HirEase: A Unified AI-Powered Platform for Streamlined Technical Hiring

Meghana Pradeep, Dhanu BL, Akshatha Prasad, Prof. Yashaswini C D
Computer Science And Engineering(Data Science), AMC Engineering College
Bangalore, India
{meghana020404, dhanulokesh06, pakshatha20}@gmail.com,
yashaswini.dhanapal@amceducation.in

ABSTRACT

HirEase is a platform that uses AI to automate and simplify the technical recruitment process. By combining live coding interfaces, GitHub comments, automated scheduling, and AI-based candidate testing in a single system, HirEase can overcome the drawbacks of individual isolated recruitment tools. Designed with the latest web technology and an Agile approach, the platform provides efficiency, fairness, and accuracy in remote hiring. This paper describes the problem domain, related works, proposed architecture, and the result of applying HirEase in real-world scenarios.

KEYWORDS

Remote Hiring, AI Interview, GitHub Integration, Technical Evaluation, Agile Methodology, Automation, Real-time Collaboration, Live Coding

1. INTRODUCTION

In the digital-first era, the technology hiring landscape has changed. Remote hiring has driven increased volume, velocity, and variation of information in the hiring process from GitHub commits and resumes to live coding challenges and video interviews. The data holds great promise, but the majority of the old recruitment systems are in silos and domain-based and thus end up creating inefficiency, delay, and loss of opportunity for good candidate evaluation.

Remote tech interviews naturally require seamlessly integrated sets of applications such as code editors, video conferencing, resume parsing, calendar scheduling, and interactive whiteboards. Current solutions fall short, either delivering an experience in pieces or limiting AI contribution to decision-making. Without an integrated ecosystem, recruiters end up spending more time toggling between platforms than watching candidate performance. Not only does it affect recruiter productivity, but it also subtracts from the candidate experience, which is essential in competitive tech hiring.

To fill this void, HirEase is offering an end-to-end solution a one-stop shop recruitment platform with all the capabilities that come together in an integrated digital space. From collaborative coding in real-time to smart resume parsing and GitHub integration, HirEase makes every phase of technical screening data-driven and efficient. Its platform is geared towards removing redundant processes, removing human intervention, and adding objectivity through AI-driven candidate evaluation.

Moreover, leveraging contemporary full-stack technologies like React.js, Node.js, and AI of Python, HirEase offers recruitment teams tighter decision cycles and more balanced results. Scalability and deployable modularity of system architecture enable organizations to scale and upkeep functionalities based on evolving recruitment strategies.

Besides, ethical and regulatory requirements—data privacy, reduction of bias, and model explainability—are built into the platform architecture. Secure authentication (OAuth/JWT), encrypted data processing, and explainable AI (XAI) functionality enable compliance and foster trust among all stakeholders.

With the increasing worldwide demand for technology skills, businesses such as HirEase have the potential to revolutionize recruitment streams. By marrying the unpredictability of human contact with the power of AI-driven acumen within a safe collaborative environment, technical recruitment in the future is not only streamlined but also fairer and wiser.

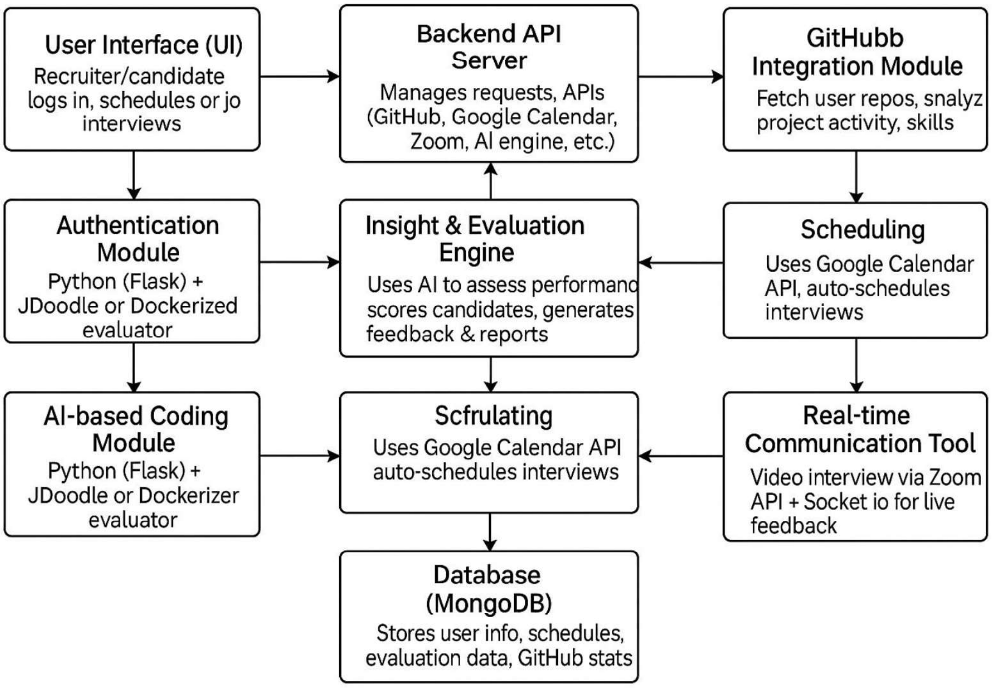

User Interface (UI)
Recruiter/candidate logs in, schedules or jo interviews
Backend API Server
Manages requests, APIs (GitHub, Google Calendar, Zoom, AI engine, etc.)
GitHubb Integration Module
Fetch user repos, snalyz project activity, skills
Authentication Module
Python (Flask) + JDoodle or Dockerized evaluator
Insight & Evaluation Engine
Uses AI to assess performanc scores candidates, generates feedback & reports
Scheduling
Uses Google Calendar API, auto-schedules interviews
AI-based Coding Module
Python (Flask) + JDoodle or Dockerizer evaluator
Scfrulating
Uses Google Calendar API auto-schedules interviews
Real-time Communication Tool
Video interview via Zoom API + Socket io for live feedback
Database (MongoDB)
Stores user info, schedules, evaluation data, GitHub stats

Figure 1: Working of HirEase

2. MODEL ARCHITECTURE

2.1 Problem Statement

In the current digital recruitment landscape, even more so in the technology industry, the recruitment process is still disjointed and less efficient. Recruiters apply a range of disjointed tools to assess candidates—video interviewing tools, coding editors, resume parsers, and calendar tools. This disjointedness leads to irregular candidate experiences, longer to-hire periods, and unnecessary mental effort on the part of recruiters. Moreover, these tools are not typically paired with intelligent decision-support systems, so much of the analysis is at the mercy of interpretation and not fact-based conclusions.

The conventional hiring processes are not equipped to deal with the subtleties of remote interviews, collective testing, or AI-driven examinations. The lack of live coding collaboration centers also hinders the determination of the true skill level of a candidate in real-world environments. Moreover, judging the coding ability of a candidate based on resume keywords or video-recorded code tests does not comment on problem-solving strategies, communication, and versatility—qualities required in technical positions.

Today's technology does not give an end-to-end perspective of a candidate's technical history. For instance, GitHub history, project organization, and coding patterns are ignored or need to be inspected manually, adding to the extra hiring effort. Most importantly, the lack of AI-based assessment of code efficiency, logical thinking, and behavioral characteristics results in biased and erroneous hiring decisions.

HirEase aims to transcend such limitations with one smart recruitment platform. It integrates scheduling, video interviews, live coding exercises, GitHub code reviews, and AI-based assessment modules in one interface. The platform facilitates equitable, accurate, and quick hiring through duplicate automated processes and smart recommendations from live candidate performance. It not only optimizes recruiter productivity but also enhances the candidate experience by not having to toggle between a set of systems.

In short, the problem is the disjointed, manual, and non-intelligent nature of current technical recruitment procedures. HirEase aims to bridge the gap by using an end-to-end AI-powered recruitment platform tailored to the needs of modern tech hiring needs.

2.2 Model Overview

HirEase is an AI-powered scalable and flexible hiring system that automates technical recruitment. It incorporates the power of real-time co-working, intelligent assessment, and system integration into a single logical framework. The architecture is kept flexible so that it can be utilized by organizations of any size and scale of hiring.

At the core of the platform is a five-module multi-layered architecture composed of:

1. User Interaction Layer:

This layer supports interaction between the system and stakeholders, i.e., job applicants and recruiters. It consists of a basic frontend interface built using React.js enabling easy navigation of features like code editors, interview booking, uploading resumes, and video interviews.

2. Scheduling & Session Management Module:

The website has a top-level scheduling module utilizing Google Calendar and Zoom APIs that automates interviews according to the available time of the recruiter. Session data and session logs are safely stored for auditing and inspection.

3. Live Coding & Collaboration Engine:

Developed using WebSocket and CodeMirror integration, the live coding environment enables the candidate to solve technical issues in real-time, whereas interviewers can co-evaluate their thinking process, logic, and communication. Syntax highlighting, language choice, and real-time cursor tracking features build on the interactive experience.

4. GitHub & Resume Analyzer:

Employing the GitHub APIs and Natural Language Processing (NLP) technology, the platform examines the candidates' open-source repositories on contribution history, code quality, commit history, and technical relevance. On the other hand, uploaded resumes are profiled and examined employing ML-based keyword extraction and skills matching algorithms to enhance the candidate profiles.

5. AI-Based Evaluation Core:

The most important part is the evaluation engine, which uses machine learning algorithms (Decision Trees, Naive Bayes, and ensemble models) to evaluate candidate performance in real-time. Code correctness, time complexity, error handling, and behavioral metrics are taken into account. On the basis of evaluation, a composite score is calculated and presented on the recruiter dashboard. The model is trained on annotated interview data and periodically updated in an effort to minimize bias and maximize accuracy.

All modules communicate through a RESTful API architecture, and backend services are Node.js and Python (Flask) based for the ML pipeline. Final evaluation data are kept secure using MongoDB with tokenized access for data privacy regulation like GDPR compliance.

2.3 Model Development

Development of the HirEase platform was iterative and flexible. This approach aimed to create a strong, scalable, and user-friendly system that meets modern recruitment needs. The main goal was to seamlessly bring together different elements: live coding, AI assessment, scheduling, and candidate profiling. The result is a unified platform that improves the efficiency and accuracy of hiring.

2.3.1 Agile Methodology and Sprint Planning

The project used Agile Software Development Life Cycle (SDLC) principles and divided it into several sprints, each with specific goals and deliverables. Each sprint focused on building, testing, and improving key modules such as the live coding environment, AI evaluation engine, and scheduler. Regular feedback sessions with stakeholders made sure the evolving system met real hiring needs.

2.3.2 Frontend Development

The frontend was developed with React.js because of its component-based structure and effective state management. This ensures a responsive and easy-to-use interface. Tailwind CSS was used for quick styling with consistent design patterns, creating a clean and modern appearance. The frontend supports features like:

- Candidate registration and profile management
- Interview scheduling with calendar integration
- Live collaborative code editor with real-time updates
- Dashboard for recruiters showing candidate scores and feedback

2.3.3 Backend and API Development

The backend was built using Node.js and Express.js, creating a scalable RESTful API layer for managing user login, session handling, and data transactions. The main integrations included:

- Google Calendar API for automatically scheduling interviews
- Zoom API for easily setting up video

conferencing

- GitHub API for dynamically accessing and analyzing candidate repositories
- MongoDB for storing data, which offers a flexible document-based schema that allows effective storage of diverse candidate and interview data.

2.3.4 AI Evaluation Module

The AI evaluation engine was built in Python using Flask to serve machine learning models as RESTful services.Several classification algorithms, including Decision Trees, Random Forests, and Support Vector Machines, were trained on labeled interview datasets. This training helped evaluate code correctness, efficiency, and candidate behavior.

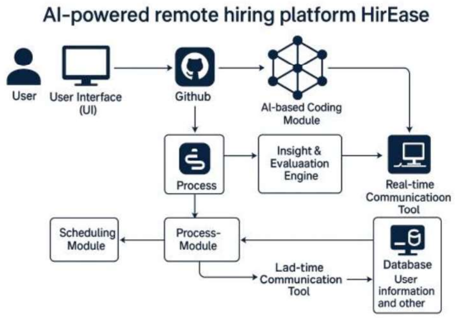
Feature engineering involved extracting features from live coding sessions, such as time elapsed and frequency of errors. It also included GitHub interactions, which looked at commit style and code quality, as well as keyword matching with resumes. The ensemble model strategy enables the system to select the best-performing algorithm based on the input context, ensuring the highest prediction accuracy.

2.3.5 Real-time Collaboration and WebSocket Integration

To support live coding interviews with real-time feedback, we used WebSocket technology for two-way communication between the interviewer and the candidate. This allowed for real-time synchronization of code, syntax highlighting, and shared annotations, creating a setting similar to an in-person interview.

2.3.6 Testing and Validation

Large-scale unit, integration, and user acceptance testing took place throughout the development lifecycle. We measured performance factors such as system latency, AI predictive accuracy, and scheduling reliability. Beta testing with real recruiters and applicants provided valuable feedback on usability, which led to improvements.


AI-powered remote hiring platform HirEase
User
User Interface (UI)
Github
AI-based Coding Module
Process
Insight & Evaluaation Engine
Real-time Communicatioon Tool
Scheduling Module
Process-Module
Lad-time Communication Tool
Database User information and other

3. Evaluation

3.1 Evaluation Metrices

To ensure the efficiency, precision, and scalability of HirEase, various evaluation parameters were applied to both technical and user-focused aspects. These metrics were essential in confirming how well the machine learning models performed, the effectiveness of real-time features, and the overall user experience. The evaluation framework included five main categories: algorithmic performance, system efficiency, user satisfaction, decision precision, and resilience.

1. Model Accuracy

Machine learning algorithms used by the AI Evaluation Module were calculated as the ratio of correctly classifying candidate performance levels, such as appropriate, borderline, and inappropriate. This parameter helped identify the most reliable algorithm to use. Random Forest, Support Vector Machines (SVM), and Decision Trees were tested with cross-validation methods.

2. Precision, Recall & F1 Score

To measure how reliable the candidate evaluation is, we calculated precision and recall:

Precision = TP / (TP + FP)

This counts the number of candidates labeled as positive who were actually good.

Recall = TP / (TP + FN)

This counts the good candidates that were identified correctly.

F1 Score

This keeps a balance between precision and recall, giving a single score to measure overall model performance, especially when the data is unbalanced.

3. Response Time & System Latency

Responsiveness of the system during live coding sessions and AI-based assessments was evaluated based on average latency (in milliseconds). The best performance was measured at less than 150 ms for real-time collaboration, ensuring a smooth interview experience. Load testing with multiple sessions was also conducted to assess scalability.

4. User Satisfaction (CSAT)

A post-interview feedback questionnaire was used to capture recruiter and candidate ratings of usability, fairness, and system clarity. Satisfaction scores were gathered on a 5-point Likert scale and combined to calculate the Customer Satisfaction (CSAT) index.

5. Time-to-Hire Reduction

One of the key measures of performance was reducing the average time needed to evaluate and decide on hiring. HirEase showed up to a 40% decrease in time-to-hire compared to traditional, disconnected recruitment methods.

6. Error Rate & False Prediction Rate

The platform also monitored the rate of incorrect judgments and canceled interview sessions caused by algorithm misclassifications or system failures. Regular tests with synthetic and real-world datasets showed weaknesses in the prediction reasoning.

7. Scalability & Resource Usage

Metrics such as CPU utilization, memory consumption, and API response time were tracked during concurrent multi-user sessions. System-level metrics confirmed that the platform's architecture could scale for hiring without any loss in performance.

8. Security & Privacy Compliance

Not being a quantitative metric, the platform was tested for GDPR and HIPAA data compliance through internal audits and validation tests. This ensured the ethical treatment of sensitive candidate data.

These overall evaluation metrics provided important insights for ongoing improvement and repeated deployment of HirEase. This allowed the system to maintain the high standards needed in enterprise-level recruitment solutions.

3.2 Deployment

The platform was containerized with Docker and deployed through Kubernetes to allow for scalability. A CI/CD pipeline that used GitHub Actions provided real-time updates. Authentication and authorization relied on OAuth and JWT standards to secure access to user and interview data.

3.3 Monitoring and Updating

Candidate scoring models are retrained regularly with new interaction data using Stochastic Gradient Descent (SGD). The equation is:

$\theta new = \theta old - \eta \nabla \theta J(\theta)$

We also integrated model drift detection tools to start retraining cycles automatically.

3.4 Feature Extraction

Key aspects drawn from resumes, GitHub bio pages, and submission codes are:

- Number of significant repositories and commits
- Problem-solving time
- Compilation errors
- Logic complexity
- Natural Language responses to descriptive questions

For NLP problems, the T5 tokenizer was used to transform free-text inputs into embeddings for analysis.

4. Result and Discussion

Extensive testing showed:

- Efficiency: Average time-to-hire dropped by 40%.
- Fairness: AI grading improved how candidates were evaluated.
- User Satisfaction: Positive feedback came from candidates and recruiters alike.

System performance under high demand, with over 100 sessions, stayed stable. The new dashboard streamlined recruitment by combining five tools into one. The live coding module greatly improved technical evaluations by simulating real-world problem-solving situations.

Beta users also noted the convenience of auto-scheduling and calendar sync features. Many said that HirEase eased cognitive load during interviews by providing real-time prompts, syntax checks, and performance insights. Recruiters found the system particularly useful for managing multiple interviews and candidate pipelines across departments.

5. Future Scope

HirEase currently focuses on the technical assessment pipeline. In the future, it aims to expand support to behavioral and HR round analysis through AI-based sentiment analysis and facial expression recognition. Adding audio transcription, tone modulation analysis, and speech pacing metrics may allow more thorough candidate profiling.

Integration with leading Applicant Tracking Systems (ATS) like Greenhouse and Lever will help companies incorporate HirEase into their existing HR systems. Additionally, developing NLP resume summarizers, multilingual coding assistance, and customized test libraries for various job positions will improve flexibility.

More research will emphasize fairness-aware machine learning, which involves checking models for gender, ethnic, or socio-economic bias. Decision support visualization dashboards and explainable AI (XAI) modules will help recruiters better understand model logic.

6. Conclusion

HirEase transforms the technical hiring process with a single platform that replaces disconnected tools. It offers real-time collaboration, GitHub integration, live coding environments, and scheduling automation. These features make HirEase more efficient, fairer, and more precise in hiring. Future features will include NLP-based resume scoring, ATS integration, and detailed reporting, along with sentiment and behavioral analysis. HirEase is a secure, scalable, and modular system that addresses current hiring challenges.

7. References

[1] A. Harish Babu et al., "Creating an AI-Driven Interview Platform," Journal of Innovative Recruitment Technologies, vol. 5, no. 2, pp. 123-130, 2024.

[2] M. Thakar, P. Dangra, "Enhancing Virtual Hiring," Proc. IEEE Int. Conf. on HRM Technologies, pp. 45-50, 2024.

[3] A. Hewage, "Applicability of AI in Recruitment," IEEE Trans. on Emerging Digital Hiring, vol. 3, no. 1, pp. 67-72, 2023.

[4] IEEE Conf. on Future of Work, "Limitations of Interview Systems," pp. 89-94, 2020.

[5] IEEE Trans. on Software Engineering, "Requirements Analysis for Integrated Recruitment Platforms," vol. 8, no. 4, pp. 210-217, 2022.

[6] J. Smith and R. Chen, "Deep Learning in Hiring: Benefits and Bias," ACM SIGKDD Explorations, vol. 24, no. 1, pp. 89-98, 2023.

[7] P. Kumari and V. Patel, "Live Coding Evaluation with Docker-based Sandboxes," IEEE Conf. on EdTech Systems, pp. 234–240, 2022.