

Automated Recruitment System

**Mr. Mahesh Dhanve¹, Mr. Devendra Mehtre², Mr. Nikhil Daund³, Mr. Shrikant Daware⁴,
Mrs. Rupali Waghmode⁵**

Sinhgad Institute of Technology and Science, Pune University, Pune, India¹⁻⁴

Guide, Sinhgad Institute of Technology and Science, Pune University, Pune, India⁵

Abstract: The growing demand for efficient and scalable hiring processes has led to the adoption of intelligent automation in recruitment systems. Conventional recruitment methods, such as manual screening and traditional interviews, fail to provide consistent, unbiased, and real-time evaluation of candidate performance. This paper presents the design and development of an AI-powered automated recruitment system using Large Language Model (LLM) integration. The proposed system utilizes a full-stack architecture with Next.js, React, and PostgreSQL, integrated with LLM APIs for intelligent question generation and candidate evaluation. Candidates interact with an AI chatbot through text and voice-based communication, along with an integrated coding environment for technical assessments. The evaluation engine generates feedback, ratings, and hiring recommendations. Experimental validation demonstrates consistent performance and efficient automation of recruitment workflows.

Index Terms: AI Recruitment, LLM, Automated Interview System, NLP, Full Stack Development, Candidate Evaluation, Next.js, Supabase, Role-Based Access Control

I. INTRODUCTION

The recruitment process plays a vital role in determining the quality and productivity of an organization's workforce. However, traditional hiring methods often involve manual resume screening, repetitive interview processes, and subjective decision-making, which can lead to inefficiencies and inconsistencies. With the increasing number of applicants for each job position, recruiters face significant challenges in managing and evaluating candidates effectively within limited timeframes.

In recent years, advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have opened new possibilities for automating complex tasks in recruitment. Large Language Models (LLMs), in particular, have demonstrated strong capabilities in understanding human language, generating contextual responses, and evaluating textual data. These technologies provide an opportunity to transform conventional recruitment systems into intelligent, automated platforms that can enhance both efficiency and accuracy.

Several existing recruitment solutions attempt to incorporate automation, primarily focusing on resume parsing and keyword-based filtering. However, these systems often lack real-time interaction, adaptability, and comprehensive evaluation mechanisms. They fail to assess critical aspects such as communication skills, problem-solving ability, and practical technical knowledge, which are essential for making informed hiring decisions.

To address these challenges, this paper proposes an AI-powered Automated Recruitment System that integrates LLM-based intelligence into the hiring process. The system enables recruiters to create structured interviews and automatically generate relevant questions based on job requirements. Candidates participate through an interactive AI chatbot that supports both text and voice communication, along with an integrated coding environment for technical assessments. Furthermore, the system incorporates an evaluation engine that analyzes candidate responses, resumes, and coding performance to generate structured feedback, ratings, and hiring recommendations. By automating multiple stages of the recruitment workflow, the proposed system reduces manual effort, improves scalability, and enhances objectivity.

II. LITERATURE REVIEW

Recent studies in automated recruitment and AI-driven hiring systems have explored various approaches to improving candidate screening, interview automation, and evaluation accuracy. While individual components such as resume parsing and video interview analysis have been studied, a unified end-to-end system combining all these capabilities remains limited in existing literature. Table I summarizes the key existing works.

TABLE I Literature Review of Existing Systems

S.No	Title	Author	Year	Method / Methodology Used	Result / Conclusion
1	Smart Recruitment System	Rakshitha Dugyala, Vinay Kumar Gaddam, Harika Eroju, Mohana Varma Dantuluri, Madhubabu Ch	2024	NLP, Image Recognition	A smart and automated system for evaluating candidates' performance
2	AI-Powered Candidate Screening System	Dr. Neha Sharma, Dr. Rajeev Kumar	2023	NLP techniques to analyze resumes and job descriptions	Achieved a 30% reduction in time-to-hire and improved candidate-job fit by 25%
3	Automated Video Interview Analysis Using Deep Learning	Dr. Sandeep Verma, Dr. Priya Desai	2024	CNNs and RNNs to analyze facial expressions and speech patterns	Enhanced evaluation accuracy by 40% and reduced interviewer bias by 15%
4	LLM-Based Interview Question Generation for Technical Roles	Ankit Joshi, Priya Mehta	2024	GPT-based LLMs to dynamically generate role-specific technical questions	Questions rated 85% relevant by domain experts; reduced manual effort
5	Resume Parsing and Skill Extraction Using Transformer Models	Dr. Sunita Rao, Kiran Patil	2023	BERT-based transformer models to extract structured skill and experience data	Achieved 92% accuracy in skill extraction compared to keyword-based parsers
6	Bias Reduction in AI-Driven Hiring Systems	Dr. Anil Kulkarni, Dr. Meera Joshi	2024	Fairness-aware ML models with standardized evaluation rubrics	Reduced demographic bias by 35% while maintaining overall hiring accuracy

From the above review, it is evident that while individual components of recruitment automation have been studied extensively, no existing system provides a complete end-to-end solution that integrates AI-based question generation, multi-modal candidate interaction (text, voice, coding), and intelligent evaluation within a single unified platform. The proposed system bridges this gap.

III. PROBLEM STATEMENT

The recruitment process in many organizations is still largely manual, time-consuming, and prone to human bias. Recruiters are required to screen resumes, schedule interviews, conduct assessments, and evaluate candidates individually, which leads to inefficiencies and inconsistent decision-making. Additionally, traditional interview methods lack standardization, making it difficult to fairly compare candidates across multiple evaluation parameters.

With the increasing number of job applicants and the need for faster hiring cycles, there is a growing demand for intelligent systems that can automate and streamline the recruitment process. Existing systems often fail to provide real-time interaction, personalized assessment, and comprehensive evaluation in a single platform.

Therefore, there is a need for an automated recruitment system that leverages modern web technologies and Artificial Intelligence to:

- Conduct structured and interactive interviews

- Evaluate candidates objectively using AI
- Provide instant feedback and hiring recommendations
- Reduce manual effort and overall hiring time
- Ensure fairness and consistency across all candidates

IV. KEY LIMITATIONS OF EXISTING SOLUTIONS

Current recruitment platforms and tools suffer from several limitations:

Lack of End-to-End Automation

Most systems only support partial automation (e.g., resume screening or scheduling) but do not cover the complete interview lifecycle from creation to evaluation.

Limited Candidate Interaction

Traditional platforms rely on static questionnaires or video recordings, lacking real-time conversational interaction with candidates.

Absence of Intelligent Evaluation

Many tools do not utilize advanced AI/LLMs to analyze candidate responses deeply, resulting in surface-level assessments.

High Dependency on Human Recruiters

Recruiters still need to manually evaluate candidates, increasing workload and introducing subjective bias.

No Integrated Coding Assessment Environment

Existing platforms often require external tools for coding tests, leading to a fragmented user experience.

Delayed Feedback

Candidates and recruiters often wait for long periods to receive evaluation results, slowing down the hiring process.

Poor Role Separation

Most existing tools do not enforce strict role-based access, leading to data visibility issues between recruiters and candidates.

V. PROPOSED SYSTEM OVERVIEW

The proposed Automated Recruitment System is designed as a comprehensive, AI-powered platform that automates the entire recruitment lifecycle, from interview creation to candidate evaluation. The platform primarily serves two types of users: Recruiters and Candidates, each having a completely separate interface and access scope.

Recruiters can securely log into the system using email-based authentication powered by Supabase. Once authenticated, they gain access to a dashboard where they can create, manage, and monitor interviews. During interview creation, recruiters specify essential parameters such as job position, job description, interview duration, and type of interview (technical, HR, or mixed). Based on these inputs, the system leverages an LLM API to automatically generate relevant, context-aware interview questions.

After configuring the interview, the system generates a unique interview link that is shared with candidates. Candidates enter their basic details, upload their resume, and begin the interview. They interact with an AI-powered chatbot that conducts the interview dynamically, supporting both text and voice-based communication for a natural, engaging experience.

For technical roles, an integrated coding environment (IDE) is provided, allowing candidates to write and execute code directly within the platform without requiring any external tools.

Upon interview completion, all responses — chat transcripts, voice interactions, and coding solutions — are processed by the LLM-based evaluation engine. The engine generates:

- Detailed feedback on candidate performance across multiple parameters
- Quantitative ratings for communication, technical skills, and problem-solving ability
- Selected, Rejected, or Consider — a final hiring recommendation

These results are displayed on the recruiter dashboard, enabling quick, informed, and data-driven decision-making.

VI. SYSTEM ARCHITECTURE

The Automated Recruitment System follows a modular, layered architecture that integrates modern web technologies with AI capabilities. The system is designed to handle multiple concurrent users while maintaining performance, security, and scalability.

A. Architecture Overview

Frontend Layer

The frontend is built using **Next.js** (App Router) and **React**, providing a server-side rendered, high-performance user interface. The UI is fully responsive, supporting both desktop and mobile devices. Separate dashboard views are implemented for Recruiters and Candidates, with strict route-level access control enforced using middleware.

Backend Layer

The backend logic is handled through **Next.js API Routes** and **Supabase Edge Functions**. API routes manage interview creation, candidate registration, link generation, and result storage. All business logic is processed server-side, ensuring that LLM API keys and sensitive data are never exposed to the client.

Database Layer

A **PostgreSQL** database managed via **Supabase** is used for persistent data storage. The schema is organized into the following key tables:

- **users** — stores recruiter and candidate profiles with role identifiers
- **interviews** — stores interview configurations (job role, description, duration, type)
- **candidates** — stores candidate details and resume references per interview
- **responses** — stores full interview transcripts and coding submissions
- **evaluations** — stores AI-generated scores, feedback, and final recommendations

Supabase Row Level Security (RLS) policies enforce data access restrictions so that recruiters can only access their own interviews and candidates cannot access evaluation data directly.

AI and LLM Integration Layer

This is the core intelligence layer of the system. LLM APIs are integrated at two key stages:

- **Question Generation:** When a recruiter creates an interview, the job role and description are sent to the LLM API, which generates a structured set of relevant interview questions.
- **Candidate Evaluation:** After interview completion, the full conversation transcript, resume content, and coding submission are sent to the LLM for evaluation. The model returns structured JSON containing scores, detailed feedback, and a hiring recommendation.

Voice interaction is handled via browser-based **Web Speech API** for speech-to-text and text-to-speech conversion, enabling real-time voice communication without external dependencies.

B. Workflow / Methodology

1. **Recruiter Authentication:** Recruiter logs in securely via email-based authentication using Supabase Auth.
2. **Interview Creation:** Recruiter fills in job role, description, duration, and interview type. The system triggers the LLM API to auto-generate interview questions.
3. **Interview Link Generation:** A unique, shareable interview link is generated and stored in the database.
4. **Candidate Registration:** Candidate accesses the link, enters name, email, and uploads resume.
5. **AI-Based Interview Execution:** The AI chatbot conducts the interview using the pre-generated questions, adapting follow-ups based on candidate responses.
6. **Voice Interaction:** Candidate optionally interacts via voice; responses are transcribed in real time using the Web Speech API.
7. **Coding Assessment:** For technical roles, the candidate writes and executes code in the integrated IDE within the platform.
8. **Data Collection:** All responses, transcripts, and code submissions are stored securely in the PostgreSQL database via Supabase.
9. **AI Evaluation:** The LLM evaluation engine processes all collected data and generates structured feedback, quantitative ratings, and a final recommendation.
10. **Dashboard Display:** Results are presented to the recruiter on the dashboard for review and decision-making.

C. Key Features

- Real-time AI-driven conversational interviews
- Automated, LLM-based question generation from job descriptions
- Multi-modal input: text, voice, and coding
- Integrated browser-based IDE for coding assessments
- Structured evaluation with scores and hiring recommendations
- Role-based access control for Recruiters and Candidates
- Scalable and modular architecture built on Next.js and Supabase

VII. IMPLEMENTATION DETAILS**A. Technology Stack**

Table II summarizes the key technologies used in the implementation of the system.

TABLE II — Technology Stack

Component	Technology Used
Frontend Framework	Next.js 14 (App Router), React
Styling	Tailwind CSS
Backend / API	Next.js API Routes, Supabase Edge Functions
Database	PostgreSQL via Supabase
Authentication	Supabase Auth (Email/Password)
AI / LLM Integration	LLM API (GPT-based) for question generation and evaluation
Voice Interaction	Web Speech API (Speech-to-Text and Text-to-Speech)
Coding Environment	Integrated browser-based IDE
Hosting / Deployment	Vercel (Frontend), Supabase Cloud (Backend & DB)

B. Role-Based Access Control

The system enforces strict separation between Recruiter and Candidate roles. Upon registration, each user is assigned a role stored in the users table. Next.js middleware intercepts all route requests and validates the user's session and role before granting access. Supabase Row Level Security (RLS) policies add a second layer of enforcement at the database level, ensuring that:

- Recruiters can only read and write data belonging to their own interviews.
- Candidates can only access the specific interview they were invited to.
- Evaluation results are only visible to the recruiter, never directly to candidates.

C. LLM Prompt Engineering

Careful prompt engineering is applied at both stages of LLM interaction:

- **Question Generation Prompt:** The system provides the LLM with the job role, job description, interview type, and duration. The prompt instructs the model to return a structured JSON array of questions categorized by type (technical, behavioral, HR).
- **Evaluation Prompt:** The evaluation prompt includes the full interview transcript, uploaded resume text, coding submission, and a predefined evaluation rubric. The LLM is instructed to return a structured JSON object containing individual scores (out of 10) for communication, technical knowledge, problem-solving, and overall performance, along with detailed written feedback and a final hiring recommendation.

VIII. RESULTS AND PERFORMANCE EVALUATION

The system was tested with a group of 20 recruiters and 50 candidates across various job roles including software engineering, data analysis, and HR positions. The key observations from experimental evaluation are summarized below.

A. System Performance

- **Question Generation Time:** The LLM API generated a complete set of interview questions in an average of 3.2 seconds per interview, making it practical for real-time use.
- **Evaluation Processing Time:** Post-interview AI evaluation was completed in an average of 8.5 seconds, delivering near-instant results to the recruiter dashboard.
- **System Uptime:** The platform maintained 99.5% uptime during the testing period, demonstrating the reliability of the Supabase and Vercel-based deployment.
- **Concurrent Users:** The system successfully handled up to 30 concurrent interview sessions without performance degradation.

B. Evaluation Accuracy

To validate the accuracy of AI-generated evaluations, a subset of 20 interviews was independently evaluated by human recruiters. The AI-generated scores were compared against human scores across four parameters: communication skills, technical knowledge, problem-solving ability, and overall suitability.

The results indicate a strong agreement rate of over 85% between AI-generated and human evaluations across all parameters, validating the reliability of the LLM-based evaluation engine.

C. Recruiter Feedback

Post-deployment feedback was collected from 20 recruiters. Key findings include:

- **90%** of recruiters reported that the AI-generated questions were relevant and appropriate for the job roles specified.
- **85%** found the evaluation feedback detailed and actionable for making hiring decisions.
- **95%** agreed that the system significantly reduced the time and effort required for the initial screening phase.

IX. SECURITY CONSIDERATIONS

Security is a core design principle of the system. The following measures are implemented to ensure data integrity and privacy:

- **Authentication:** All users authenticate via Supabase Auth using secure email and password-based login. Session tokens are managed server-side to prevent unauthorized access.
- **Row Level Security (RLS):** Supabase RLS policies ensure that database queries automatically filter data based on the authenticated user's role and identity, preventing cross-user data access.
- **API Key Protection:** LLM API keys are stored as environment variables on the server and are never exposed to the client-side application.
- **HTTPS Enforcement:** All communication between the client and server is encrypted using HTTPS, protecting data in transit.
- **Input Validation:** All user inputs, including resume uploads and interview responses, are validated and sanitized on the server side before processing or storage.
- **Candidate Data Privacy:** Candidate personal information and interview data are accessible only to the recruiter who created the interview, enforced at both the API and database levels.

X. CONCLUSION

The Automated Recruitment System represents a significant advancement in the field of hiring and talent acquisition by integrating Artificial Intelligence with modern web technologies. The system effectively addresses the limitations of traditional recruitment methods, such as manual effort, lack of standardization, and delayed decision-making.

By automating the entire recruitment workflow — from interview creation to final evaluation — the system enhances efficiency and reduces the workload on recruiters. The use of LLM-driven question generation ensures that interviews are consistent and tailored to specific job roles. The conversational AI interface provides an interactive and engaging experience for candidates, closely simulating real-world interview scenarios.

The experimental results demonstrate that the AI evaluation engine achieves over 85% agreement with human evaluators, validating its reliability for production use. The system successfully handled concurrent interview sessions with minimal latency, and recruiter feedback confirmed high satisfaction with the quality of AI-generated questions and evaluations. The role-based architecture, built on Next.js and Supabase, ensures clear separation between recruiter and candidate experiences, with robust security enforced at both the application and database levels.

In conclusion, the proposed system demonstrates how AI can revolutionize recruitment by making it more efficient, scalable, fair, and data-driven. Future enhancements may include emotion and sentiment analysis, automated proctoring mechanisms, resume ranking using vector embeddings, and multi-tenant SaaS support for enterprise deployments.

REFERENCES

- [1]. R. G. Gonnade, "An Empirical Study on Recruitment Management Systems: Start of a New Era," Asset Analytics, pp. 389–401, Jan. 2023.
- [2]. K. Mariani, "The Use of AI and Algorithms for Decision-making in Workplace Recruitment Practices," Journal of Student Research, vol. 12, iss. 1, Feb. 2023.

- [3]. R. Dugyala et al., “Smart Recruitment System,” International Journal of Advanced Research in Computer Science, vol. 15, no. 2, 2024.
- [4]. N. Sharma and R. Kumar, “AI-Powered Candidate Screening System Using NLP,” International Journal of Artificial Intelligence and Machine Learning, vol. 8, no. 3, pp. 45–58, 2023.
- [5]. S. Verma and P. Desai, “Automated Video Interview Analysis Using Deep Learning,” IEEE Transactions on Human-Machine Systems, vol. 54, no. 1, pp. 112–121, 2024.
- [6]. A. Joshi and P. Mehta, “LLM-Based Interview Question Generation for Technical Roles,” Proceedings of the International Conference on AI in Human Resources, pp. 78–85, 2024.
- [7]. S. Rao and K. Patil, “Resume Parsing and Skill Extraction Using Transformer Models,” Journal of Natural Language Processing, vol. 30, no. 4, pp. 201–215, 2023.
- [8]. A. Kulkarni and M. Joshi, “Bias Reduction in AI-Driven Hiring Systems,” ACM Conference on Fairness, Accountability, and Transparency (FAccT), pp. 310–319, 2024.