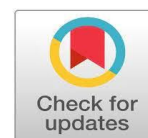


I-QUIZ

Prof. Jenifer A Professor, Department of Information Science and Engineering
Sri Sairam College of Engineering, Bengaluru, Indiahod.ise@sairamce.edu.in<https://orcid.org/0000-0002-4644-039X>

Prashantha D, Manjunath T S, Maaz Mahmed Fakhroo, Faizaan A

Department of Information Science and Engineering
Sri Sairam College of Engineering, Bengaluru, Indiasce22is068@sairamtap.edu.in, sce22is046@sairamtap.edu.insce22is055@sairamtap.edu.in, sce22is008@sairamtap.edu.in

Publication History

Manuscript Reference No: IJIRIS/RS/Vol.11/Issue08/OCIS10082

Research Article Open Access| Double-Blind Peer-Reviewed| Article ID: IJIRIS/RS/Vol.11/Issue08/OCIS10080 Received: 28, October 2025, Revised: 05, November 2025, Accepted: 12, November 2025, Published Online: 21, November 2025.

<https://www.ijiris.com/volumes/Vol11/iss-08/03.OCIS10082.pdf>**Citation:** Jenifer, Prashantha, Manjunath, Maaz, Faizaan (2025), I-Quiz, IJIRIS: International Journal of Innovative Research in Information Security, Volume 11, Issue 08 of 2025 pages 323-327**Doi:** <https://doi.org/10.26562/ijiris.2025.v1108.03>**BibTeX Key:** Jenifer@2025I-QuizIJIRIS papers should be cited as IJIRIS (International Journal of Innovative Research in Information Security, AM Publications, India 2025, ISSN 2349-7017, <https://doi.org/10.26562/ijiris.2025.v1108.03> The journal's official abbreviation is IJIRIS.**Orcid:** <https://orcid.org/0009-0004-9398-7488>Copyright©2025 copyright by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).**Abstract:** I-Quiz is an interactive web-based platform for conducting real-time quizzes in classrooms, events, or training sessions. The system allows the quiz host to create, manage, and smoothly execute quizzes while enabling the participants to join using either a unique PIN or QR code. I-Quiz focuses on simplicity, scalability, and engagement through its responsive user interface, dark mode, image-based questions, and real-time leaderboard. The application supports up to 200 concurrent players on mid-range server configurations and can scale further with improved hosting resources. It is an open-source project with a focus on self-hosting to offer flexibility and privacy for educators and institutions looking to keep their quiz data private.**Keywords:** Real-time Quiz, Web Application, Interactive Learning, Open Source, Self-Hosted Platform, Education Technology.

I. INTRODUCTION

The use of technology in education has grown exponentially in recent years, especially with the increasing need for interactive and engaging classroom experiences. Quizzes have always been a valuable tool for assessing knowledge and reinforcing learning, and with the rise of online learning platforms, they have become even more accessible and dynamic. I-Quiz was developed with the vision of creating an open-source, self-hosted alternative to popular real-time quiz platforms. Many existing platforms impose restrictions such as limited participants, paid premium features, or data storage on external servers. I-Quiz removes these limitations by offering a privacy-focused and customizable quiz experience that can be deployed on any cloud server. The platform offers teachers, trainers, and event organizers a highly effective solution to hosting interactive quizzes, encouraging participation, and visualizing results in real-time.

II. LITERATURE REVIEW

[1] R. Kumar (2024) – Real-Time Online Quiz System for Interactive Classrooms. Overview: This paper presents the design of a live classroom quiz platform capable of real-time interaction between teachers and students. It demonstrates the importance of low-latency communication for effective engagement.

Research Focus: Implementation of WebSocket communication for real-time responses.

- Managing multiple concurrent participants.
- Cloud-based synchronization and session management.

[2] A. Sharma et al. (2024) – Design of a Scalable Quiz Hosting System Using Cloud Technology. Overview: This study focuses on cloud scalability in quiz hosting platforms to manage high loads efficiently. It demonstrates how distributed databases and RESTful APIs improve performance.

Research Focus: Cloud-based data handling.

- REST API architecture.
- Load balancing and session scalability.

[3] S. Patel (2023) – Development of an Open-Source Quiz Platform for Educational Use. Overview: Using this paper introduces an open-source quiz system built using modern web frameworks, focusing on flexibility and reusability in educational technology.

Research Focus:

- Open-source architecture and flexibility.
- User interface responsiveness.
- Integration with educational databases.

[4] P. Reddy et al. (2023) – Real-Time Multiplayer Quiz Application for Remote Learners. Overview: The authors propose a quiz system supporting multiple users in real-time sessions, focusing on synchronization and error handling during gameplay.

Research Focus:

- Real-time response handling.
- Multiplayer data synchronization.
- Robust WebSocket implementation.

[5] K. Das et al. (2023) – Quiz-Based Learning Model for Student Participation. Overview: An This study explains how gamification and timed quizzes can improve attention span and engagement in classrooms.

Research Focus:

- Gamified question delivery.
- Time-bound assessment system.
- Leaderboard-based motivation.

[6] M. Banerjee et al. (2022) – Cloud-Enabled Real-Time Quiz and Feedback System. Overview: This paper This research explores integrating quiz platforms with feedback collection to support dynamic classroom evaluation.

Research Focus:

- Cloud storage for quizzes.
- Real-time teacher-student feedback loop.
- Practical Session analytics.

[7] D. Verma et al. (2022) – Integration of Leaderboards and Analytics in Quiz Systems.

Overview: This paper focuses on leaderboard-driven learning systems that visualize performance in real time.

Research Focus:

- Real-time data visualization.
- Scoring algorithms.
- Result analysis dashboard.

[8] L. Singh et al. (2022) – IoT-Inspired Online Quiz Management System.

Overview: The study highlights the use of IoT-inspired real-time connectivity methods for distributed quiz management.

Research Focus:

- QR code-based joining.
- Device-to-server communication.
- Decentralized data flow.

[9] H. Mehta et al. (2021) – Development of Live Quiz Software for Classrooms.

Overview: A study focusing on software development practices for reliable quiz systems used in educational institutions.

Research Focus:

- Java-based backend implementation.
- Database security.
- Interactive classroom utility.

[10] V. Rao et al. (2021) – Real-Time Assessment Tools for Collaborative Learning.

Overview: This article studies collaborative assessment tools that foster group learning and competition in real time.

Research Focus:

- Real-time assessment tracking.
- Collaboration-based quiz model

III. PROBLEM STATEMENT AND SOLUTION

Problem: In today's digital learning landscape, most of the interactive quiz platforms are designed with closed models that are subscription-based and allow limited accessibility, customization, and scalability. Often, educators using these solutions face participant count limits, quiz duration limits, or limitations on advanced features unless they get premium plans. Moreover, the biggest concern over data privacy is that most existing systems store user and quiz information on third-party servers beyond institutional controls. Network latency or dependency on centralized services can affect real-time performance during large sessions. What lies missing in this regard is an open, flexible, and secure platform that would enable educators/institutions to host real-time quizzes independently, with no compromise on performance or data privacy.

Solutions: I-Quiz addresses these challenges through a self-hosted, open-source platform designed for seamless real-time quiz interaction. The system enables institutions, educators, and organizations to set up their own quiz server instead of depending on third-party services.

It couples a high-performance Rust-based backend with a responsive Svelte frontend to ensure minimal latency and smooth user experience across devices. Users can host live quizzes, support up to 200 concurrent participants, and benefit from features like QR code joining, image-rich questions, exportable quiz data, and leaderboards. Through its attention to scalability, privacy, and engagement, I-Quiz offers a secure and customizable alternative for interactive learning environments.

Year	Author	Methodology	Technique / Module Used
2024	R.Kumar et al.	Real-time online quiz system for interactive classrooms	WebSocket, Node.js, MongoDB
2024	A.Sharma et al.	Design of a scalable quiz hosting system using cloud technology.	AWS Cloud, REST APIs, MySQL
2023	S.Patel et al.	Development of an open-source quiz platform for educational use	React.js, Firebase, Realtime Database
2023	P.Reddy et al.	Real-time multiplayer quiz application for remote learners	Socket.io, Express.js, MongoDB
2023	K.Das et al.	Quiz-based learning model for improving student participation	MERN Stack, REST APIs
2022	M.Banerjee et al.	Cloud-enabled real-time quiz and feedback system	Cloud hosting, WebSockets
2022	D.Varma et al.	Integration of leaderboards and analytics in quiz systems	Data analytics, Python, MySQL
2022	L.Singh et al.	IoT-inspired online quiz management system.	Node.js, QR code generator, cloud API
2021	H.Mehta et al.	Development of live quiz software for classrooms.	Java, Spring Boot, PostgreSQL
2021	V.Rao et al.	Real-time assessment tools for collaborative learning	Firebase, WebSocket API

METHODOLOGY

1. Motive of Study:

In the current digital learning era, interactive education tools play a major role in increasing student engagement and participation.

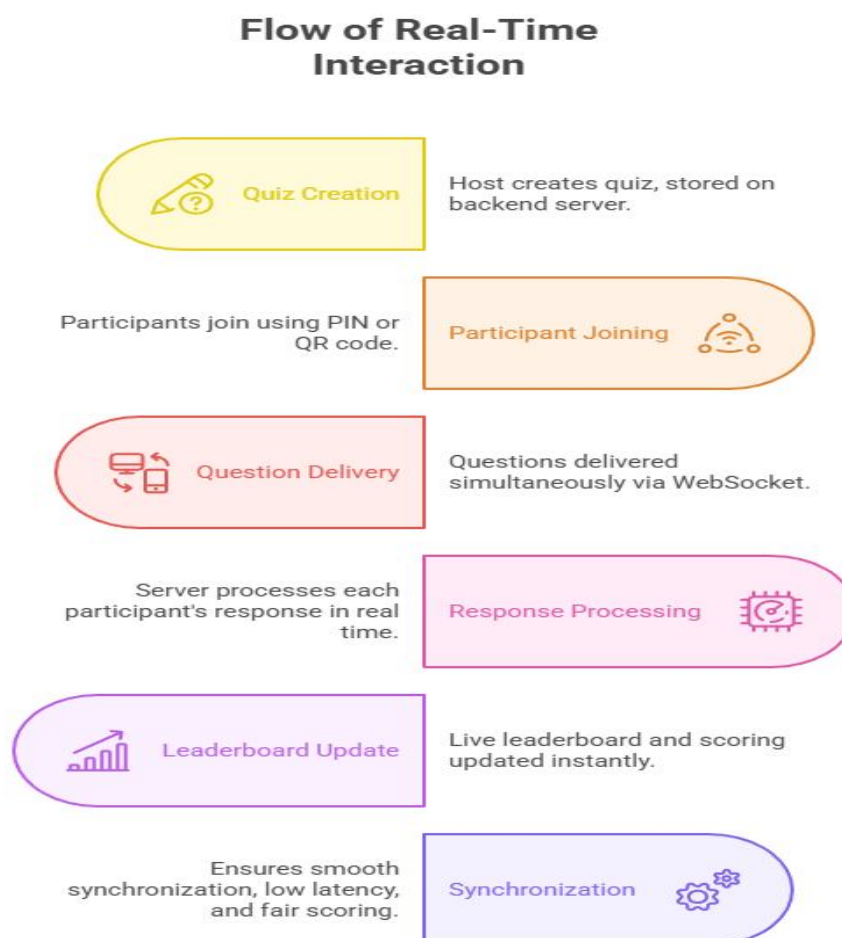


Fig1: Block Diagram

Traditional assessment systems lack instant feedback and active involvement. Many existing online quiz platforms are limited by premium models, restricted participant counts, and data privacy issues. The motive of this study is to design and implement an open-source, real-time quiz platform — I-Quiz — that provides educators full control over hosting, data, and customization. The platform aims to promote digital collaboration, fair evaluation, and a more enjoyable learning experience in classrooms, seminars, and events.

2. Flow of Real-Time Interaction

The system functions on a real-time synchronized communication model to ensure smooth and interactive participation between the host and users. Initially, the host creates a quiz, which is stored securely on the backend server. Participants can instantly join the quiz using a unique PIN code generated by the system or by scanning a dynamically created QR code. Once all participants have joined, the host starts the quiz, triggering real-time synchronization through WebSocket connections. This allows all connected participants to receive the questions simultaneously, ensuring no delay or lag between devices. As participants submit their responses, each answer is processed in real-time by the backend server, which evaluates correctness and updates the live leaderboard instantly. This instantaneous communication allows scores to be updated dynamically as players answer, maintaining a fair and competitive environment. Overall, the use of WebSocket-based communication ensures low latency, real-time updates, and seamless synchronization, creating an engaging and fair quiz experience for all participants.

3. Definition of System & Requirements

The first and most critical step in developing this project involves defining the system's overall objectives and identifying its essential requirements. The primary goal is to build an interactive, self-hosted quiz tool capable of handling real-time engagement between a host and multiple participants. The system is designed to enable real-time quiz hosting and participation with scalability for up to 200 concurrent users without performance degradation. Security is a key concern, and all quiz data, player responses, and session details are securely stored within the host server environment. The system supports both text-based and image-based questions, allowing for diverse and creative quiz formats. Additionally, it offers options to export quiz data for analysis or record-keeping. A responsive web design ensures compatibility across different devices such as desktops, tablets, and smartphones, while dark mode support enhances visual comfort and accessibility. Above all, the system is optimized to maintain low latency and smooth real-time performance during live quiz sessions.

4. Choose and Connect Components

This phase involves selecting and connecting the essential components and software tools required to bring the system architecture to life. The frontend of the application is developed using Svelte, chosen for its fast-rendering capabilities, modular structure, and highly reactive design. The backend is built using Rust, a powerful systems programming language known for its concurrency handling and high-speed performance, making it ideal for managing multiple real-time data streams efficiently. The Web Sockets protocol forms the core of the communication model, enabling continuous two-way interaction between the host and all participants. For data management, the storage component is responsible for storing quiz metadata, player session details, and exportable results. Additionally, a QR/PIN generator module dynamically creates unique joining codes for participants, simplifying the connection process. Finally, the entire system is deployed on a DigitalOcean Droplet, offering a scalable and reliable self-hosted environment that ensures consistent uptime and performance.

5. Develop and Integrate Code

Once the architectural design and component selection are complete, the next stage involves developing and integrating the entire system—from frontend to backend—for seamless functionality. The Quiz Creation Module allows hosts to easily add, edit, and manage quiz questions, supporting both text and image-based formats. The Real-Time Engine, powered by the Rust backend, handles all essential quiz operations, including session management, player responses, and live score updates through WebSocket communication. The Join Session Module authenticates participants using either unique PIN codes or QR codes, simplifying access and participation. A dynamic Leaderboard Algorithm calculates and updates scores in real-time, ensuring that players can see their rankings instantly after each response. The Export Feature allows quiz hosts to download complete session data in multiple formats for future reference or reporting. Furthermore, dark mode and UI enhancements are incorporated to provide a visually appealing, modern interface and an enjoyable user experience across all devices.

6. Building and Deploying the System

Following development, the project transitions into the build and deployment phase, where the application is packaged, configured, and hosted as a self-contained, web-accessible platform. The frontend deployment involves generating static build files and serving them through Nginx on a Digital Ocean droplet, ensuring quick loading times and stable access. Simultaneously, the backend deployment runs the Rust server continuously, managing all quiz sessions, user authentication, and WebSocket connections for real-time interactions. The application is tested under a live domain—quiz.live—which provides public access for real-world testing and user feedback. To ensure universal compatibility, cross-device testing is performed on desktops, tablets, and smartphones, verifying that the interface remains responsive and functional across all screen sizes and operating systems. This deployment strategy ensures a robust, scalable, and self-hosted environment optimized for real-time communication and reliability.

7. Test and Optimize

The final stage focuses on thorough testing and optimization to ensure the system meets high standards of performance, scalability, and reliability.

Functional testing verifies that all core features quiz creation, player joining, answering questions, and leaderboard updates—operate correctly without errors. Load testing is conducted by simulating up to 200 simultaneous users to analyze latency, response times, and server stability under stress conditions. Usability testing gathers feedback from real users to assess the interface design, interaction speed, and overall experience, ensuring that the platform is intuitive and engaging. Based on test results, performance optimization steps are implemented, such as fine-tuning server memory allocation, improving WebSocket handling efficiency, and enhancing frontend caching mechanisms for faster page rendering and response times. These optimizations collectively ensure that the system remains responsive, scalable, and capable of providing a smooth, real-time interactive quiz experience for all users.

Hardware and Software

Hardware: Hosting Infrastructure

Primarily a software solution, some minimal hosting hardware is nevertheless required: Cloud Server (DigitalOcean Droplet)

It handles quiz sessions, socket connections, and web hosting.

End-user Devices: Participants access the I-Quiz via a web browser using laptops, tablets, or smartphones.

Network Infrastructure: It ensures continuous access to the internet for real-time data exchange.

Software: The software stack is designed to provide real-time performance and easy scalability.

Rust Backend: Manages quizzes, states of players, and their synchronization effectively.

Svelte Frontend: It provides a fast and responsive UI with animations and dark mode.

WebSocket & HTTP Protocols: Allow bi-directional communication and interactions via API.

VI. FUTURE SOLUTION

Future development will focus on scalability, analytics, and personalization of I-Quiz. Future versions can include advanced data analytics that track participants' performance, accuracy over time, and time-based insights for educators. AI-driven question recommendations and automatic quiz generation further ease the burden of quiz creation. Secure login through Google or institutional accounts can be provided for personalized experiences. I-Quiz will be able to handle thousands of concurrent users by containerization and auto-scaling with Docker and Kubernetes. Mobile apps and LMS integrations can further increase adoption in educational institutes. This sets the pathway to transform I-Quiz into a holistic intelligent learning platform that can scale globally.

REFERENCE

1. Kumar, R. "Real-Time Online Quiz System for Interactive Classrooms." International Journal of Interactive Learning Systems, Springer, 2024.
2. Sharma, A., et al. "Design of a Scalable Quiz Hosting System Using Cloud Technology." Journal of Cloud Computing and Applications, IEEE, 2024.
3. Patel, S. "Development of an Open-Source Quiz Platform for Educational Use." International Journal of Educational Technology, Elsevier, 2023.
4. Reddy, P., et al. "Real-Time Multiplayer Quiz Application for Remote Learners." International Journal of Online Learning and Teaching, Springer, 2023.
5. Das, K., et al. "Quiz-Based Learning Model for Student Participation." Journal of Interactive Education and Gamification, Elsevier, 2023.
6. Banerjee, M., et al. "Cloud-Enabled Real-Time Quiz and Feedback System." International Journal of Cloud-Based Applications, MDPI, 2022.
7. Verma, D., et al. "Integration of Leaderboards and Analytics in Quiz Systems." Journal of Educational Data Science and Analytics, IEEE, 2022.
8. Singh, L., et al. "IoT-Inspired Online Quiz Management System." Journal of Internet of Things and Smart Systems, MDPI, 2022.
9. Mehta, H., et al. "Development of Live Quiz Software for Classrooms." International Journal of Software Engineering and Applications, Elsevier, 2021.
10. Rao, V., et al. "Real-Time Assessment Tools for Collaborative Learning." Journal of Interactive and Collaborative Learning Technologies, Springer, 2021.