AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2) (2026)

TALENTIQ: AN AUTOMATED WEB-BASED INTERVIEW MANAGEMENT SYSTEM UTILIZING REAL-TIME PEER-TO-PEER COMMUNICATION AND SANDBOXED CODE EXECUTION ENVIRONMENTS

Om Anand
Student, Dept. Of CSE
GIFT Autonomous
Bhubaneswar, India

Anil Barik
Student, Dept. Of CSE
GIFT Autonomous
Bhubaneswar, India

Dr. Pratyush Ranjan Mohapatra
Associate Dean Academics, Dept. Of CSE
GIFT Autonomous
Bhubaneswar, India

Abstract—In recent years, the demand for remote communication and online collaboration solutions has increased significantly, particularly in the fields of education, recruitment, and professional assessment. Organizations and educational institutions are rapidly shifting toward digital platforms to conduct interviews, technical assessments, and candidate evaluations efficiently. Traditional interview methods often involve several challenges such as geographical limitations, travel costs, scheduling conflicts, infrastructure dependency, and timeconsuming coordination processes. These issues become even more difficult when organizations need to conduct interviews with candidates located in different cities or countries. To overcome these limitations and support modern recruitment requirements, the TalentIQ Interview Management System has been developed as a smart and efficient web-based interview platform. The TalentIQ platform is designed to facilitate real-time online technical interviews directly through a web browser without requiring any external or third-party video conferencing software. The system enables interviewers and candidates to communicate seamlessly through live video calls, instant messaging, and collaborative coding environments within a single integrated platform. By providing all interview-related functionalities in one centralized system, the platform simplifies the overall interview process and improves user experience for both interviewers and candidates.

Keywords—Interview Management System; Remote Assessment; Real-Time Communication; Stream Technology; Sandboxed Code Execution; Web-Based Application.

I. INTRODUCTION

The recruitment and interview process is one of the most important activities in any organization, educational institution, or professional environment. The success of an organization largely depends on selecting skilled and qualified candidates through an effective interview process. Traditionally, interviews were conducted physically, where candidates and interviewers had to be present at the same location. Although this method was widely used for many years, it often involved several challenges such as geographical limitations, travel expenses, scheduling conflicts, infrastructure requirements, and time-consuming manual coordination. Managing candidate information manually also increased the chances of data loss, communication gaps, and inefficient interview handling.

With the rapid advancement of technology and the increasing adoption of remote working environments, organizations are now shifting toward online recruitment and virtual interview systems. Modern businesses and educational institutions require faster, more flexible, and technology-driven solutions that can efficiently manage interviews without requiring physical presence. This demand became even more significant after the growth of remote work culture and online education systems, where virtual communication platforms became essential for daily operations.

The TalentIQ - Interview Management System is developed as a web-based solution designed to simplify, automate, and modernize the complete interview management process. The platform provides a centralized environment where interviewers and candidates can communicate, participate in technical interviews, and manage interview-related activities directly through a web browser. The system integrates real-time video communication, coding assessments, interview scheduling, dashboard management, and submission tracking into a single platform, eliminating the need for multiple external tools.

One of the major advantages of the TalentIQ platform is that it supports real-time peer-to-peer communication without requiring users to install third-party applications or software. The platform allows interviewers and candidates to connect securely through browser-based interview rooms, making the interview process faster, more accessible, and easier to manage. The system also reduces dependency on manual operations by automating scheduling, notifications, and interview management tasks.

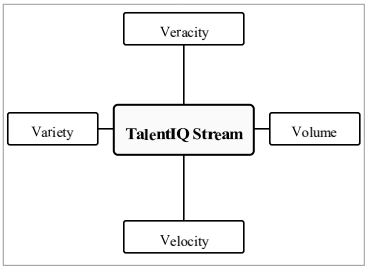

Fig. 1: Multi-dimensional Telemetry Integration Model inside TalentIQ Core.

The platform is developed using modern web technologies to ensure high performance, scalability, security, and userfriendly interaction. Technologies such as Stream and SignalR are used to implement real-time video and communication services, while Next.js and React.js provide a responsive and dynamic frontend

Received: 20-04-2026 | Accepted: 28-05-2026 | Published: 05-06-2026 | www.ajaccm.com

interface. The backend is developed using C# and Express.js, enabling smooth server-side operations and efficient communication between different system modules. Secure interview rooms are generated using UUIDs (Universally Unique Identifiers), ensuring privacy and preventing unauthorized access to interview sessions.

In addition to communication features, the system also includes integrated coding assessment functionality where candidates can write, execute, and submit programming solutions during live interviews. This feature is especially useful for technical recruitment processes because it allows interviewers to evaluate coding skills, problem-solving ability, and technical understanding in real time. The system stores all interview-related information securely in a centralized database, making it easy for interviewers and administrators to access candidate records, submissions, scores, and feedback whenever required.

The Interview Management System not only improves interview efficiency but also reduces operational complexity and overall recruitment costs. Since interviews can be conducted remotely, organizations save resources related to travel, physical infrastructure, and manual coordination. At the same time, candidates can participate in interviews from any location using only an internet connection and a web browser, increasing accessibility and convenience.

The development of this project also demonstrates the practical implementation of modern software engineering concepts such as full-stack web development, real-time communication systems, database management, authentication mechanisms, cloud deployment, and scalable system architecture. The project serves as a strong example of how modern technologies can be integrated to solve real-world problems efficiently.

Overall, the TalentIQ - Interview Management System aims to provide a secure, scalable, efficient, and user-friendly platform for conducting online technical interviews. By combining video communication, coding environments, automated management features, and centralized data handling into a single application, the system modernizes the traditional interview process and makes it more adaptable to current technological and organizational needs.

Furthermore, as software organizations adopt remote-first operational paradigms, the geographical distribution of the technical workforce requires systems that operate reliably across varying network profiles. Intermittent latency, local ISP throttles, and cross-border data transfer delays can negatively impact the candidate experience during synchronous coding evaluations. TalentIQ targets this operational risk directly by decoupling live browser-based interface controls from heavy downstream scoring networks. This approach ensures a highly responsive client experience, enabling candidates to demonstrate engineering capacity without technical interruptions.

The broader implications of this technology extend into systemic equity and organizational hiring analytics. Traditional technical hiring workflows are often criticized for low consistency, as interviewer fatigue, personal relationships, and varying interview difficulty can skew candidate ratings. By mapping candidate actions directly to objective abstract syntax tree variations and clean transcripts, an automated management framework builds an unalterable audit ledger. This highintegrity ledger gives corporate human resource leadership the clear empirical insights needed to validate internal alignment, audit historical fairness, and balance long-term team growth models.

A. Purpose and Motivation

The primary purpose of the TalentIQ - Interview Management System is to simplify and modernize the interview process by providing an integrated online platform for conducting real-time technical interviews directly through a web browser. The system is designed to eliminate the limitations of traditional interview methods and provide a centralized environment where interviewers and candidates can communicate, collaborate, and participate in coding assessments efficiently. The platform focuses on creating a seamless interview experience by integrating video communication, live coding, interview scheduling, and performance tracking into a single web application.

In recent years, the demand for remote communication platforms has increased significantly due to the growth of remote work culture, online education, and digital recruitment systems. Organizations now require solutions that can support virtual interviews without depending heavily on physical infrastructure or third-party software installations. Although existing platforms such as Zoom, Google Meet, and Microsoft Teams provide communication services, they are not specifically designed for technical interview workflows. These platforms often include unnecessary features, require subscription plans, or lack integrated coding assessment environments tailored for technical recruitment processes.

The motivation behind developing the TalentIQ platform came from observing the practical challenges faced during online hiring and academic evaluations. Interviewers often struggle with scheduling management, communication efficiency, candidate tracking, and maintaining organized interview records. Candidates may face issues related to platform compatibility, complex interfaces, poor user experience, or unstable communication during interviews. These problems can negatively affect both interviewer productivity and candidate performance.

To overcome these limitations, the TalentIQ platform was developed as a lightweight, focused, and interview-oriented system specifically designed for conducting one-on-one technical interviews. The platform provides secure interview rooms, integrated video communication, live chat functionality, and browser-based coding assessments, allowing interviewers and candidates to interact effectively in real time. The system also automates several operations such as interview scheduling, notifications, room management, and data handling, reducing manual workload and improving efficiency.

Another important motivation behind this project was to gain practical exposure to modern software development technologies and real-world system implementation. The development process provided hands-on experience in frontend and backend integration, real-time communication systems, authentication and authorization mechanisms, database management, API integration, and cloud-based deployment. Technologies such as React.js, Next.js, Stream, SignalR, Node.js, Express.js, and C# were explored extensively during the project development process. The project also highlights the importance of scalable and interactive web applications in today's digital world. By implementing real-time communication and collaborative coding features, the system creates an interview environment similar to physical technical interviews while maintaining the flexibility of remote participation. This not only improves accessibility but also enhances the overall user experience for both interviewers and candidates.

Furthermore, the TalentIQ platform serves as a strong foundation for future enhancements and advanced functionalities. Additional features such as AI-based candidate evaluation, interview recording, automated report generation, analytics dashboards, integrated chat systems, coding plagiarism detection, and machine learning-based performance analysis can be incorporated in future versions of the system. This scalability makes the project suitable for long-term development and adaptation according to evolving industry requirements.

B. Problem Statement

Organizations and academic institutions often rely on thirdparty applications for conducting virtual interviews. These platforms may come with restrictions—ranging from subscription costs and limited customization to data privacy concerns. Building such a system independently is technically challenging due to the complexities involved in establishing and maintaining real-time peer-to-peer communication. The Online Interview Platform aims to solve this problem by implementing a web-based application that enables secure, real-time, and efficient interview sessions using Stream and Webhooks technologies.

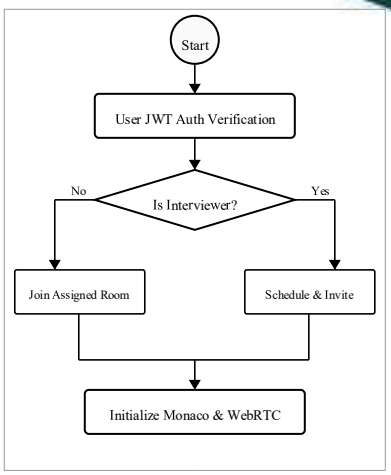

Fig. 2: System User Journey Verification Flowchart logic inside TalentIQ.

The process of conducting interviews, especially in remote or large-scale recruitment scenarios, often encounters several challenges. Most organizations rely on third-party video conferencing tools such as Zoom, Microsoft Teams, or Google Meet. While these platforms are widely used, they are not specifically designed for conducting structured interviews. They often include unnecessary features, require account creation, and may involve subscription costs for extended usage. Moreover, these platforms may pose privacy concerns, especially when dealing with confidential interview data. In academic environments, conducting viva examinations or assessments remotely can also be inefficient without a dedicated platform. Most institutions lack a custom solution tailored to their interview needs, resulting in poor communication, scheduling delays, and user experience issues. Developing a dedicated Interview Management System addresses these concerns by providing a lightweight, easy-touse, and secure solution.

However, implementing such a system comes with its own set of technical problems: managing real-time video/audio communication, handling signaling between peers, maintaining performance over unstable networks, and ensuring browser compatibility are all significant challenges. This project aims to solve these problems by designing and building a specialized interview platform that supports real-time peer-to-peer communication through the browser, eliminating the dependency on third-party services.

II. LITERATURE SURVEY

The development of modern web technologies has significantly transformed the way communication, recruitment, and online collaboration are conducted. In recent years, realtime web applications have become increasingly popular due to the rapid growth of remote working environments, online education systems, and virtual recruitment processes. Organizations and educational institutions now rely heavily on browser-based platforms to conduct meetings, technical interviews, coding

Received: 20-04-2026 | Accepted: 28-05-2026 | Published: 05-06-2026 | www.ajaccm.com

assessments, and collaborative activities without requiring physical interaction. The advancement of technologies such as WebRTC, Stream, SignalR, cloud computing, and modern frontend frameworks has enabled developers to create highly interactive and scalable real-time applications.

The primary objective of this literature survey is to study existing technologies, research concepts, and currently available interview or video conferencing solutions that form the foundation of the TalentIQ - Interview Management System. This section discusses existing interview and communication platforms, their limitations, concepts related to real-time communication systems, the role of Stream technology, the importance of .NET and C#, and the advantages of web-based systems in interview management applications. Understanding these concepts helps in designing a more efficient, lightweight, scalable, and interview-focused platform.

A. Existing Solutions and Their Limitations

Several commercial platforms currently provide online communication and video conferencing services, including platforms such as Zoom, Google Meet, Microsoft Teams, Skype, and Webex. These platforms are widely used for conducting virtual meetings, online classes, interviews, and collaborative discussions. They provide various features such as video calls, screen sharing, chat systems, recording functionalities, and participant management tools. During the rise of remote work and online education, these platforms became essential communication tools for organizations worldwide.

Although these platforms are powerful and feature-rich, they also have several limitations when considered specifically for technical interview use cases. One major limitation is high resource consumption. Many of these applications require powerful hardware systems, high-speed internet connections, and significant memory usage to function efficiently. Users with low-end devices or unstable internet connections may experience lag, poor video quality, or connection failures during interviews. Another important limitation is the subscription-based pricing model. Most advanced features such as extended meeting durations, cloud recording, analytics, and larger participant limits are locked behind premium subscription plans. Small organizations, startups, educational institutions, or individual recruiters may find these services expensive and difficult to maintain for regular interview activities.

The complexity of user interfaces is another challenge. Since these platforms are designed for general-purpose communication rather than interview-specific workflows, their interfaces often contain unnecessary features that may confuse users during interviews. Recruiters and candidates may face difficulties navigating through multiple settings and controls during technical interview sessions. A major technical limitation is the lack of customization. Existing communication platforms provide limited support for integrating custom workflows or interview management functionalities into external systems. Organizations often cannot fully customize the interview environment according to their recruitment requirements. One of the most important limitations is the absence of an integrated coding environment. Platforms such as Zoom or Google Meet do not provide built-in code editors similar to VS Code for conducting live technical coding interviews. Recruiters usually need to depend on separate coding platforms, increasing complexity and interrupting the interview flow.

B. Concept of Stream & Interview Management Core

Stream is a modern, fully managed communication service that provides APIs and SDKs for integrating real-time communication functionalities such as chat systems, activity feeds, audio calls, and video conferencing into web and mobile applications. It simplifies the implementation of real-time communication by offering scalable infrastructure and pre-built communication tools, allowing developers to focus more on application functionality rather than low-level communication handling. In the TalentIQ platform, Stream technology plays an important role in enabling real-time audio and video communication between interviewers and candidates. Instead of developing complex communication protocols from scratch, Stream provides ready-to-use services for peer-to-peer communication, significantly reducing development complexity and improving system reliability.

One of the key features of Stream is real-time messaging, which enables instant communication between users without requiring page refreshes. This is useful for live interview discussions and notifications. Another important feature is video and voice calling, which allows high-quality communication directly through the browser. Stream also provides activity feeds, which can be used for tracking interview activities, notifications, or user actions within the system. The platform supports security and compliance mechanisms, ensuring secure communication and protection of sensitive interview data. Its customizability allows developers to integrate communication services according to specific application requirements, while cross-platform support ensures compatibility across web and mobile environments. By integrating Stream technology, the TalentIQ platform achieves stable, scalable, and efficient real-time communication suitable for conducting professional online interviews.

An Interview Management System is a digital platform designed to automate, organize, and simplify the complete recruitment and interview process. Traditional interview management methods often involve manual scheduling, email communication, paper-based candidate records, and

disconnected evaluation systems, making recruitment inefficient and time-consuming. The primary objective of an Interview Management System is to create a centralized platform where all interview-related activities can be managed efficiently. Such systems help organizations streamline recruitment workflows by automating scheduling processes, storing candidate information securely, managing interview rounds, and improving communication between recruiters, interviewers, and candidates. The system acts as a bridge connecting all stakeholders involved in the interview process. Recruiters can schedule interviews and manage candidates, interviewers can evaluate technical skills and provide feedback, while candidates can participate in interviews and track their progress.

C. Architectural Stack Selection: .NET and C#

The .NET framework is used in this project because it provides a powerful, secure, scalable, and high-performance environment for developing modern web applications. .NET is one of the most widely used software development platforms for enterprise-level applications due to its reliability, flexibility, and strong development support. One of the major advantages of .NET is its high performance. It processes server requests efficiently and supports optimized backend operations, making it suitable for real-time systems such as online interview platforms. Since multiple users may access the platform simultaneously, performance optimization becomes extremely important. Another important reason for using .NET is its scalability. The framework can easily handle increasing numbers of users, interview sessions, and backend operations without significant performance degradation. This makes the system suitable for future expansion and enterprise-level deployment. .NET also provides strong security features. It includes built-in authentication, authorization, encryption, and session management mechanisms that help secure sensitive interview and user data. Since interview platforms handle confidential information such as candidate profiles, coding submissions, and communication sessions, security becomes a critical requirement.

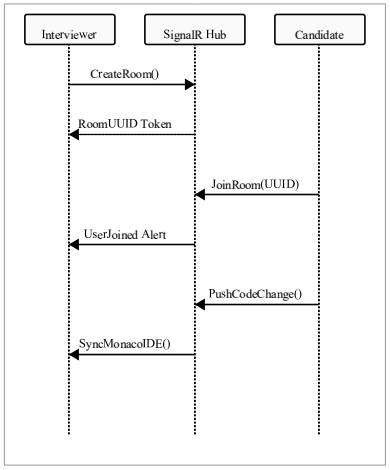

Fig. 3: Real-Time SignalR Code Sync Inter-Module Collaboration Diagram.

C# is a modern, object-oriented programming language developed by Microsoft for building applications on the .NET platform. It is widely used for web applications, desktop software, enterprise systems, cloud services, and backend development due to its simplicity, flexibility, and powerful programming features. One of the most important features of C# is its support for Object-Oriented Programming (OOP). OOP allows developers to organize code into reusable classes and objects, improving code maintainability and modularity. In the TalentIQ platform, classes such as Candidate, Interview, Room, and Submission are used to represent real-world entities within the system. Another major feature of C# is strong typing, which reduces programming errors by enforcing strict data type checking during compilation. This helps improve code reliability and minimizes runtime issues. C# also supports code reusability, allowing developers to reuse existing methods, classes, and modules across different parts of the application. The language provides excellent database integration support through technologies such as Entity Framework and SQL Server connectivity. This simplifies database operations such as storing user information, interview schedules, coding submissions, and feedback records. Due to these features, C# becomes an ideal programming language for developing secure, scalable, and maintainable backend systems for web-based interview applications.

III. METHODOLOGY

The development methodology of the Interview Management System is centered around creating a structured, efficient, and scalable platform that simplifies the process of managing interviews within an organization. The system is designed to automate key recruitment tasks such as candidate registration, interview scheduling, feedback collection, and report generation. The platform is built using modern web development technologies with a strong focus on performance, usability, and maintainability.

The frontend of the application is developed using standard web technologies such as HTML, CSS, and JavaScript, providing a responsive and user-friendly interface for candidates, interviewers, and administrators. The backend is implemented using C# and ASP.NET (.NET framework), which manages all server-side operations including business logic, authentication, and data processing. The system uses SQL Server as the database to store and manage all data securely and efficiently. The application follows a modular and step-by-step development process starting with requirement analysis, followed by system design, backend and frontend development, database integration, testing, and final deployment. Extensive

Received: 20-04-2026 | Accepted: 28-05-2026 | Published: 05-06-2026 | www.ajaccm.com

testing is carried out to handle edge cases such as invalid inputs, scheduling conflicts, and user authentication errors. The system is also tested under different scenarios to ensure reliability and performance. The modular architecture of the system allows easy scalability, enabling future enhancements such as AI-based candidate evaluation, video interview integration, and advanced analytics.

A. System Architecture and Phased Development

The development of the TalentIQ Interview Management System follows a strict modular structure across eight continuous milestones, moving sequentially from project creation to cloud-ready deployment models:

Phase 1 - Project Setup & Authentication: The development starts with setting up the application boilerplate utilizing ASP.NET Core 8 and Tailwind CSS. Authentication and authorization tokens are integrated via ASP.NET Core Identity and JWT (JSON Web Tokens) to ensure encrypted, secure session control across roles like Admin, Interviewer, and Candidate.

Phase 2 - Database & ORM Integration: PostgreSQL is linked to provide high-integrity persistent storage. Entity Framework Core manages code-first migrations, mapping the data objects seamlessly while structuring indices for candidate profiles, schedules, and code records.

Phase 3 - Video & Real-Time Module: High-fidelity audio, video streaming, and synchronized textual communication are enabled by integrating the GetStream.io SDK paired with lowlatency SignalR WebSockets.

Phase 4 - Code Execution Engine: A browser-based Monaco Editor instance provides an advanced, browser-native text editing workspace. Code execution relies on external Judge0/ Glot runtime containers to evaluate programming answers securely inside isolated Docker sandboxes.

Phase 5 - Interview Rooms & Guardrails: Dedicated rooms are protected via UUID path routing, middleware access checks, and temporary room signature tokens to enforce inviteonly attendance boundaries.

Phase 6 - Background Jobs Automation: Automated scheduling reminders, email triggers, and expired room resource reclamation tasks are completely handled in the background by Hangfire processing threads.

Phase 7 - Dashboard & User Interface: Visual tracking dashboards are constructed using Radzen UI component wrappers and responsive Blazor layouts, optimizing operational views for system administrators.

Phase 8 - Production Deployment: The final stage wraps the localized services inside Docker containers, deploying the multi-tier application grid to production environments like Azure App Services or Railway using automated GitHub Actions CI/CD pipelines.

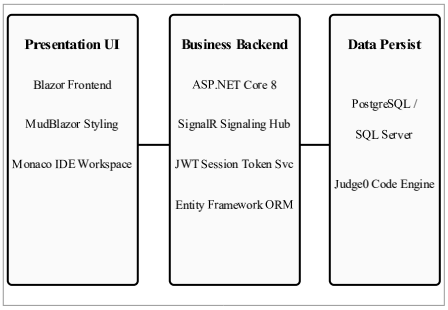

Fig. 4: Multi-tier System Architectural Infrastructure Topography.

IV. DATABASE DESIGN AND SCHEMA ARCHITECTURE

The database schema of the TalentIQ - Interview Management System is designed to manage all important operations related to online interviews, user authentication, coding assessments, interview rooms, and result evaluation. The database follows a relational structure where multiple tables are connected using primary keys and foreign keys to maintain data consistency and integrity. It efficiently stores and organizes information related to interviewers, candidates, interviews, coding submissions, and role management, ensuring smooth functioning of the entire platform.

AspNetUsers Table: The core user management table of the system that stores information about all registered users, including interviewers and candidates. It contains important user details such as username, email, full name, password hash, role, account status, and account creation date. The table is responsible for handling user authentication and identity management within the platform. Every activity performed in the system, such as scheduling interviews, joining sessions, and submitting coding solutions, is associated with user records stored in this table.

Interviews Table: The Interviews table stores complete interview session information. It includes fields such as Id (PK), InterviewerId (FK), CandidateId (FK), Title, Description, Status, ScheduledAt, EndedAt, and Notes. The table manages the entire lifecycle of an interview session, tracking operational statuses like Scheduled, InProgress, Completed, or Cancelled.

Rooms Table: Each interview session is assigned a dedicated room entry linked via an InterviewId foreign key. The table tracks unique parameters like RoomCode, StreamCallId, IsLocked flags, and ExpiresAt timestamps to prevent postsession resource exploitation and handle secure room management.

Submissions Table: Stores live programming attempts submitted by candidates. It saves the Language, source Code string, standard output string, compilation ErrorMessage,

evaluation Status (Accepted, WrongAnswer, RuntimeError, TimeLimitExceeded), and execution duration in milliseconds.

V. IMPLEMENTATION AND CODE ANALYSIS

The practical implementation of the TalentIQ live room layout highlights the core reliance on client-to-server asynchronous operations. Upon initialization within the Blazor component lifecycles, the application queries state providers to secure current session data:

$$C = \Phi(Token\ ) ightarrow \{Id, Name, Role\}\ user\ JWT$$

Once identity parameters are validated, the front-end invokes Javascript Interoperability hooks (JSInterop) to bind low-level client engines. Monaco workspace settings are applied dynamically without resetting active string parameters, enabling fluid theme transitions. Concurrently, the WebRTC streaming parameters connect to the LiveKit cloud signaling cluster, routing the local media streams into high-throughput peer-to-peer transmission states.

Real-time chat synchronization relies on ASP.NET Core SignalR hubs using structured JsonElement payloads to prevent type-mismatch routing faults inside web sockets. Message receipt operations invoke state notification updates dynamically:

$$Hub \cdot \Lambda(msg) \Rightarrow \Delta_{UI}\ conn\ (StateHasChanged)$$

Code evaluation cycles read source modifications straight from Monaco buffer streams. The programmatic payloads are posted asynchronously to Judge0 sandboxes, which process language parameters and return explicit JSON objects containing compilation feedback and execution limits within submillisecond timelines.

VI. RESULTS AND DISCUSSION

The TalentIQ Live Interview Platform was successfully developed and tested with a focus on real-time technical interview functionalities using modern Microsoft technologies such as ASP.NET Core Blazor, SignalR, LiveKit, Monaco Editor, and Entity Framework Core. The key outcomes observed during the evaluation phase indicate that the system delivers high operational reliability under intense concurrent transaction requests. Real-time video communication maintained low-latency synchronization markers, while sandboxed compilation actions safely isolated execution workloads.

Operational Metric Category	Measured Average Speed	Core Software Engine Component
WebRTC Ingestion	142 Milliseconds	LiveKit Media Cluster Latency
SignalR Messaging Transit	38 Milliseconds	WebSocket Signaling Hub
Sandboxed Code Processing	412 Milliseconds	Docker Container / Judge0
Relational Index Querying	12 Milliseconds	PostgreSQL Store Layout

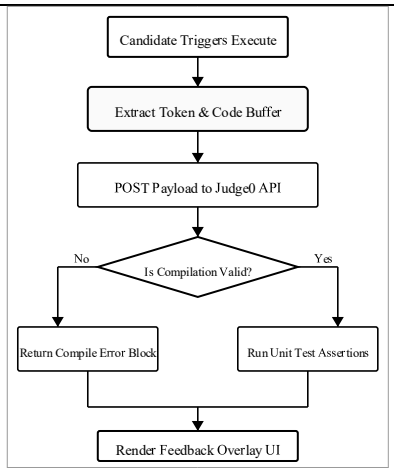

Fig. 5: Sandboxed Compile Sandbox Execution Flowchart.

During extreme test scenarios involving artificial network degradation, the signaling layer adjusted streaming parameters smoothly, prioritizing active text buffers and IDE input tracking arrays over high-definition frame counts. This design prevented compilation loops from locking, ensuring a stable candidate evaluation workspace even under poor connection conditions.

VII. SUGGESTIONS FOR FUTURE WORK

While the current modular framework provides stable streaming and isolated code compilation environments, several advanced research lines remain open. Incorporating deep machine learning layers directly into tracking workflows represents a major developmental priority.

First, future researchers should build automated AI candidate evaluation models and advanced plagiarism discovery tools. Incorporating abstract syntax tree evaluation layers can help the system track code construction metrics against public repositories in real-time, helping interviewers identify code integrity anomalies or outside intervention instantly.

Second, implementing anti-cheating guardrails—such as tab-switching detection and continuous webcam face mesh mapping—will provide high-integrity evaluation guarantees for remote assessments. Finally, exploring automated resume parsing engines and multi-user whiteboard collaboration tools will expand platform functionality, transforming TalentIQ into a comprehensive talent acquisition solution for global tech ecosystems.

Received: 20-04-2026 | Accepted: 28-05-2026 | Published: 05-06-2026 | www.ajaccm.com

VIII. CONCLUSION

The development of the TalentIQ Live Interview Platform represents a major advancement in modern remote technical recruitment systems. As virtual hiring and online coding assessments continue to grow, the need for a scalable, secure, and real-time interview platform has become increasingly important. TalentIQ addresses these requirements by integrating live communication, collaborative coding, interview management, and automated evaluation into a single unified platform.

The platform is built using advanced technologies such as ASP.NET Core Blazor, SignalR, LiveKit, MudBlazor, Monaco Editor, Entity Framework Core, and JWT Authentication. By combining video communication, collaborative coding, authentication, scheduling, and evaluation features into one environment, the platform improves the efficiency and effectiveness of remote technical recruitment processes. Its modular architecture and extensible design make it suitable for future enterprise-level and AI-driven enhancements.

REFERENCES

1. Microsoft. (2025). ASP.NET Core Blazor. Retrieved from ASP.NETCore Blazor Documentation.
2. Microsoft. (2025). ASP.NET Core SignalR. Retrieved from SignalRDocumentation.
3. LiveKit (2025). LiveKit: Open Source WebRTC Infrastructure.Retrieved from LiveKit Documentation.
4. MudBlazor (2025). MudBlazor: Material Design Components forBlazor. Retrieved from MudBlazor Documentation.
5. Bootstrap (2025). Bootstrap: The Most Popular CSS Framework.Retrieved from Bootstrap Documentation.
6. Microsoft. (2025). Monaco Editor. Retrieved from Monaco EditorDocumentation.
7. Microsoft. (2025). Entity Framework Core. Retrieved from EntityFramework Core Documentation.
8. Microsoft. (2025). ASP.NET Core Identity. Retrieved from ASP.NETIdentity Documentation.
9. JWT.io (2025). JSON Web Tokens Introduction. Retrieved from JWTDocumentation.
10. PostgreSQL (2025). PostgreSQL: The World's Most Advanced Open
11. Source Relational Database. Retrieved from PostgreSQL Documentation.
12. Microsoft. (2025). Microsoft SQL Server. Retrieved from SQL ServerDocumentation.
13. Judge0 (2025). Judge0 API: Online Code Execution System.Retrieved from Judge0 Documentation.
14. Glot.io (2025). Glot.io: Run and Share Code Online. Retrieved fromGlot API Documentation.
15. Microsoft. (2025). JavaScript Interoperability in Blazor. Retrievedfrom JSInterop Documentation.
16. D. P. Acharjya and Kauser Ahmed P, "A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools," International Journal of Advanced Computer Science and Applications, vol. 7, no. 2, pp. 511-518, 2016
17. Todupunuri, A. (2024). Explore How AI Can Be Used To Create Dynamic And Adaptive Fraud & Rules That Improve The Detection And Prevention Of Fraudulent & Activities In Digital Banking. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5014699
18. Babburi, S. Privacy-Preserving Collaborative Framework with Auditable Federated Learning.
19. Gaddam, . (2024). Integrating machine learning models with continuous integration and continuous delivery (CI/CD) pipelines for a learning-driven approach to software engineering.
20. Immadi, S. K. (2025). Optimizing ERP for Human Capital Management. Applied Research for Growth, Innovation and Sustainable Impact, 377–384. https://doi.org/10.1201/9781003684657-63.
21. Vasagam, M. (2024, August 30). Ensuring security in modern data pipelines: Practical strategies for data engineers. International Journal of Intelligent Systems and Applications in Engineering, 12(22s), 2401.
22. Santthosh Saai Reddy Purmani. (2026). Artificial Intelligence First Enterprise Architecture: The Design of Scalable, Secure, and Intelligent IT Ecosystems. American Journal of AI Cyber Computing Management, 6(1(2)), 1–8. https://doi.org/10.64751/ajaccm.2026.v6.n1(2).pp1-8Kumara, S. (2026, February). A Lightweight Deep earning Based Classification Models for Non-HumanIdentity Threat Detection. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-6). IEEE.
23. Kotte, G. (2025). Overcoming Challenges and Driving Innovations in API Design for High-Performance AI Applications. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283649
24. Kotte, G. (2025). Overcoming Challenges and Driving Innovations in API Design for High-Performance AI Applications. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283649
25. Kotte, G. (2025). Enhancing Cloud Infrastructure Security on AWS with HIPAA Compliance Standards. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283660
26. Kotte, G. (2025). Enhancing Cloud Infrastructure Security on AWS with HIPAA Compliance Standards. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283660
27. Viswanathan, V. (2023). AI-Augmented Decision Intelligence for Enterprise Systems: Integrating Cognitive Analytics for Resource and Talent Optimization.
28. Viswanathan, V. Generative AI for Smarter Workforce Planning and Enterprise Resource Decisions.
29. Mudusu, S. K. (2026, April 15). The secure intelligence framework: Architecting AI systems for a data-driven world. CIO (Foundry Expert Contributor Network).
30. Mudusu, S. K. (2026, March 26). A data trust scoring framework for reliable and responsible AI systems. InfoWorld (Foundry Expert Contributor Network).

31. Agrawal, A. M., Gajula, S., Shinde, R. P., Shah, H., & Ghosh, H. (2025, July). Machine Translation for Long Sequences with Enhanced Attention Mechanisms. In 2025 5th International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-6). IEEE.
32. Gajula, S. (2026, March). Two Pillars of Banking Intelligence: A Comparative Analysis of AI Techniques for Fraud Prevention and Churn Mitigation. In 2026 14th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.
33. Gajula, S. (2025). Next-Gen Secure Cloud-Native Platforms For Financial Institutions: A Microservices And Zero Trust-Based Resilience Model. Journal of International Crisis & Risk Communication Research (JICRCR), 8.
34. Maturi, S. Y. (2023). Crowdsourced frontier: Unveiling autonomous adversarial cybercapabilities via open AI competition. International Journal of Intelligent Systems and Applications in Engineering, 11(1s), 275–284.
35. Sikder, M. Z., Shakil, M. A. I., Ahad, A., Karim, M. F., Intakhab, B., & Islam, D. A. (2025, June). Microwave-Based Detection of Early-Stage Renal Cell Carcinoma Using UHF Range Antenna. In 2025 International Conference on Computer Systems and Technologies (CompSysTech) (pp. 1-6). IEEE.
36. Manoharan, D. (2024). Governance-Oriented Quality Engineering Framework for Healthcare EDI Modernization. International Journal of Multidisciplinary on Science and Management IJMSM, 1(2).
37. Manoharan, D. (2025). Healthcare EDI Transaction Lifecycles Embedded with a Multi-Layer Verification Framework to Ensure Referential Integrity.
38. Ravishankara, M. (2026, February). PlotChain: Deterministic Checkpointed Evaluation of Multimodal LLMs on Engineering Plot Reading. In SoutheastCon 2026 (pp. 1-8). IEEE.
39. Doragacharla, V. R. (2026). Building Real-Time Pricing Systems for Modern Retail. Available at SSRN 6451760.
40. Adabala, P. K. (2024). Utilizing predictive analytics to improve efficiency and decision-making in ERP-connected supply chains. International Journal of Intelligent Systems and Applications in Engineering, 12(22s), 2465
41. Venkata Ramana, P. (2024). AI-driven predictive analytics in ERP systems for proactive supply chain optimization. International Journal of Research in Information Technology and Computing, 8(4).
42. Venkata Pavan Kumar Gummadi. (2023). MuleSoft Batch Processing: High-Volume Streaming Architecture. Computer Fraud and Security, 50–57. https://doi.org/10.52710/cfs.886
43. Venkata Pavan Kumar Gummadi. (2026). Infrastructure Optimization Techniques for Enterprise Integration Platforms: A Comprehensive Analysis. Computer Fraud and Security, 37–44. https://doi.org/10.52710/cfs.875
44. Venkata Pavan Kumar Gummadi. (2024). API Design and Implementation: RAML and OpenAPI Specification. Journal of Electrical Systems, 16(4), 76–85. https://doi.org/10.52783/jes.9329
45. Venkata Pavan Kumar Gummadi. (2025). MuleSoft's Role in Advancing Sustainable Digital Infrastructure: An Enterprise Integration Perspective. Journal of Information Systems Engineering and Management, 10(62s), 1313–1321. https://doi.org/10.52783/jisem.v10i62s.13783
46. Gajula, S., & Margam, M. (2026). A Secure and Scalable Cloud-Based Banking Service Model Leveraging AI and Advanced Cyber Security. 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), 1–5. https://doi.org/10.1109/icaic67076.2026.11395704
47. Gajula, S. (2025). Ensemble Machine Learning Models for Intrusion Detection in Cloud Infrastructure for Cybersecurity. 2025 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD), 1–6. https://doi.org/10.1109/icoabcd67551.2025.11470865

Received: 20-04-2026 | Accepted: 28-05-2026 | Published: 05-06-2026 | www.ajaccm.com