

AI-POWERED AUTOMATED INTERVIEW SYSTEM FOR RECRUITMENT PROCESS

Yuvraj Bargaje

Department of Computer Engineering, (SPPU)
G. H. Raison College of Engineering & Management
Pune, India

Dr. Pushpi Rani

Department of Computer Engineering, (SPPU)
G. H. Raison College of Engineering & Management
Pune, India

Jaydeep Shinde

Department of Computer Engineering, (SPPU)
G. H. Raison College of Engineering & Management
Pune, India

Manali Sapkal

Department of Computer Engineering, (SPPU)
G. H. Raison College of Engineering & Management
Pune, India

Abstract— *The recruitment process in many organizations requires significant time and human effort. Traditional interview methods involve manual resume screening, written tests, and face-to-face interviews, which may sometimes lead to delays and inconsistent evaluation. To address these challenges, this paper presents an AI-Powered Automated Interview System that assists in evaluating candidates efficiently using artificial intelligence techniques. The proposed system automates multiple stages of the hiring process, including resume analysis, MCQ assessment, verbal interview evaluation, coding assessment, and report generation. The system analyzes resumes using artificial intelligence to extract important information such as skills, projects, and experience. Candidates are then evaluated through multiple choice questions and interview responses, which are assessed using semantic similarity techniques. Computer vision techniques are also used to monitor candidates during the interview and detect suspicious activities such as the presence of multiple faces. In addition, the system provides personalized learning recommendations and relevant educational resources to help candidates improve their skills. Finally, detailed performance reports are generated for both candidates and administrators. The system is implemented using Flask as the backend framework, SQLite for database management, Google Gemini API for AI analysis, Sentence Transformers for answer evaluation, and OpenCV for face detection. The proposed solution helps improve the efficiency, fairness, and automation of the recruitment process.*

Keywords— *Artificial Intelligence, Automated Interview System, Resume Analysis, Recruitment Automation, Natural Language Processing, Computer Vision, Candidate Evaluation, Machine Learning, Coding Assessment, Online Recruitment System..*

I. INTRODUCTION

Recruitment is an important process for every organization because companies need skilled and capable employees to achieve their goals. Traditionally, the recruitment process includes several steps such as resume screening, written tests, and personal interviews conducted by human recruiters. These steps require significant time and effort, especially when a large number of candidates apply for a job. Manual evaluation may also lead to inconsistency and bias in decision-making. Therefore, many organizations are now exploring the use of

Artificial Intelligence (AI) and automation technologies to improve the efficiency and fairness of the hiring process [1].

In recent years, artificial intelligence has been widely used in different domains such as healthcare, finance, education, and human resource management. AI technologies can analyze large amounts of data, identify patterns, and provide intelligent recommendations. In the field of recruitment, AI can help organizations automatically analyze resumes, evaluate candidate responses, and assist recruiters in making better hiring decisions. Automated recruitment systems reduce manual workload and allow organizations to conduct faster and more structured evaluations of candidates [2].

Online interviews and digital assessments have also become more common due to the growth of remote work and online hiring platforms. However, traditional online interview systems still require human interviewers to evaluate candidate answers, which may lead to subjective judgement. Natural Language Processing (NLP) and machine learning techniques can be used to analyze candidate responses and measure semantic similarity between answers and expected responses. These techniques help in evaluating answers more objectively and consistently [3].

Another challenge in online interview systems is maintaining fairness and preventing cheating. Candidates may try to take help from others during online interviews. Computer vision technologies such as face detection can be used to monitor candidates through webcams and detect suspicious situations like the presence of multiple faces or absence of the candidate from the screen. Such technologies help maintain the integrity of the interview process and ensure that the evaluation is fair [4].

Considering these challenges, this project proposes an AI-Powered Automated Interview System that integrates multiple technologies to automate the recruitment process. The system performs resume analysis using artificial intelligence, conducts multiple-choice question (MCQ) tests, evaluates candidate answers in a verbal interview using semantic

similarity techniques, and monitors candidates using face detection. It also includes a coding assessment round where candidate solutions are evaluated using predefined test cases. After the interview process, the system generates performance reports and provides learning recommendations to help candidates improve their skills [5].

The proposed system aims to reduce the workload on recruiters while ensuring consistent and objective evaluation of candidates. By combining AI, natural language processing, and computer vision technologies, the system provides an efficient platform for automated recruitment and candidate assessment. This approach can help organizations streamline their hiring process and identify suitable candidates more effectively.

II. LITERATURE REVIEW

Many researchers have studied how Artificial Intelligence can be used to improve the recruitment process. In traditional hiring systems, human recruiters manually review resumes, conduct interviews, and evaluate candidates based on their knowledge and experience. However, this manual process is time-consuming and sometimes leads to inconsistent evaluation. Recent studies show that AI-based recruitment systems can automatically analyze candidate information and assist organizations in selecting suitable candidates more efficiently. These systems use machine learning and natural language processing techniques to analyze resumes and match candidate skills with job requirements [6].

Resume screening is one of the first and most important stages of recruitment. Several research works have proposed automated resume parsing systems that extract useful information such as skills, education, projects, and work experience from resumes. Natural Language Processing techniques help in converting unstructured resume text into structured information that can be easily analyzed by the system. Such systems help recruiters quickly filter large numbers of applications and identify candidates who meet the required qualifications [7].

Another important area of research is automated interview evaluation. Researchers have explored the use of language models and semantic similarity techniques to evaluate candidate answers during interviews. Sentence embedding models and transformer-based architectures can measure how closely a candidate's answer matches the expected answer. This allows the system to assign scores automatically without relying only on keyword matching. Such AI-based evaluation methods help in providing fair and consistent scoring of interview responses [8].

Online interview systems also face challenges related to cheating and identity verification. To address this problem, many studies have used computer vision techniques such as face detection and facial recognition to monitor candidates during online assessments. These systems analyze webcam video streams to detect the presence of a candidate and identify suspicious situations such as multiple people appearing on the screen. Computer vision-based monitoring improves the reliability and security of online interview platforms [9].

In addition to candidate evaluation, modern recruitment systems also focus on providing feedback and learning support to candidates. AI-based recommendation systems can analyze candidate performance and suggest topics for improvement, learning resources, and practice materials. These systems help candidates understand their strengths and weaknesses and improve their skills for future opportunities. Therefore, integrating resume analysis, interview evaluation, computer vision monitoring, and learning recommendations into a single system can create a more effective and intelligent recruitment platform [10].

III. WORKING PRINCIPLE

The AI-Powered Automated Interview System works by combining multiple technologies such as Artificial Intelligence, Natural Language Processing, Machine Learning, and Computer Vision to automate different stages of the recruitment process. The system is designed to evaluate candidates in a structured and efficient manner. Each stage of the system performs a specific task, starting from candidate registration to final report generation. The working principle of the system is explained below in a step-by-step manner.

A. Candidate Registration and Resume Upload

The first step of the system begins when a candidate registers on the platform. The candidate provides basic information such as name, email address, and educational institution. After filling in these details, the candidate uploads their resume in PDF format. The system stores the uploaded resume in the server and records the candidate information in the database using SQLite and SQLAlchemy. This step helps in maintaining organized records of all candidates participating in the recruitment process. The resume is later used for further analysis by the AI module to extract useful information related to the candidate's qualifications and experience [11].

B. Resume Text Extraction and AI-Based Analysis

Once the resume is uploaded, the system extracts the text content from the PDF file using a document processing library. The extracted text is then sent to an AI model for analysis. The AI model processes the resume and identifies important information such as skills, projects, achievements, education, and work experience. It also analyzes the candidate's experience level and suggests suitable job roles based on the information present in the resume. The AI generates structured information in JSON format, which makes it easier for the system to store and analyze the data. This automated resume analysis helps recruiters quickly understand the candidate profile without manually reading the entire document [12].

C. Role Selection by Candidate

After the resume analysis step, the candidate is allowed to select a job role for which they want to apply. Examples of roles include Python Developer, Data Analyst, or other technical positions. Based on the selected role, the system prepares a set of evaluation questions that match the required skills for that particular job. This ensures that candidates are

assessed according to the relevant technical knowledge and abilities required for the selected role [13].

D. MCQ Assessment

In the next stage, the candidate must complete a Multiple Choice Question (MCQ) test. The MCQ test is designed to evaluate the candidate's basic understanding of the chosen domain. For most roles, the questions are stored in CSV files and are loaded by the system when required. For some roles, such as Python Developer, the system dynamically generates MCQ questions using an AI model. Each candidate is given a fixed number of questions, and their answers are automatically evaluated by the system. A passing score is defined for the test. Only candidates who achieve the required score are allowed to proceed to the next stage of the interview process. This step helps filter candidates and ensures that only those with basic knowledge continue further in the evaluation process [14].

E. Verbal Interview Question Phase

After successfully passing the MCQ test, the candidate enters the verbal interview stage. In this stage, the system asks a series of technical and behavioral questions. Technical questions test the candidate's understanding of programming concepts, algorithms, and software development practices. Behavioral questions help evaluate the candidate's problem-solving approach, teamwork ability, and career motivation. Candidates are required to type their answers in text form. The responses are stored in the database so that they can be evaluated by the AI system in the next step [15].

F. AI-Based Answer Evaluation

The answers provided by the candidate during the interview are evaluated using semantic similarity techniques. The system uses sentence embedding models to convert both the candidate's answer and the reference answer into numerical vectors. These vectors are then compared using cosine similarity to determine how closely the candidate's answer matches the expected answer. The similarity score is used to calculate marks for each question. This approach allows the system to evaluate answers based on meaning rather than exact keywords. As a result, candidates who express correct ideas in different wording can still receive appropriate scores.

G. Face Detection for Anti-Cheating Monitoring

During the interview process, the system continuously monitors the candidate using a webcam. Computer vision techniques are used to detect the presence of faces in the video stream. The system checks whether a single face is present in the frame. If no face is detected or if multiple faces appear repeatedly, the system considers it as suspicious activity. In such cases, the interview session may be terminated automatically. This feature helps maintain fairness in the evaluation process and prevents candidates from receiving external assistance during the interview.

H. Coding Assessment Round

In addition to theoretical questions, the system also includes a coding round. In this stage, the candidate is given a programming problem that needs to be solved using code. The candidate writes the code in the provided interface. The system then runs the code against predefined test cases to verify whether the solution produces the correct output. If the code passes all the test cases, the candidate receives full marks for the coding round. Otherwise, partial or zero marks may be assigned depending on the correctness of the solution. This stage helps evaluate the candidate's practical programming skills and logical thinking ability.

I. AI-Based Learning Recommendations

After the completion of the interview process, the system analyzes the candidate's performance in different stages such as MCQ tests, interview answers, and coding tasks. Based on this analysis, the AI generates personalized feedback and suggestions for improvement. The system may recommend specific topics that the candidate should study further or suggest practice exercises related to their weak areas. These recommendations help candidates understand their mistakes and improve their knowledge for future opportunities.

J. Video Learning Suggestions

To support the candidate's learning process, the system also recommends educational videos related to interview preparation and technical topics. These videos are fetched using an online video platform API. The recommendations are selected based on the candidate's performance and the role they applied for. Providing video learning resources helps candidates gain additional knowledge and prepare better for future interviews.

K. Report Generation

The final stage of the system is report generation. After completing all evaluation stages, the system calculates the overall score of the candidate. A detailed report is generated in PDF format using a document generation library. The report contains information such as candidate details, MCQ scores, interview evaluation scores, coding performance, and improvement suggestions. Two types of reports are generated: one for the candidate and one for the administrator. The candidate report provides feedback and learning suggestions, while the administrator report summarizes the candidate's performance and helps recruiters make hiring decisions.

IV. METHODOLOGY

The methodology of the proposed AI-Powered Automated Interview System explains how different technologies and components are used together to automate the recruitment process. The system is designed in a modular manner so that each stage of candidate evaluation can be performed efficiently. The methodology includes resume processing, candidate assessment, AI-based evaluation, monitoring mechanisms, and report generation. The overall system follows a sequential workflow where the output of one stage becomes the input for the next stage.

A. System Architecture Design

The system follows a web-based architecture where the backend is developed using the Flask framework. Flask handles all server-side operations such as candidate registration, resume processing, question management, answer evaluation, and report generation. SQLite is used as the database to store candidate details, resume information, MCQ scores, interview responses, and final evaluation results. The modular architecture allows different components such as AI analysis, interview evaluation, and computer vision monitoring to work together smoothly. Each module communicates with the backend server to process data and return results to the user interface. This architecture makes the system simple, scalable, and easy to maintain [16].

B. Resume Data Processing

After the candidate uploads the resume, the system extracts the textual content from the PDF file using a document processing tool. Since resumes often contain unstructured text, Natural Language Processing techniques are applied to interpret the information. The extracted text is sent to an AI-based language model which analyzes the resume and identifies key details such as skills, projects, education, achievements, and years of experience. The model also identifies possible job roles that match the candidate's profile. The analyzed data is returned in a structured format such as JSON. This structured data is stored in the database and used later during candidate evaluation and report generation. Automated resume processing helps reduce the time required for manual resume screening [17].

C. MCQ-Based Knowledge Assessment

The system includes an MCQ assessment stage to test the candidate's fundamental knowledge in the selected domain. Questions are stored in CSV files for different job roles and are loaded dynamically when the candidate starts the test. For some roles, the system also generates questions dynamically using an AI model. Each candidate receives a fixed number of questions and must select the correct answer from the available options. The system automatically evaluates the candidate's responses by comparing them with the correct answers stored in the database. The score obtained in the MCQ test determines whether the candidate is eligible to proceed to the next stage of the interview process. This step acts as a filtering mechanism to ensure that candidates possess basic domain knowledge [18].

D. Natural Language Processing for Interview Evaluation

In the verbal interview stage, candidates provide written answers to technical and behavioral questions. Evaluating these answers manually would require significant effort and time. Therefore, the system uses Natural Language Processing techniques to perform automatic answer evaluation. The candidate's answer and the reference answer are converted into numerical representations called sentence embeddings. These embeddings capture the semantic meaning of the text rather than just individual words. The similarity between the two embeddings is calculated using cosine similarity. A higher similarity score indicates that the candidate's answer is closer in meaning to the expected answer. The system then converts this similarity score into marks for the candidate. This approach allows the system to evaluate answers more accurately even if the wording is different [19].

E. Computer Vision-Based Monitoring

Online interviews face challenges related to cheating and candidate authenticity. To address this issue, the system uses computer vision techniques for monitoring candidates during the interview process. The system accesses the candidate's webcam and continuously captures frames during the interview session. Face detection algorithms are applied to identify the number of faces present in each frame. If the system detects no face or more than one face repeatedly, it flags the situation as suspicious activity. In such cases, the interview may be terminated automatically to maintain fairness. Computer vision monitoring ensures that the candidate completes the interview without external assistance [20].

F. Coding Assessment and Test Case Evaluation

The system also evaluates practical programming skills through a coding round. Candidates are given programming problems that must be solved using code. The candidate's code is executed in the system environment and tested using predefined input test cases. The output generated by the candidate's code is compared with the expected output stored in the system. If the outputs match for all test cases, the candidate receives full marks for the coding question. This automated evaluation process helps measure the candidate's logical thinking, programming ability, and problem-solving skills in a fair and consistent manner.

G. Performance Analysis and Report Generation

After completing all evaluation stages, the system calculates the candidate's overall performance. The final score is derived from multiple factors such as MCQ performance, interview answer evaluation, and coding assessment results. Based on this performance data, the system generates a detailed report that summarizes the candidate's strengths and weaknesses. The report is generated in PDF format and includes important information such as candidate details, scores, feedback, and improvement suggestions. Two versions of the report are created: one for the candidate and another for the administrator. The administrator report helps recruiters make hiring decisions, while the candidate report provides learning recommendations and guidance for future improvement.

V. SYSTEM ARCHITECTURE

The system architecture of the proposed AI-Powered Automated Interview System defines how different components of the system interact with each other to perform the recruitment process automatically. The architecture follows a modular web-based design, where multiple technologies work together to perform tasks such as resume analysis, candidate evaluation, cheating detection, and report generation. The system integrates Artificial Intelligence, Natural Language Processing, Computer Vision, and database management to provide an efficient and reliable recruitment platform.

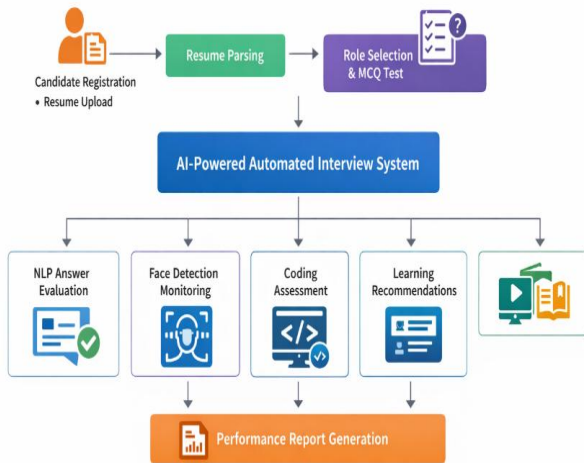


Fig.1-Proposed System Architecture Diagram

A. Overall System Structure

The proposed system is designed using a client–server architecture. In this architecture, the candidate interacts with the system through a web interface, while the server handles data processing and AI-based analysis. The frontend interface allows candidates to register, upload resumes, answer MCQ questions, participate in interviews, and view their performance reports. The backend server, developed using the Flask framework, processes candidate inputs and communicates with various modules such as AI services, databases, and computer vision monitoring systems. This architecture allows the system to manage multiple candidates while maintaining organized data processing and evaluation workflows [21].

B. Web Application Layer

The web application layer acts as the communication interface between the user and the system. Candidates access the system through a web browser and interact with different pages such as registration, test interface, interview interface, and result pages. The Flask framework manages the routing of requests and responses between the client and the server. When a candidate submits information or answers a question, the request is sent to the backend server where the data is processed and stored in the database. This layer ensures smooth communication between different system modules and provides an interactive experience for users [22].

C. Database Management Layer

The system uses SQLite as the database to store candidate information and evaluation results. SQLAlchemy is used as the Object Relational Mapping (ORM) tool to interact with the database in an efficient manner. The database stores different types of information such as candidate registration details, resume analysis results, MCQ scores, interview answers, coding results, and generated reports. Proper database management ensures that candidate data is organized and easily accessible when needed. This structured storage approach helps maintain system reliability and enables administrators to review candidate performance efficiently [23].

D. Artificial Intelligence Processing Layer

The AI processing layer is one of the most important components of the system. This layer is responsible for performing intelligent tasks such as resume analysis, MCQ generation, answer evaluation, and learning recommendation generation. The system sends candidate data to AI models that analyze textual information and generate useful insights. For example, resume data is processed to identify skills and experience, while interview answers are evaluated using semantic similarity techniques. By using AI-based analysis, the system can automatically evaluate candidate performance and provide accurate results without requiring manual assessment by human interviewers [24].

E. Computer Vision Monitoring Module

To maintain fairness during online interviews, the system includes a computer vision monitoring module. This module uses webcam input to detect faces during the interview session. Face detection algorithms analyze video frames and determine whether a single candidate is present in front of the camera. If the system detects multiple faces or if the candidate disappears from the frame for a long time, the system flags the activity as suspicious. This module helps prevent cheating and ensures that candidates complete the interview independently without external assistance [25].

F. Assessment and Evaluation Module

The assessment module is responsible for conducting different evaluation stages such as MCQ tests, verbal interview questions, and coding assessments. The system automatically records candidate responses and processes them using evaluation algorithms. MCQ answers are checked by comparing them with the correct answers stored in the system. Interview responses are evaluated using semantic similarity techniques, and coding solutions are tested using predefined test cases. The evaluation module calculates scores for each stage and combines them to generate the final performance result for the candidate.

G. Report Generation Module

The final component of the architecture is the report generation module. After completing all evaluation stages, the system compiles the candidate's performance data and generates a detailed report. The report includes candidate information, MCQ scores, interview evaluation scores, coding performance results, and AI-generated improvement suggestions. The report is generated in PDF format so that it can be easily shared and stored. Two versions of the report are produced: one for the candidate and one for the administrator. This helps both parties understand the candidate's performance and identify areas for improvement.

Overall, the system architecture integrates multiple technological components into a single automated recruitment platform. By combining web technologies, artificial intelligence, computer vision, and database management, the

proposed system provides a structured and efficient approach to conducting technical interviews and evaluating candidate skills.

VI. SYSTEM OVERVIEW

The AI-Powered Automated Interview System is designed to automate the recruitment process by integrating artificial intelligence, natural language processing, computer vision, and automated evaluation techniques. The system provides a structured platform where candidates can participate in different stages of assessment, including resume analysis, knowledge testing, interview evaluation, and coding assessment. The main objective of the system is to reduce manual effort in recruitment and provide fair and consistent evaluation of candidates.

The system begins with the candidate registration process. In this stage, the candidate enters basic personal information such as name, email address, and educational institution. After registration, the candidate uploads their resume in PDF format. The uploaded resume is stored in the system database and is used for further analysis. Once the resume is uploaded, the system performs resume analysis. The text from the resume is extracted and analyzed using artificial intelligence techniques. The system identifies important details such as candidate skills, educational background, projects, achievements, and work experience. This information helps the system understand the candidate's profile and determine suitable job roles.

After resume analysis, the candidate is allowed to select a job role for which they want to apply. The system supports multiple technical roles such as Python Developer, Data Analyst, and other related positions. Based on the selected role, the system prepares appropriate evaluation questions that match the required skills and knowledge for that role. The next stage is the MCQ assessment. In this stage, the candidate must answer multiple-choice questions related to the selected role. These questions test the candidate's basic understanding of programming concepts and technical knowledge. The system automatically evaluates the answers and calculates the score. Candidates who achieve the required passing marks are allowed to continue to the next stage of the process.

After passing the MCQ test, the candidate proceeds to the verbal interview stage. During this stage, the system presents technical and behavioral questions to the candidate. The candidate provides written responses to these questions. The system then evaluates the answers using semantic analysis techniques that measure the similarity between the candidate's answer and the expected answer. This allows the system to assign marks based on the meaning and correctness of the response. To ensure fairness during the interview, the system includes a face detection monitoring feature. The system accesses the candidate's webcam and continuously checks whether a single face is present in front of the camera. If the system detects suspicious situations such as multiple faces or

the absence of the candidate, it may terminate the interview session to prevent cheating.

In addition to theoretical evaluation, the system also includes a coding assessment round. In this stage, candidates are given programming problems that require writing code. The system runs the candidate's code against predefined test cases and verifies whether the output is correct. This helps evaluate the candidate's programming ability and logical thinking skills. After all assessment stages are completed, the system performs performance analysis. The candidate's scores from MCQ tests, interview evaluation, and coding assessment are combined to calculate the overall performance. Finally, the system generates a detailed performance report. The report contains candidate information, test scores, interview results, coding performance, and improvement suggestions. Personalized learning recommendations and educational resources may also be provided to help the candidate improve their skills.

Overall, the AI-Powered Automated Interview System provides an efficient and intelligent platform for conducting interviews and evaluating candidate abilities. By automating different stages of the recruitment process, the system helps organizations save time, improve evaluation accuracy, and make better hiring decisions.

VII. CONCLUSION

The AI-Powered Automated Interview System presented in this work aims to improve and simplify the recruitment process by using modern technologies such as Artificial Intelligence, Natural Language Processing, and Computer Vision. Traditional recruitment methods require significant human effort and time for reviewing resumes, conducting interviews, and evaluating candidate performance. These processes may sometimes lead to delays and inconsistent evaluation. The proposed system addresses these challenges by providing an automated platform that performs multiple stages of candidate assessment efficiently.

The system integrates several important modules including resume analysis, MCQ-based knowledge testing, AI-based interview evaluation, face detection monitoring, coding assessment, and report generation. Resume analysis helps extract useful information about candidate skills, projects, and experience. The MCQ assessment evaluates the candidate's fundamental technical knowledge, while the interview evaluation module analyzes candidate responses using semantic similarity techniques. This allows the system to evaluate answers based on their meaning rather than exact wording.

The use of computer vision technology helps ensure fairness during online interviews by detecting suspicious activities such as the presence of multiple faces or the absence of the candidate from the camera. The coding assessment module

further evaluates the candidate's programming ability by testing their solutions against predefined test cases. These combined evaluation stages provide a comprehensive assessment of the candidate's knowledge, technical skills, and problem-solving abilities.

Another important advantage of the system is the generation of detailed performance reports and personalized learning recommendations. These reports help candidates understand their strengths and areas where improvement is needed. At the same time, the administrator reports assist recruiters in making informed hiring decisions based on objective evaluation results.

Overall, the proposed system demonstrates how artificial intelligence and automation can significantly improve the recruitment process. By reducing manual workload, increasing evaluation accuracy, and maintaining fairness in candidate assessment, the AI-Powered Automated Interview System provides a practical solution for modern recruitment challenges. This system can be further expanded and improved to support large-scale recruitment processes and advanced AI-based interview techniques in the future.

VIII. REFERENCES

- [1] A. Upadhyay and K. Khandelwal, "Applying artificial intelligence: Implications for recruitment," *Strategic HR Review*, vol. 17, no. 5, pp. 255–258, 2018.
- [2] S. Raghavan, J. Barocas, J. Kleinberg, and K. Levy, "Mitigating bias in algorithmic hiring: Evaluating claims and practices," in *Proc. Conf. Fairness, Accountability, and Transparency*, 2020, pp. 469–481.
- [3] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.
- [4] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP*, 2019, pp. 3982–3992.
- [5] A. Vaswani et al., "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.
- [6] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Draft, Stanford University, 2021.
- [7] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2001, pp. 511–518.
- [9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
- [10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR Workshop*, 2013.
- [11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [13] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. ACM SIGKDD*, 2016, pp. 1135–1144.
- [14] J. Brown et al., "Language models are few-shot learners," in *Proc. Advances in Neural Information Processing Systems*, 2020.
- [15] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," To appear, 2017.
- [16] R. Feldman and J. Sanger, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2007.
- [17] M. Russell, *Mining the Social Web*, 3rd ed. O'Reilly Media, 2018.
- [18] F. Chollet, *Deep Learning with Python*, 2nd ed. Manning Publications, 2021.
- [19] T. Mitchell, *Machine Learning*. McGraw-Hill Education, 1997.
- [20] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. O'Reilly Media, 2019.
- [21] D. Sculley et al., "Hidden technical debt in machine learning systems," in *Proc. Advances in Neural Information Processing Systems*, 2015.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [23] J. Redmon et al., "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016.

[24] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[25] Google, "Gemini API documentation," Google AI, 2024. [Online]. Available: <https://ai.google.dev>