ARTIFICIAL INTELLIGENCE

AI Mock Interview System: A Web-Based Framework for Structured Interview Practice and Explainable Answer Assessment

Mohammed Nazeer Ahmed, N. Akshith, N. B. Yashwanth, K. Barath, Dharmendra Roy

Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management (HITAM), Hyderabad, Telangana, India

Published November 2026

Abstract

AI Mock Interview System is a web-based platform designed to help students and job seekers practise technical and human-resources interviews in an organized way. Interview preparation needs repeated practice, relevant questions, and feedback that explains how an answer can be improved, but candidates do not always have a mentor who can provide this. The current application provides role selection, interview-category configuration, question presentation, session management, and a results interface using frontend web technologies, Java Spring Boot REST services, and MySQL for relevant application data. A clear distinction is made between this existing workflow and an AI assessment capability, which has not yet been verified in the implementation. To go beyond interface integration, the paper proposes a rubric-guided answer-assessment module that evaluates technical correctness, relevance, completeness, and clarity, and returns feedback linked to evidence in the response. The proposed evaluation compares automated scores with independent human ratings and with baseline approaches using error, agreement, and robustness measures. No experimental accuracy or effectiveness result is claimed, because the assessment component, dataset, and tests remain to be completed. Overall, the paper documents the problem, system architecture, implementation considerations, evaluation plan, limitations, and future work.

Keywords: AI-assisted interview practice; automated answer assessment; explainable AI; semantic analysis; rubric-based scoring; personalized feedback

1 Introduction

Interview performance depends on more than recalling technical facts. Candidates must interpret a question, select relevant knowledge, explain their reasoning, and communicate clearly under time constraints. Students preparing for placements may not always have access to a mentor who can conduct repeated mock interviews and give consistent feedback. A digital practice environment can make preparation more accessible by organizing sessions around job roles and interview categories.

AI Mock Interview System is a web-based application whose demonstrated workflow includes job-role selection, interview configuration, question presentation, session completion, and a results interface. The backend is described as Java Spring Boot with REST APIs, while MySQL supports relevant application data. These capabilities form a structured practice platform, but a result screen alone does not establish that candidate answers are evaluated by an AI model.

Research on automatic short-answer scoring shows that semantic understanding and evaluation design matter when scoring open-ended responses [1], [2]. Work on automated interview scoring

also stresses validating score meaning, reliability, and alignment with human judgments rather than assuming automated scores are valid [3], [4].

This paper therefore has two aims: to document the current workflow accurately, and to define an answer-assessment extension that can be implemented and evaluated. The intended contributions are (1) a modular design for integrating a rubric-based assessment component, (2) a scoring formulation that separates criterion-level judgments from the final score, and (3) an evaluation protocol comparing automated results with human ratings and simpler baselines. The novelty of the approach remains provisional until implementation and a systematic comparison with related work are completed.

The scope is limited to practice and formative feedback. The system is not meant to rank applicants for employers, infer personality from appearance, or make employment decisions, since high-stakes assessment needs evidence of reliability, validity, fairness, and generalizability [3], [4].

2 Problem statement

Students and job seekers need useful and understandable feedback on their interview answers, but a mock interview tool that only shows a score gives little help if the score is unexplained. The central problem is how to produce useful feedback without treating an unexplained score as proof of competence. Existing practice tools may present questions and results, yet it is often unclear how an answer was judged or what should be improved.

The platform is intended to let users:

- Choose a job role for interview practice
- Select an interview category such as Technical, HR, or Technical + HR where supported
- Enter a question-based interview session
- Submit answers and complete the session
- View the result information available in the interface
- Receive criterion-level feedback (proposed assessment module)

The exact question-generation mechanism, answer-capture method, score calculation, and persistence of interview history must be verified from the source code before publication. The following research questions guide future evaluation; they are planned questions, not results already established:

- **RQ1:** How closely do automated rubric scores agree with independent human ratings?
- **RQ2:** Does evidence-linked, criterion-level feedback help reviewers understand why a score was assigned?
- **RQ3:** How does the proposed method compare with keyword-coverage and semantic-similarity baselines?
- **RQ4:** How stable are scores when answers are paraphrased or give a valid alternative explanation?

3 Objectives

The main objectives of AI Mock Interview System are to:

- Provide one web-based platform for structured interview practice.
- Organize sessions around job roles and interview categories.
- Present interview questions and manage interview sessions.
- Display session results through a results interface.
- Propose a rubric-guided assessment of correctness, relevance, completeness, and clarity.
- Give feedback linked to evidence in the candidate's answer.
- Compare automated scores with independent human ratings and baseline methods.
- Keep the system a practice aid and not a hiring decision tool.
- Provide a base for future validated scoring and analytics.

4 Research gap

Related research already addresses semantic grading and automated interview assessment, so AI-based scoring or explainability cannot be claimed as new in itself. Many practice platforms present questions and a result, but the way an answer is judged is often not transparent, and a high-stakes style of scoring needs evidence of reliability and validity.

The defensible contribution is not the use of an AI API alone. It is the definition and evaluation of a rubric-guided, evidence-linked feedback workflow inside an existing interview-practice application. The proposed system brings together:

- Role- and category-based interview sessions.
- Question-specific rubrics.
- Criterion-level scoring with explanations.
- Review routing for low-confidence cases.
- Comparison with human raters and baselines.

This contribution depends on the specific implementation, dataset, evaluation results, and comparison with prior methods.

5 Literature review

5.1 Automatic short-answer assessment

Automatic short-answer grading (ASAG) has explored lexical overlap, semantic similarity, feature-based models, and neural language representations. Burrows et al. reviewed the development of ASAG systems and the diversity of methods and evaluation settings [1]. Mohler and Mihalcea showed the importance of comparing meaning rather than relying on exact word overlap [2]. More recent work studies neural and large-language-model-based scoring, but performance still depends on question type, reference quality, data distribution, and evaluation protocol.

5.2 Automated interview scoring

Interview responses may be longer, less structured, and meant to measure several competencies. Liff et al. examined the reliability and validity of automated video-interview competency assessments [3].

Hickman et al. studied how training sample size, reference-rating reliability, and NLP choices affect language-based interview scores [4]. These findings motivate transparent scoring criteria and empirical validation before scores are treated as meaningful feedback.

5.3 Rubric-guided and framework-based assessment

GradeAid shows how short-answer assessment can be designed, implemented, and evaluated as a complete pipeline rather than a single similarity calculation [6]. Rubric-guided research aligns answer content with rubric criteria, which supports criterion-level evidence [7]. The present work adopts these principles as design guidance and does not claim to reproduce or outperform those methods.

6 Proposed system flow

The application follows a client–server architecture. The frontend presents role selection, interview setup, questions, and results. REST requests connect the interface to Spring Boot, which coordinates operations and database access, and MySQL stores relevant records where persistence is implemented. The proposed assessment service is invoked after answer submission and before feedback is displayed, as a separate service boundary so it can be tested independently or disabled without breaking basic practice.

6.1 Main system flow

Role Selection → Interview Configuration → Question Presentation → Answer Submission → Rubric Assessment (proposed) → Feedback and Results

The proposed module receives a question, a candidate response, and a question-specific rubric. Its stages are:

- **Validation:** check that question, response, and rubric are present and compatible.
- **Preprocessing:** normalize whitespace and formatting but keep code identifiers, notation, negation, and technical terms.
- **Mapping:** link response statements to required concepts and rubric criteria.
- **Criterion scoring:** estimate performance per criterion and attach evidence or a brief explanation.
- **Aggregation:** combine criterion scores, generate actionable feedback, and record the evaluator version.

Scoring formulation. Let $s_j \in [0, 1]$ be the score for criterion j and w_j its predeclared weight, with $\Sigma w_j = 1$. The proposed aggregate score is $S = 100 \times \Sigma(w_j s_j)$. Rubric and weights should be fixed before evaluation and reviewed by subject-matter experts. This is a proposed design, not a verified description of the current result calculation.

Explainability and uncertainty. Semantic similarity shows whether a response addresses the question but cannot establish factual correctness, so it should be combined with a rubric or curated concept list. The evaluator should distinguish a missing concept from an incorrect claim, and route low-confidence, conflicting, or out-of-scope cases to human review instead of giving a falsely precise score. For example, for a question on database indexing, a response that mentions faster lookup but omits the storage or update trade-off should receive partial credit, with feedback naming the missing criterion.

7 Major modules

- **Role and Category Selection** – lets the user choose a job role and interview type.
- **Session Management** – creates, tracks, and completes interview sessions.
- **Question Presentation** – shows the configured questions to the candidate.
- **Answer Submission** – captures candidate responses for the session.
- **Results Interface** – displays the available result information.
- **Backend REST Services** – Spring Boot controllers that validate requests and coordinate operations.
- **Data Management** – MySQL database (ai_mock_interview in the development setup) for application data.
- **Rubric Assessment Module (proposed)** – scores answers by criterion with evidence-linked feedback.
- **Human Review (proposed)** – handles low-confidence or out-of-scope cases.

8 Methodology

1. Problem identification and literature study
2. Requirement analysis (functional and non-functional)
3. System design and architecture
4. Frontend, backend, and database development
5. Rubric design and weighting (proposed)
6. Assessment module implementation (proposed)
7. Functional testing and debugging
8. Evaluation against human raters and baselines (planned)

Non-functional requirements. The application should validate incoming data, return controlled errors, keep the database consistent, and separate frontend interaction from backend processing, with authentication and authorization for protected functions. A numerical score should not be shown when the response is missing, the question–rubric mapping is unavailable, or the evaluator cannot give a supported judgment; such cases should be marked for review.

Data model. A minimal model for the evaluator would associate each question with a stable identifier, expected concepts, a rubric version, and criterion weights. Each response references the interview session and question, and assessment records keep the evaluator version, criterion-level outputs, and review status. These are design recommendations to be reconciled with the existing schema.

9 Security and privacy considerations

Security and Privacy Considerations Interview responses and account information may contain personal data. Security measures to consider include:

- Validation of incoming requests.
- Authentication and authorization for restricted operations.
- Limiting the information that is stored.
- Access controls and retention periods for stored answers.

- Avoiding unnecessary identifying information in external model calls.
- Disclosing whether response text is processed by a third-party AI provider.
- Recording model version and handling timeouts if an external AI model is used.

If an LLM is used, it may produce unsupported explanations, overvalue fluent but incorrect answers, or score valid alternatives inconsistently. Rubrics, evidence-linked explanations, human review, and periodic audits can reduce but not eliminate these risks.

10 Expected outcomes

The system is expected to support the following outcomes:

- A single platform for structured interview practice.
- Role- and category-based interview sessions.
- Clear session states and predictable navigation.
- Readable, criterion-level feedback once the assessment module is added.
- Scores whose meaning can be checked against human ratings.
- Separation of software correctness from assessment quality.

11 Results

The current application demonstrates the basic workflow of interview practice: selecting a role and category, entering a question-based session, completing it, and viewing a results interface. Successful page navigation, however, does not show that scores are accurate or educationally useful.

Functional testing. Representative functional checks were specified for each module. Each scenario must be run on the deployed build, and its actual outcome, build identifier, environment, and defect notes recorded.

API tests should also cover malformed requests, empty answers, invalid session identifiers, database failures, and repeated submissions.

Proposed AI evaluation. Once the assessment module exists, a dataset of questions and answers across topics and difficulty levels should be prepared, including correct, partially correct, incomplete, incorrect, and paraphrased answers. At least two qualified raters should score each response with the same rubric, with disagreements resolved by a documented procedure. A data split should be fixed before tuning, with no near-duplicate leakage between development and test sets. The proposed method is compared with a keyword or concept-coverage baseline and a semantic-similarity baseline, using MAE, RMSE, correlation with human ratings, and an agreement statistic such as intraclass correlation or weighted kappa, with confidence intervals and error breakdowns where sample size permits. Robustness checks cover paraphrases, spelling variation, incomplete answers, and valid alternative explanations. Small pilot results should be called preliminary.

No accuracy, agreement, user-satisfaction, or placement-improvement result is reported, because the evaluator and experiments have not been verified.

User interface and output screens. Figures 1–5 show the main screens and the architecture. They are illustrative interface mockups prepared for this paper, not captured output of a measured evaluation. Any scores shown are sample values.

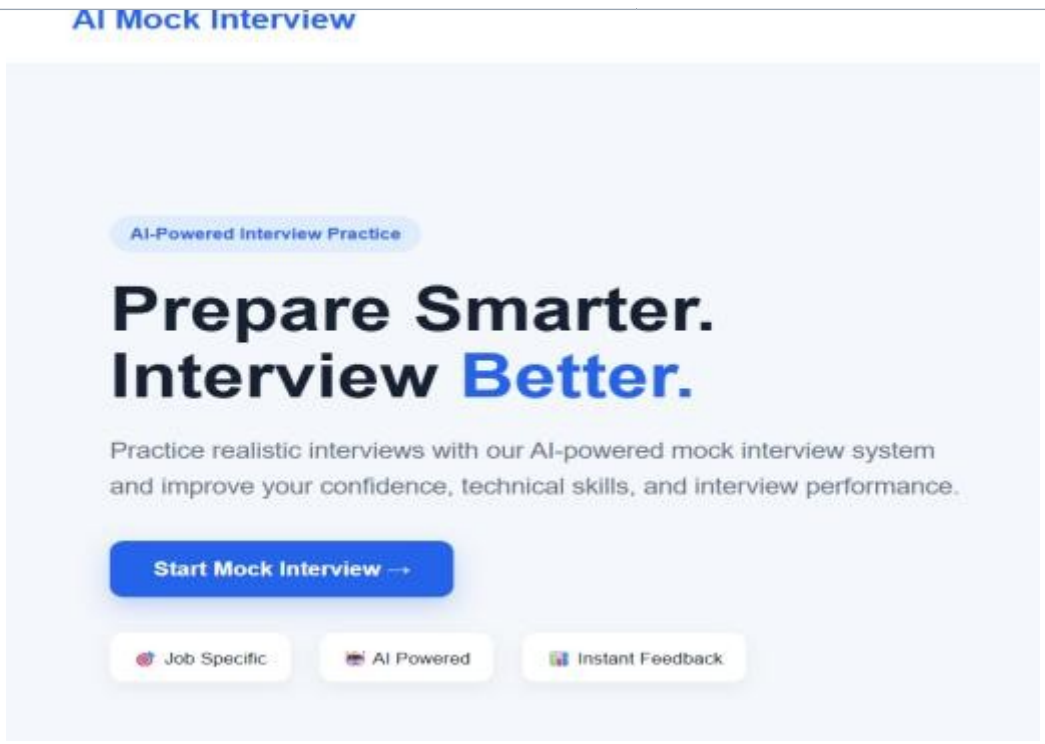


Figure 1. Home page of the AI Mock Interview System.



Figure 2. Interview results screen.

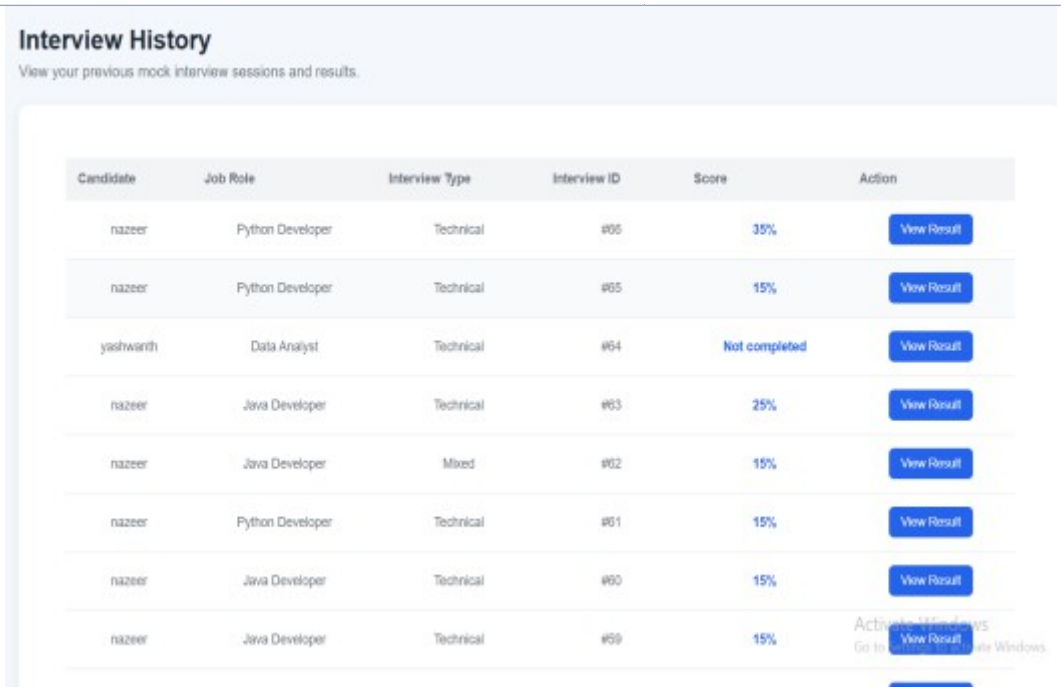


Figure 3. Interview history screen.

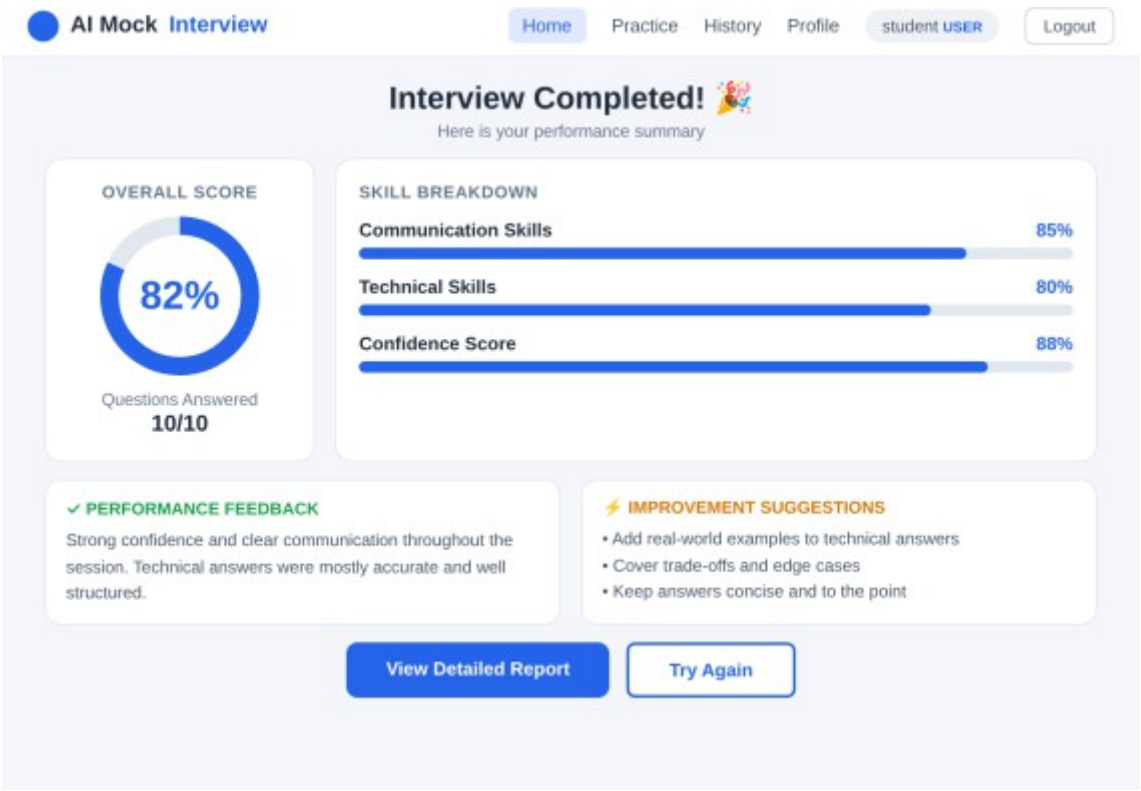


Figure 4. Session results screen (illustrative layout, sample data).

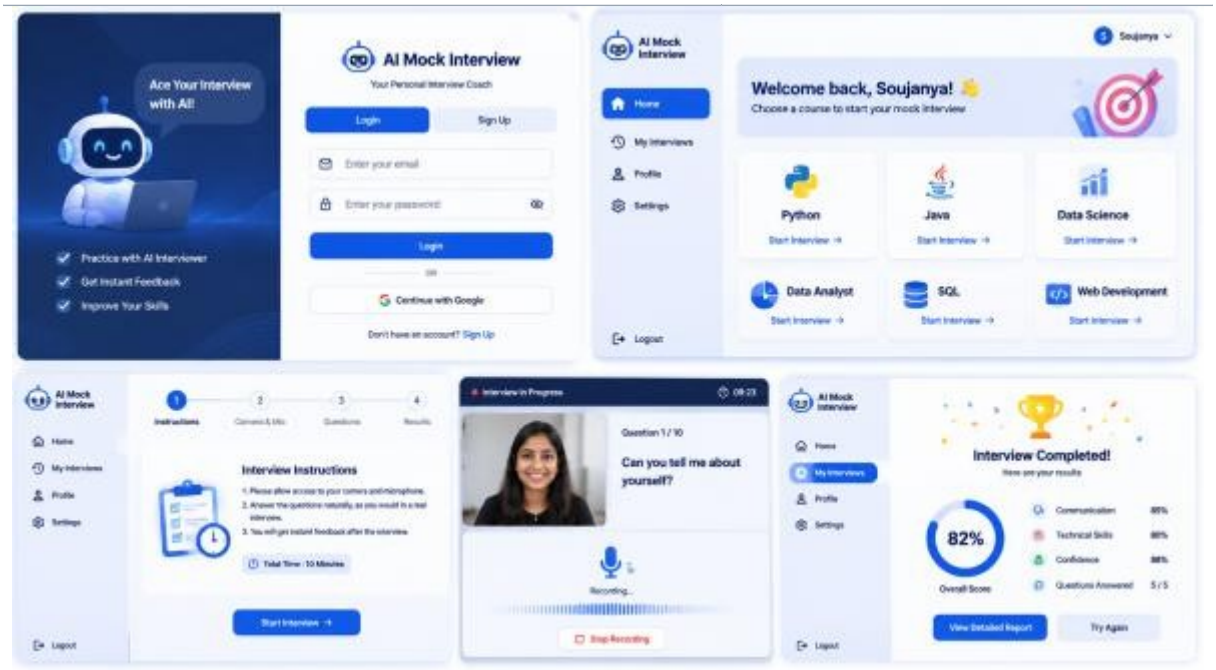


Figure 5. Overview of the main interface screens (illustrative mockups).

12 Discussion

The platform integrates interview configuration, question presentation, backend operations, and a result interface in a single workflow. This is a practical base for structured practice, but the research contribution will be stronger once the assessment process is implemented and evaluated.

The principal limitation is that the exact answer-capture and score-calculation logic remains unverified, and the AI evaluator is a proposed extension and not a confirmed feature. Other limitations include rubric ambiguity, sensitivity to wording, incomplete coverage of valid alternative answers, and fluent but incorrect answers receiving too much credit. The system should not infer a candidate's intelligence, personality, accent quality, or employability from text-only answers; speech or video features would need separate consent, privacy controls, bias analysis, and validation.

A user study could evaluate perceived usefulness, feedback clarity, and ease of use. Claims about improved confidence, interview performance, or placement rates need a suitable study design and cannot be inferred from functional testing. Scalability tests should measure API latency, AI-service response time, failure recovery, and database behavior under load.

13 Future scope

Possible extensions for later versions include:

- Implementation and validation of the rubric engine.
- A small controlled pilot with consented, anonymized responses.
- Role-aware question generation.
- Secure interview history.
- Speech-to-text for spoken answers, if required.
- Personalized practice suggestions.
- Performance analytics based on validated scores.

- Question-bank versioning and adaptive difficulty.
- Multilingual support.
- A human-review dashboard.

These enhancements should be introduced only after core scoring reliability is established and their added benefit can be measured against a clear baseline.

14 Conclusion

AI Mock Interview System is a web-based platform for structured interview practice, built with a frontend, Spring Boot REST services, and MySQL. This paper also proposes an explainable answer-assessment extension that would assess responses against explicit criteria and give evidence-linked feedback.

Because the assessment implementation and experiments have not yet been verified, this paper does not claim measured AI accuracy or improved interview outcomes. Future work must implement the evaluator, validate it against human ratings and baseline methods, complete a systematic literature review, and report reproducible results.

Acknowledgment

The authors acknowledge the academic guidance and support received during the development of the system.

References

[1] Burrows S, Gurevych I, Stein B. The eras and trends of automatic short answer grading. *Int J Artif Intell Educ*. 2015;25:60–117. https://doi.org/10.1007/s40593-014-0026-8

[2] Mohler M, Mihalcea R. Text-to-text semantic similarity for automatic short answer grading. In: *Proc*. 12th Conf. European Chapter of the ACL (EACL); 2009. p. 567–575. https://doi.org/10.3115/1609067.1609130

[3] Liff J, Mondragon N, Gardner C, Hartwell CJ, Bradshaw A. Psychometric properties of automated video interview competency assessments. *J Appl Psychol*. 2024;109(6):921–948. https://doi.org/10.1037/apl0001173

[4] Hickman L, Liff J, Rottman C, Calderwood C, et al. The effects of the training sample size, ground truth reliability, and NLP method on language-based automatic interview scores' psychometric properties. *Organ Res Methods*. 2024. https://doi.org/10.1177/10944281241264027

[5] Szandała T. ChatGPT vs human expertise in the context of IT recruitment. *Expert Syst Appl*. 2025;264:125868. https://doi.org/10.1016/j.eswa.2024.125868

[6] del Gobbo E, Guarino A, Cafarelli B, Grilli L. GradeAid: a framework for automatic short answers grading in educational contexts—design, implementation and evaluation. *Knowl Inf Syst*. 2023;65:4295–4334. https://doi.org/10.1007/s10115-023-01892-9

[7] Gombert S, et al. Are rubrics all you need? Towards rubric-based automatic short answer scoring via guided rubric-answer alignment. In: *Proc*. 16th Int. Learning Analytics and Knowledge Conf. (LAK '26); 2026. p. 272–282. https://doi.org/10.1145/3785022.3785064

[8] Hickman L, Tay L, Woo SE. Are automated video interviews smart enough? Behavioral modes, reliability, validity, and bias of machine learning cognitive ability assessments. *J Appl Psychol*. 2025;110(3):314–335. https://doi.org/10.1037/apl0001236