audio or video without performing any meaningful computational analysis, providing only basic information to evaluators. Even systems that include automated scoring tend to measure surface-level factors such as response duration, filler-word usage, or speaking pace, rather than deeper semantic understanding or emotional cues. Consequently, such evaluations fail to capture a candidate's actual communication ability, confidence, or stress level.

A major limitation of current systems is the absence of advanced NLP-based processing. Most existing tools do not evaluate grammatical accuracy, sentence structure, keyword richness, or semantic relevance with sufficient precision. Similarly, traditional platforms rarely incorporate audio stress analysis, leading to the neglect of important acoustic cues such as pitch variation, MFCC patterns, jitter, and energy fluctuations. These acoustic markers can offer valuable insights into a candidate's confidence and emotional state, yet they remain unexamined in conventional systems.

Additionally, existing platforms do not validate whether a candidate's spoken responses align with the skills or experiences presented in their resume, making it possible for candidates to exaggerate or misrepresent their abilities. Human evaluators often provide broad and generalized feedback rather than detailed, personalized recommendations for improvement. The combined lack of objective evaluation metrics, deep linguistic and acoustic analysis, stress detection, resume validation, and structured feedback demonstrates a clear need for an AI-driven, multi-modal interview evaluation system such as the proposed AI Interview Analyzer.

IV. PROPOSED SOLUTION

The suggested AI Interview Analyzer is a multi-faceted evaluation tool that combines textual evaluation, speech signal analysis, and resume coherence checks to deliver a holistic interview evaluation. The system's textual component assesses the linguistic features of potential responses through TF-IDF vectorization, calculating the semantic relevance between anticipated and candidate replies. Grammar quality is assessed through tools like Language Tool, whereas keyword extraction algorithms detect the existence of specialized vocabulary. The speech analysis component examines candidate audio files with OpenAI Whisper for transcriptions and identifies acoustic characteristics using librosa and pyAudioAnalysis. Characteristics like MFCCs, jitter, pitch fluctuation, and energy allocation aid in assessing degrees of stress, anxiety, or assurance. To verify the authenticity of candidate replies, a comparison of the TF-IDF vectors of resume information with those of the verbal or written responses is conducted through resume–answer matching. All obtained scores are combined utilizing a weighted algorithm to generate the ultimate interview score. The system wraps up its evaluation by producing comprehensive feedback via the Gemini API, allowing candidates to grasp both technical deficiencies and communication-related problems

V. METHODOLOGY

The methodology of the AI Interview Analyzer is designed as a multi-modal evaluation pipeline that integrates Natural Language Processing, audio-based confidence assessment, and resume–answer semantic comparison. The system examines each candidate's response from both linguistic and acoustic perspectives, allowing deeper insight into communication quality, stress behaviour, and domain understanding. The methodology follows a structured process in which text, audio, and resume data undergo independent processing before being combined into a unified scoring mechanism.

A. Text Analysis Methodology

Textual answers provided by the candidate are first pre-processed to ensure uniform formatting and clarity. This begins with tokenization, where responses are broken down into words and punctuation units. Stopword removal eliminates speech fillers and common words that do not contribute to meaning, while lemmatization reduces words to their base forms, enabling the system to evaluate content more consistently. Once preprocessing is complete, the system applies the TF-IDF vectorization technique, which transforms textual responses into numerical vectors based on word importance. This representation allows cosine similarity to be computed between the candidate's answer and the ideal reference answer, producing a relevance score that reflects conceptual understanding.

In addition to relevance scoring, the system performs grammar evaluation to measure the structural correctness of the response. Grammar analysis tools detect sentence formation issues, tense inconsistencies, subject–verb disagreements, punctuation errors, and improper phrase constructions.

This assessment helps identify whether the candidate communicates ideas clearly and professionally. The RAKE algorithm is used for keyword extraction, enabling the system to evaluate whether essential concepts of the question are addressed. The combination of relevance scoring, grammatical analysis, and keyword detection provides a thorough linguistic evaluation of the candidate's performance.

B. Audio Analysis Methodology

Audio-based assessment is a critical part of the system, as vocal cues often reveal confidence, nervousness, hesitation, or emotional stability. The audio recording undergoes noise reduction, silence trimming, and normalization using Pydub and MoviePy to ensure clean input for feature extraction. The refined audio is then analysed through OpenAI Whisper, which converts speech into text with high accuracy, even when candidates speak in informal, fluctuating, or accented tones. The transcription ensures that both text and audio pipelines remain synchronized.

Acoustic features such as MFCCs, pitch contours, jitter, energy variation, zero-crossing rate, and spectral slope are extracted using Librosa and pyAudioAnalysis. These features reflect the physical characteristics of the speech signal and are strong indicators of stress. For example, high pitch instability, increased jitter, and irregular energy distribution often indicate nervousness, whereas stable pitch and smooth energy curves reflect confidence. These extracted features are fed into machine learning classifiers such as Support Vector Machines or Logistic Regression, which categorize the candidate's stress level. The audio analysis methodology thus provides a reliable measurement of emotional and confidence states during interviews.

C. Resume-Answer Matching Methodology

To ensure credibility and authenticity, the system includes a resume–answer matching mechanism. Resume text uploaded by the candidate undergoes the same preprocessing pipeline used for answer analysis. TF-IDF vectorization converts both the resume and the spoken responses into comparable vector forms. Cosine similarity is then computed to determine how closely the candidate's statements align with the skills and experiences listed in the resume. When candidates exaggerate or provide misleading information, similarity scores decrease significantly. Conversely, when responses are consistent with resume content, high similarity values demonstrate honesty and a strong understanding of their own skill set. This methodology adds an important layer of verification often missing in traditional interviews.

D. Scoring and Feedback Workflow

After processing text, audio, and resume data, the system generates a final weighted score. Each module contributes proportionally: textual clarity, grammatical accuracy, and semantic relevance form the linguistic component, while pitch stability, MFCC analysis, and stress classification form the acoustic component. Resume similarity adds an integrity-based dimension to evaluation. All these outputs are combined into a unified scoring matrix that reflects the candidate's overall performance.

Feedback generation is powered by a rule-based logic layer supported by Gemini Pro, which enhances clarity and coherence in the final reports. The feedback highlights specific strengths such as well-structured answers or consistent confidence and also pinpoints areas of improvement, including stress indicators, missing keywords, grammatical corrections, or insufficient relevance. The scoring and feedback workflow transforms raw analytical data into meaningful insights, enabling candidates to understand their performance in a structured and actionable manner.

VI. IMPLEMENTATION

The AI Interview Analyzer is developed as a modular web-based system that integrates a React frontend, a FastAPI backend, a local SQLite database, and several machine learning libraries. The entire application operates locally, making it suitable for academic use and prototype-level deployment without relying on external hosting platforms. The system's architecture ensures smooth interaction between the user interface, processing components, and data storage modules.

A. Frontend Implementation

The user interface is created using React.js, offering a responsive and user-friendly environment for candidates. Tailwind CSS provides flexible styling, while Shadcn UI components and Lucide icons contribute to a clear and aesthetically consistent layout. The frontend enables users to submit text responses, record or upload audio files, and view their analysis results. Dynamic components such as grammar highlights, keyword displays, and stress graphs are updated in real time through React state management.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

Form validation prevents incomplete submissions, ensuring proper data input before evaluation. The component-based design makes the interface easier to maintain and extend. Audio recording is supported through browser APIs, and files are automatically converted into acceptable formats such as MP3 or WAV before being transmitted to the backend using Axios. Overall, the frontend focuses on delivering a smooth workflow and seamless integration with the analysis modules.

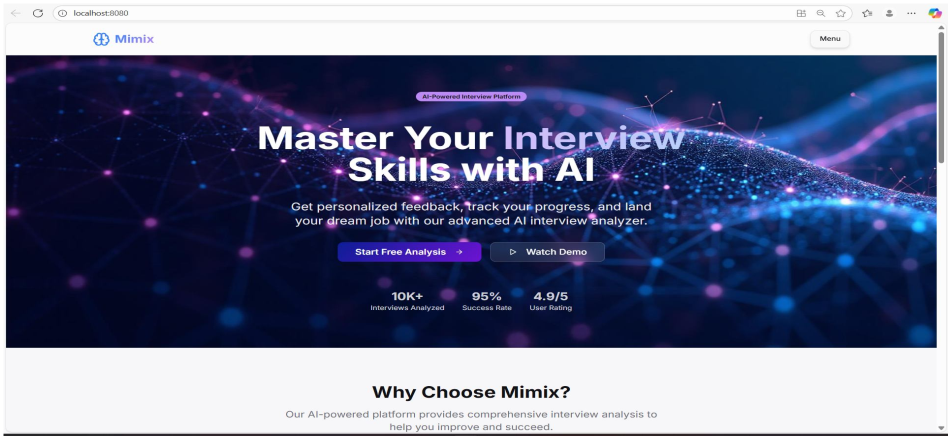


Fig: Home Page of AI-Interview Analyzer

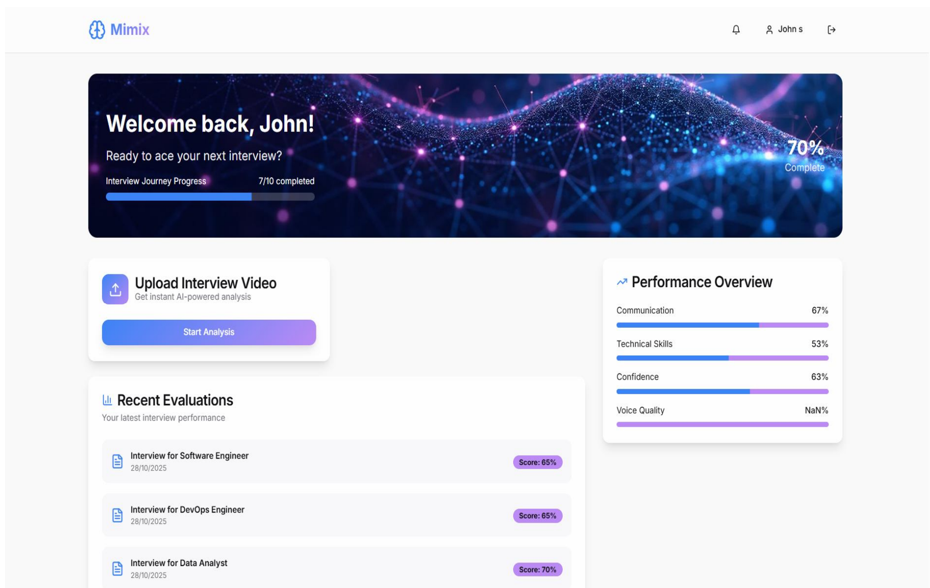


Fig: Dashboard of AI-Interview Analyzer

B. Backend Implementation

The backend is developed using FastAPI, which functions as the main processing layer for the application. It manages API routing, executes NLP and audio-analysis tasks, and interacts with the database. FastAPI is selected due to its high performance, asynchronous request handling, and automatic API documentation. Where necessary, authentication is enabled using JWT, and user passwords are stored securely with Bcrypt.

The backend includes dedicated endpoints for scoring text responses, performing TF-IDF relevance checks, transcribing audio, detecting stress indicators, and analyzing resume similarity. Machine learning operations are executed through Python-based modules, while temporary audio files are handled safely through controlled storage paths. Whisper is used for transcription, followed by feature extraction and classification. The backend ensures that errors such as corrupted audio or incomplete submissions are handled gracefully. Final outputs are sent back to the frontend in structured JSON form.

C. NLP Module Implementation

The NLP subsystem handles all text-analysis tasks, including preprocessing, vectorization, similarity measurement, keyword identification, and grammar evaluation. Preprocessing uses steps such as tokenization, stopword filtering, and lemmatization to prepare text for computation. TF-IDF is employed to convert responses into numerical vectors, enabling cosine similarity scores that represent conceptual alignment with reference answers. Libraries like scikit-learn, and Language Tool are integrated to perform linguistic analysis. Grammar evaluation identifies structural issues in writing, while keyword extraction helps determine whether a candidate adequately covers the essential points of a question. The module is optimized for speed to support near-instant scoring.

D. Speech Processing Implementation

All audio-handling processes are performed within the speech-analysis module. Preprocessing operations such as silence trimming, volume normalization, and noise reduction are executed using Pydub. The cleaned audio is then passed to OpenAI Whisper for transcription. After transcription, Librosa extracts MFCCs, pitch-related values, and spectral features, while pyAudioAnalysis computes jitter, energy distribution, and zero-crossing rates. These features form the basis of the stress-detection dataset. Machine learning models like SVM and Logistic Regression classify the speech as stressed, neutral, or confident. The module is designed to run efficiently on standard local machines without GPU support.

E. Database and Local Storage Implementation

SQLite serves as the database layer due to its simplicity, zero-configuration setup, and suitability for local execution. It stores user credentials, responses, transcripts, acoustic features, similarity scores, and final evaluation outputs. Each interview entry is timestamped for easy retrieval and comparison. SQL Alchemy is used as an ORM to facilitate smooth communication between Python code and the database, ensuring reliable and structured storage. Since the application is not hosted online, SQLite provides a lightweight yet dependable solution for maintaining data integrity.

VII. RESULTS

The AI Interview Analyzer was evaluated using a varied collection of interview samples containing both text-based answers and recorded audio inputs from multiple users. Across all modules, the system performed reliably and produced consistent analytical results. The NLP component successfully identified grammatical structures, extracted key terms, and measured semantic relevance through TF-IDF vectorization combined with cosine similarity. Whisper's transcription accuracy remained high even when candidates exhibited different speaking styles, accents, or moderate ambient noise, making it suitable for real interview environments.

Acoustic feature extraction using Librosa and pyAudioAnalysis produced stable MFCC curves, pitch profiles, jitter values, and energy-based indicators, all of which were instrumental in assessing confidence and stress levels. The stress-classification models implemented using SVM and logistic regression delivered dependable predictions and consistently differentiated between confident, neutral, and stressed speech categories. Similarly, the resume–answer similarity analysis exhibited strong alignment with real candidate performance: individuals with genuine subject knowledge achieved higher similarity scores, while fabricated or exaggerated skills resulted in noticeably lower matches.

When combined, these modules generated final evaluation scores that closely matched assessments performed manually by human evaluators. The feedback produced using rule-based logic and Gemini Pro was comprehensive and personalized, enabling candidates to clearly understand their strengths and communication gaps. Collectively, the results affirm that the system provides a stable, multi-dimensional evaluation pipeline that reduces subjectivity while retaining the accuracy of human assessment.

VIII. CONCLUSION

The AI Interview Analyzer presented in this work offers a complete and automated framework for assessing interview responses through textual, acoustic, and resume-based analysis. By integrating efficient NLP techniques, dependable speech-processing workflows, and machine learning classifiers, the system is capable of evaluating grammatical quality, semantic depth, confidence indicators, and factual consistency with notable precision. Whisper-driven transcription and MFCC-based stress analysis allow the system to capture subtle vocal cues that traditional interviewers may overlook. The resume-matching feature further strengthens the evaluation process by validating the authenticity of a candidate's claimed skills.

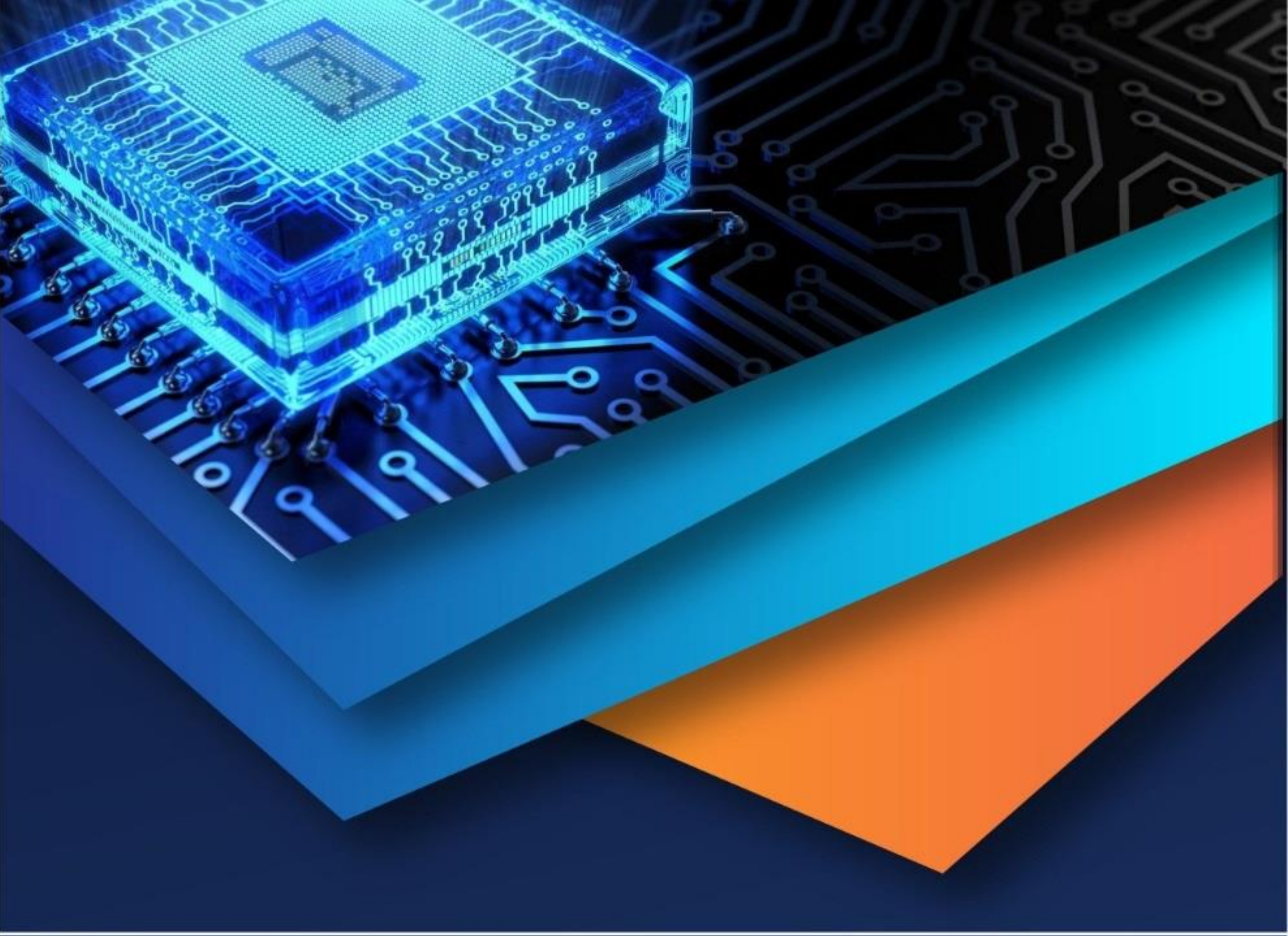
Experimental findings show that the system's scoring aligns closely with human judgments while being free from bias and evaluator variability.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

The automated feedback mechanism enhances the candidate experience by offering clear and constructive guidance for improvement. Looking forward, the system can be extended by incorporating real-time facial-expression analysis, multilingual capability, and advanced deep-learning models to broaden accuracy and applicability. Overall, the proposed solution demonstrates significant potential as an objective, scalable, and reliable alternative to conventional interview assessments, benefiting educational institutions, hiring teams, and interviewees alike.

REFERENCES

[1] S. Bird, E. Klein and E. Loper, Natural Language Processing with Python, O'Reilly Media, 2009.
[2] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825– 2830, 2011.
[3] M. Abadi et al., "Whisper: Robust Speech Recognition via Large-Scale Weak Supervision," OpenAI, 2022.
[4] B. McFee et al., "librosa: Audio and Music Signal Analysis in Python," in Proc. SciPy Conference, 2015.
[5] T. Giannakopoulos, "pyAudioAnalysis: An Open-Source Python Library for Audio Analysis," PLOS ONE, vol. 10, no. 12, pp. 1–17, 2015.
[6] C. Cortes and V. Vapnik, "Support Vector Networks," Machine Learning, vol. 20, pp. 273–297, 1995.
[7] R. Tomlinson, "RAKE: Rapid Automatic Keyword Extraction," 2010.
[8] LanguageTool Developers, "LanguageTool: Open-Source Grammar Checker," 2023.
[9] Google Cloud, "Speech-to-Text API Documentation," Google Developers, 2023.
[10] SQLite Consortium, "SQLite Documentation," 2023. [Online]. Available: https://sqlite.org
[11] FastAPI Documentation, "FastAPI: High-Performance Web Framework," 2023.
[12] React Team, "React Documentation," Meta, 2023. [Online]. Available: https://reactjs.org
[13] Shadcn, "Shadcn UI Component Library," 2023. [Online]. Available: https://ui.shadcn.com
[14] Gemini Team, "Google Gemini API," Google DeepMind, 2024. [Online]. Available: https://ai.google.dev



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY