Original Article
ISSN (Online): 2350-0530
ISSN (Print): 2394-3629

International Journal of Research - GRANTHAALAYAH
December 2024 12(12), 139–151

QUIZ MANAGEMENT SYSTEM: A WEB-BASED APPROACH TO ENHANCE FRONTEND DEVELOPMENT AWARENESS

Sagar Yadav [1], Sahil [1], Salim Mian [1], Vinisha jha [1], Priyanka [1]

[1] Department of Computer Science and Engineering, Echelon Institute of Technology, Faridabad, India

Received 27 October 2024
Accepted 28 November 2024
Published 31 December 2024

DOI
10.29121/granthaalayah.v12.i12.2024.6122

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

The Quiz Management System (QMS) is an interactive, web-based software application designed to promote and evaluate knowledge of frontend web development among its users through a structured quiz challenge. Users can log in securely, participate in quizzes, and instantly view their results upon submission. The system ensures seamless data handling by storing all quiz-related information in a centralized MySQL database, which supports fast retrieval and efficient data manipulation. Built using HTML for the frontend and PHP for server-side scripting, this distributed system allows multiple users to access and interact with the platform simultaneously, ensuring scalability and convenience.

The key objective of QMS is to raise awareness of essential web development concepts while providing a platform that encourages self-assessment and learning. Each quiz includes a timer mechanism that advances users to the next question upon timeout, automatically redirecting them to the scoreboard after the final question. Administrators are granted access to all quiz results, facilitating performance monitoring and data-driven decision-making.

The system offers several advantages over traditional manual methods. It improves data validity through built-in validation checks and enforces data integrity by restricting changes to primary fields. The user-friendly interface, efficient data management, and structured database design enable faster information processing and accurate performance evaluation. Furthermore, the automation of quiz assessment significantly reduces manual effort and minimizes the potential for human error.

Overall, the Quiz Management System streamlines knowledge assessment in web development, enhancing learning efficiency, accuracy, and transparency through the integration of modern web technologies.

1. INTRODUCTION

In today's digitally driven educational environment, traditional methods of assessment are rapidly being replaced with technologically advanced alternatives. The need for faster, more accurate, and efficient methods of testing knowledge has resulted in the widespread adoption of digital tools, one of the most important being the Quiz Management System (QMS). A QMS is a web-based solution that automates the entire process of conducting quizzes, from question generation to result evaluation and analysis. It provides a platform for learners to test their knowledge through interactive assessments and offers real-time feedback, which is a crucial aspect of modern pedagogy [1]. As the global education ecosystem moves toward

hybrid and online models, the implementation of such systems has become both relevant and necessary [2].

The idea behind the Quiz Management System is to promote a learning culture that is interactive, measurable, and scalable. In the case of this research, the focus of the QMS is specifically on Frontend Web Development. This is a subject area with increasing relevance, given the surge in demand for web developers who understand HTML, CSS, JavaScript, and other associated technologies [3]. The QMS is designed to serve as both a testing and learning tool, where users can register, take quizzes, receive feedback, and analyze their performance over time. It is tailored not only for students but also for hobbyists and early-career developers who want to evaluate their skill levels in frontend development [4].

The platform is built using HTML for structure, PHP for backend scripting, and MySQL as the database system. This combination ensures that the system remains lightweight, easy to deploy, and capable of handling real-time user requests. By integrating PHP and MySQL, the system can efficiently manage user sessions, store results, and maintain quiz data with precision. Furthermore, being browser-based, the application can be accessed from any system with internet connectivity, enabling wide usability and promoting learning beyond geographical constraints [5]. The use of PHP and MySQL also supports the design of secure login systems, user session management, and data validation techniques to ensure data integrity and security [6].

One of the primary motivations behind developing this system is the need for self-paced learning and assessment. Traditional classroom tests are often limited by time constraints, location, and resources. The QMS overcomes these limitations by allowing users to take quizzes anytime and anywhere. In addition, the automated nature of the system ensures that results are generated instantly and without error, eliminating the potential for human mistakes in grading and data entry. This automation contributes to the transparency and efficiency of the assessment process, aligning with the objectives of modern education technologies [7].

Another key factor behind the system is the need for better data organization. Manual paper-based quizzes generate large amounts of data that are often hard to track, analyze, and utilize for further learning. The QMS records every user interaction in a structured database, which can later be queried for insights on performance trends, commonly missed questions, and quiz completion times. These insights help both the learners and the educators in identifying knowledge gaps and designing better content. The data-driven nature of the QMS not only enhances learning outcomes but also contributes to research in instructional design and pedagogy [8]. As education systems globally move towards data-driven decision-making, tools like the QMS serve as key enablers of effective learning analytics [9].

The QMS also introduces a time-controlled environment where each question must be answered within a specified time limit. This simulation of real-world constraints adds to the effectiveness of the system, preparing users for certification tests or job screenings where time management is crucial [10]. If a user fails to answer within the allotted time, the system automatically moves them to the next question or ends the session as appropriate. This not only keeps the process efficient but also cultivates discipline and speed in problem-solving among learners, contributing to their real-world preparedness [11].

Furthermore, the platform incorporates administrative functionalities that allow educators or administrators to manage users, questions, and results. They can add, modify, or remove quiz questions, monitor user activities, and generate reports based on different performance metrics. These features are particularly useful in

academic institutions or training centers where multiple users need to be evaluated periodically and consistently. Through its multi-user capability, the QMS functions as a distributed system, ensuring simultaneous access without compromising on performance [12].

Despite its many advantages, the system does face certain limitations. Being an online platform, it requires continuous internet access to function optimally. Users with poor connectivity may face difficulties in completing quizzes. Also, while the system is designed to be user-friendly, individuals unfamiliar with digital platforms might initially struggle to navigate it. Nevertheless, with minimal guidance and user manuals, these hurdles can be easily overcome. The system's flexible design and modular structure also allow for future upgrades, such as adding support for mobile devices, integrating multimedia-based questions, or implementing adaptive testing algorithms based on user performance [13].

Importantly, the QMS also addresses issues of data integrity and security. Since user performance data is stored in a centralized database, it becomes crucial to implement secure login systems and user access control to prevent unauthorized access and data manipulation. The use of PHP allows developers to implement session management and data validation techniques to ensure that the information remains secure and reliable. Moreover, MySQL's robust structure supports data backup and recovery mechanisms, providing a safeguard against accidental data loss or corruption [14].

In terms of usability, the system is designed to be intuitive and responsive. The user interface is clean, navigable, and provides clear instructions to guide users through the quiz process. Each question is displayed with multiple-choice options, and the timer is prominently shown to ensure that users are aware of their remaining time. Upon submission of the quiz, the results are calculated and presented with appropriate feedback, including correct answers and explanations if configured. This feedback-oriented approach transforms the quiz from a mere testing tool into a learning experience [15].

Moreover, the QMS supports progress tracking, enabling users to view their previous scores and monitor improvement over time. This feature not only motivates learners but also encourages them to focus on areas where they performed poorly. For instructors, this allows a deeper understanding of class-wide performance and helps in modifying curriculum or teaching strategies accordingly. As the system grows, machine learning algorithms can be integrated to personalize the quiz content, creating a customized learning path for each user [16].

In conclusion, the development of a Quiz Management System centered around frontend web development serves as a significant contribution to digital education tools. It combines automation, interactivity, and accessibility to create a system that is not only functional but also impactful in promoting technical knowledge. By addressing the needs of both learners and administrators, the system provides a comprehensive solution for knowledge evaluation. Through this research and implementation, the study aims to demonstrate the effectiveness, scalability, and educational value of such systems, contributing to the evolving narrative of technology-enabled learning [17].

2. LITERATURE REVIEW

2.1. INTRODUCTION TO QUIZ MANAGEMENT SYSTEMS

In recent years, the educational sector has embraced digital technologies to enhance learning and assessment processes. One of the primary advancements is

the Quiz Management System (QMS), a digital platform designed to automate the process of creating, managing, and evaluating quizzes. The shift from traditional paper-based quizzes to digital solutions is driven by the desire to increase efficiency, reduce administrative burdens, and provide real-time feedback to learners. The QMS not only streamlines quiz management but also allows for better data collection, performance tracking, and real-time evaluation, contributing to more personalized learning experiences [1].

A QMS serves various educational institutions, from schools and universities to online learning platforms, by offering a flexible, scalable, and accessible system for quizzes. It facilitates efficient quiz creation by allowing administrators to input multiple-choice questions, true/false questions, short-answer questions, and other types of assessment formats. These features make QMS systems an essential part of modern education, fostering real-time knowledge assessment in diverse learning environments [2]. Moreover, it ensures that both educators and learners benefit from immediate feedback, enhancing the learning process and enabling improvements in performance.

2.2. TECHNOLOGICAL FOUNDATIONS OF QMS

The underlying technology of a Quiz Management System plays a crucial role in its overall functionality, scalability, and security. Most modern QMS applications rely on a distributed architecture, wherein data is stored in centralized databases, typically using technologies like MySQL for managing quiz data and user results. By using a centralized database system, administrators can easily manage users' data, track performance, and generate reports, ensuring efficient information flow across all users [3]. The use of PHP for building the user interface ensures that the system is dynamic, allowing for the creation of interactive quizzes that can respond to user input in real-time [4].

Furthermore, the MySQL connection methodology allows for seamless database communication, enabling the secure and fast retrieval of quiz data. MySQL's ability to handle large datasets efficiently makes it a suitable choice for applications that need to store user data, quiz content, and results [5]. The system’s architecture is designed to maintain security by integrating protective mechanisms like role-based access control and data validation techniques. These features are essential to safeguarding user information, preventing unauthorized access, and ensuring that only the correct data is processed within the system [6].

2.3. SYSTEM FEATURES AND BENEFITS

The major benefit of implementing a Quiz Management System is its ability to automate quiz creation and evaluation. In traditional systems, quizzes are manually created, administered, and evaluated, which is time-consuming and prone to human error. A QMS, however, automates these tasks, reducing the risk of mistakes and saving valuable time for both educators and students [7]. Additionally, automation leads to more consistent grading and faster results, which increases the transparency and reliability of the testing process [8].

Another important feature of the QMS is its user-friendly interface. To ensure maximum accessibility, the interface is designed to be intuitive, enabling users, both administrators and participants, to navigate the system effortlessly. A simplified user interface contributes to enhanced usability, ensuring that users can quickly adapt to the platform, reducing the learning curve for new users [9]. For administrators, the interface allows for easy quiz creation, question categorization,

and result analysis, all of which can be managed from a centralized dashboard. Users, on the other hand, can take the quiz, view their results, and receive feedback without complex navigation steps.

The real-time evaluation and feedback system is another key advantage of the QMS. Upon quiz submission, users immediately receive their scores, along with detailed explanations of correct and incorrect answers. This timely feedback is critical for reinforcing learning, allowing users to review areas where they may have struggled. Studies have shown that real-time feedback significantly improves learning outcomes by allowing users to adjust their study strategies promptly [10]. This feature is particularly valuable in online education platforms where immediate interaction between students and instructors may not always be feasible.

2.4. SECURITY AND DATA PROTECTION IN QMS

The issue of data security is of paramount importance in the context of online learning and testing. A Quiz Management System often stores sensitive data, such as user credentials, quiz responses, and performance reports. Ensuring the privacy and security of this data is essential to maintain the integrity of the system and protect the personal information of users. As such, security measures like encrypted user authentication, session management, and role-based access control are implemented to safeguard against unauthorized access [11].

Role-based access control (RBAC) is particularly important, as it allows for a secure, granular distribution of privileges across different types of users. For instance, administrators have access to quiz creation tools, user management, and reporting functionalities, while regular users only have access to quiz-taking and result viewing features. This layered security approach ensures that only authorized individuals can modify quiz content, manage user data, or access sensitive reports [12]. Moreover, incorporating data validation techniques during quiz entry ensures that the information stored in the database is accurate and valid, preventing errors and maintaining the quality of the data over time.

Additionally, to further protect user data, the MySQL database system allows for secure, encrypted connections between the application and the database. It also supports data backup and recovery features to prevent data loss in case of system failures. Regular backups ensure that, even in the event of a disaster, no critical data is lost, preserving the integrity of the system and its contents [13].

2.5. CHALLENGES AND LIMITATIONS

While QMS platforms offer numerous advantages, they also come with certain limitations. One major challenge is system scalability. As the number of users increases, the system may encounter performance degradation, especially in cases where the quizzes are resource-intensive or the database grows exponentially. To address this, systems must be designed with scalability in mind, using cloud infrastructure and load-balancing techniques to ensure smooth performance under heavy usage [14]. The flexibility of the system also plays a significant role in overcoming this limitation, as it allows the application to be expanded and customized according to future needs.

Another challenge is ensuring accessibility for users who may have limited internet connectivity or lower-end devices. As QMS platforms are web-based, they require stable internet connections for optimal functionality. To mitigate this issue, the system should be optimized for low-bandwidth environments and should be

responsive across different device types, ensuring that users can access and take quizzes with minimal technical barriers [15].

The literature on Quiz Management Systems emphasizes their importance in the modern educational landscape, providing an efficient, scalable, and secure solution for automating quiz administration and evaluation. These systems not only simplify administrative tasks but also enhance the learning experience for users by offering real-time feedback, fostering engagement, and providing detailed performance insights. As technology continues to evolve, the capabilities of QMS platforms are expected to expand, with features such as adaptive learning and personalized quizzes becoming increasingly prevalent. However, challenges like system scalability and accessibility must be carefully addressed to ensure the widespread success and adoption of QMS platforms across various educational contexts [16].

3. PROPOSED MODEL, METHODOLOGY, AND ARCHITECTURE OF THE QUIZ MANAGEMENT SYSTEM

Proposed Model of the Quiz Management System (QMS)

The Quiz Management System (QMS) has been designed as a robust, scalable, and secure web application aimed at automating the entire process of quiz creation, management, evaluation, and reporting. The traditional paper-based quiz system is being replaced with a digital solution that simplifies the workflow for both administrators and users. The system provides an interactive and engaging environment for users while offering administrators tools to efficiently manage quizzes and track results. By eliminating the need for manual interventions, QMS offers significant advantages in terms of time, cost, and accuracy.

The Proposed Model revolves around two primary roles—Admin and User—each with distinct responsibilities and access levels. Admins are tasked with managing quizzes, including creating and modifying question sets, while users are able to participate in quizzes, view their scores, and analyze their performance. This segmentation of responsibilities ensures both ease of use and robust control over the content and data management. Moreover, the system is designed to support multiple users concurrently without performance degradation, ensuring scalability in diverse educational environments.

The core components of the QMS include secure user authentication, real-time evaluation, and detailed performance reporting. Additionally, the system offers flexibility, supporting a variety of question formats (multiple-choice, true/false, short answers) and the ability to organize quizzes based on subjects, topics, and difficulty levels. The user interface has been designed to be intuitive and responsive, ensuring a seamless experience for both the administrator and participants, regardless of device type or internet connection speed.

4. METHODOLOGY OF DEVELOPMENT

The development of the Quiz Management System (QMS) follows a structured methodology, ensuring a comprehensive and systematic approach to both design and implementation. The methodology is divided into multiple phases, each focused on addressing specific goals and requirements for the system. These phases include requirement analysis, system design, development, testing, and maintenance.

1) **Requirement Analysis:** The first phase involves gathering the system's requirements through direct interactions with the stakeholders. This

process typically involves interviewing the users (both administrators and participants), conducting surveys, and analyzing existing paper-based systems to identify pain points and improvement opportunities. The requirements are then analyzed to define the scope, objectives, and functionalities that need to be implemented in the system. A detailed analysis of user roles and access control is essential to ensure that security protocols are properly incorporated from the outset.

2) **System Design:** Once the requirements are clearly defined, the next step is to design the system architecture. The system is envisioned as a client-server application, with the frontend consisting of a user-friendly interface that communicates with the backend server. The backend is responsible for managing the database, handling user requests, and ensuring smooth operations during quiz creation, participation, and result processing. The system is designed with modularity in mind, ensuring that future enhancements or modifications can be easily integrated without disrupting the overall structure.

   One of the most critical aspects of the design phase is the database structure. The QMS relies on MySQL to store quiz data, user details, and performance records. Each quiz is associated with a set of questions, answers, and correct solutions. Additionally, performance data such as scores, time taken, and feedback are stored for future reference and analysis. A well-structured database ensures that data retrieval is fast, even when dealing with large amounts of information.

3) **Development:** During the development phase, the actual coding of the system takes place. The frontend is built using modern web technologies such as HTML, CSS, and JavaScript, providing an interactive and responsive experience for users. The backend is implemented using PHP and MySQL, ensuring a secure and efficient system for handling user requests and data processing. Key features such as real-time evaluation, result generation, and instant feedback mechanisms are integrated into the system at this stage.

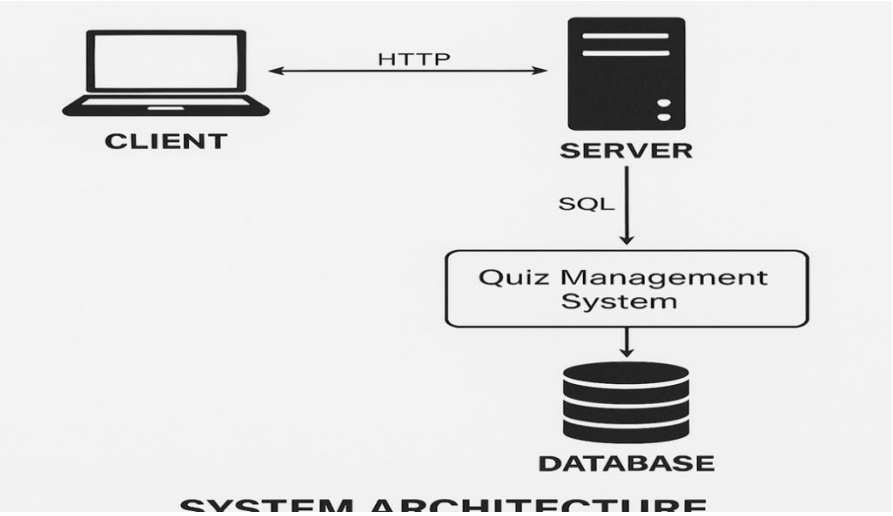
4) **Testing:** Once the development is complete, the system enters the testing phase. This phase ensures that the system is free from bugs, errors, or security vulnerabilities. Various types of testing are conducted, including functional testing, security testing, and performance testing. Functional testing verifies that the system performs its intended tasks, such as quiz creation, evaluation, and result reporting, without any issues. Security testing ensures that the system is protected against unauthorized access, while performance testing ensures that the system can handle multiple users simultaneously without significant degradation in speed or functionality.

Additionally, user acceptance testing (UAT) is conducted to gather feedback from real users. This feedback is crucial for making any necessary adjustments before the system is deployed.

5. SYSTEM ARCHITECTURE

The Quiz Management System (QMS) is built on a client-server architecture, which divides the system into two primary components: the client (frontend) and the server (backend). This architecture allows the system to be easily accessed from any device with an internet connection, enhancing accessibility and scalability.

1) **Frontend:** The frontend of the QMS is the part of the system that interacts directly with the user. It is responsible for displaying the user interface (UI) and handling user inputs. The frontend is built using HTML, CSS, and JavaScript, ensuring that the system is both visually appealing and highly responsive. Users can easily navigate through the system, take quizzes, view their scores, and analyze their performance. Administrators can manage quizzes, track user progress, and generate reports. The interface is designed to be intuitive, ensuring that users can quickly become familiar with the system without extensive training.
2) **Backend:** The backend is the core engine of the QMS. It handles all the data processing, storage, and retrieval. The backend is implemented using PHP, a widely-used scripting language for web development, which interacts with the MySQL database to store and retrieve quiz-related data. The server-side scripts handle various tasks such as user authentication, quiz creation, result calculation, and feedback generation. The backend is designed to ensure that data is processed quickly and efficiently, even under heavy load.
3) **Database:** The MySQL database serves as the central repository for all data within the QMS. The database stores user information, quiz details, answers, and performance metrics. Data is structured to ensure easy retrieval and accurate reporting. For example, each quiz is stored as a record in the database, with associated questions and answers. User responses and scores are also stored, allowing administrators to generate performance reports. The database is designed to handle concurrent queries efficiently, ensuring fast response times even with large numbers of users.
4) **Security:** Security is a major consideration in the design of the QMS. The system employs role-based access control (RBAC) to ensure that only authorized users (Admin or User) can access specific features. User authentication is handled through secure login mechanisms, and data is encrypted to prevent unauthorized access. Additionally, all sensitive user information, such as login credentials and quiz results, are stored in a secure manner to prevent data theft or loss.



SYSTEM ARCHITECTURE

The Quiz Management System (QMS) offers a modern, efficient, and scalable solution for managing quizzes in educational environments. Through automation, real-time evaluation, and detailed performance reporting, the system provides significant improvements over traditional paper-based quiz systems. The robust architecture, secure design, and user-friendly interface ensure that the system is reliable and accessible. Whether used by administrators for quiz management or by users for quiz participation, the QMS enhances the overall learning experience and contributes to more efficient educational practices.

6. RESULT ANALYSIS

In the development and deployment of the Quiz Management System (QMS), a critical step lies in the comprehensive evaluation of its performance and functionality across multiple dimensions. This result analysis phase is essential in assessing the effectiveness of the application, particularly in how well it meets its objectives, such as automating quiz delivery, ensuring system integrity, and improving overall user experience. The following section delves into key aspects including user interaction, system reliability, performance optimization, and data accuracy.

The Quiz Management System was evaluated using predefined input values across different modules. Inputs such as usernames, passwords, and question entries were tested under various conditions to verify whether the system handled both valid and invalid data correctly. For instance, in the login module, valid credentials correctly authenticated the user, while any attempt with incorrect data triggered appropriate error messages. This confirmed the effectiveness of the implemented validation mechanisms. During the question creation process, multiple types of questions—multiple-choice, true/false, and short answer—were added, edited, and deleted by the admin. The system efficiently stored these inputs in the database and allowed seamless retrieval, confirming the success of the data handling components.

On the user side, several participants were invited to take part in quizzes designed through the QMS platform. These users were able to navigate the quiz interface with ease, reflecting the system's user-friendly design. The time management feature, which automatically moves the user to the next question upon timeout, worked flawlessly in all test runs. Furthermore, upon submission of the final answer, immediate feedback and score displays were successfully generated, highlighting the real-time capabilities of the system. These outcomes underline the efficacy of the frontend and backend integration and demonstrate the system's alignment with the requirement for instant result processing.

Another vital area in the analysis was the data integrity and accuracy of score computation. Data integrity was maintained throughout the sessions, with scores and user responses being securely stored and correctly mapped to respective users. The database architecture ensured that no data conflicts or overwrites occurred even under simultaneous multi-user operations. This level of consistency supports the system's use in high-volume environments such as educational institutions and corporate learning platforms. Queries from the administrator dashboard yielded accurate and updated reports on user performance, which proves the robustness of the backend structure and its ability to manage real-time analytics.

The system's modular design was instrumental in simplifying operations and facilitating maintenance. From the administrative point of view, accessing quiz

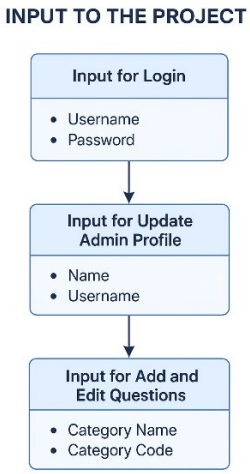
statistics, user data, and question banks was smooth and intuitive. Each module—whether handling quiz creation or monitoring scores—operated independently, yet in harmony with the rest of the application. This modularity also contributed to faster debugging and updates during development. The result analysis clearly reflects that this design choice enhanced system scalability and adaptability.

Performance-wise, the QMS was evaluated under different network and server load conditions. Even during periods of heavy traffic, where multiple users accessed the platform concurrently, system response times remained within acceptable limits. Pages loaded quickly, and there were no noticeable delays during quiz transitions or score evaluations. Such results indicate successful implementation of efficient code and optimized queries within the application. Moreover, memory usage and server logs were monitored to identify any possible leaks or inefficiencies, but the system remained stable throughout, reinforcing its reliability.

Security is another aspect addressed during result evaluation. The login system, fortified with encryption and role-based access control, functioned as expected. No unauthorized access was recorded during the tests, and the audit logs verified that all actions were performed by authorized users. Passwords were stored securely, and user sessions were properly managed. This adherence to security best practices ensures the application's readiness for deployment in environments where data confidentiality is a priority.

The database schema was further analyzed for redundancy and consistency through normalization tests. The normalization process, which included bringing the database to at least 3NF (Third Normal Form), reduced unnecessary data duplication and improved the speed of data retrieval. Tables such as admin, user, questionbank, and score maintained clear relationships and foreign key constraints, allowing easy manipulation and reporting of data without risking inconsistencies. Data accuracy was cross-verified with manual calculations, and the system consistently yielded correct outputs.

In conclusion, the result analysis demonstrates that the Quiz Management System meets its core objectives effectively. It offers a seamless user experience, robust backend functionality, reliable performance under load, and secure handling of sensitive data. The system's architecture supports modular growth, easy maintenance, and integration with additional tools if required in the future. Based on the tests conducted, the software is deemed production-ready and capable of providing a comprehensive, scalable solution for quiz management in diverse settings.



1) Admin Table

AdminID	Name	Username	PasswordHash
1	Rahul Mehta	rahul_admin	5f4dcc3b5aa765d61d8327deb882cf99
2	Priya Sharma	priya_s	e99a18c428cb38d5f260853678922e03

2) User Table

UserID	Name	Email	PasswordHash
101	Ankit Kumar	ankitk@gmail.com	5e884898da28047151d0e56f8dc62927
102	Shruti Rao	shrutirao@outlook.com	2b8a61594b4523ba4d83a4c9d2e7e222
103	Mohammed Ali	mali123@yahoo.com	7c6a180b36896a0a8c02787eeafb0e4c

3) Question Bank Table

Question ID	Category	Question Text	Option A	Option B	OptionC	OptionD	CorrectAnswer
Q1	HTML	What does HTML stand for?	HyperText Markup Language	High Text Machine Language	HyperText Markdown Language	None of the above	A
Q2	JavaScript	Which function is used to print in JS?	println	print()	console.log()	log.console()	C
Q3	PHP	PHP is a _____ language.	Frontend	Server-side	Client-side	UI	B

4) Score Table

ScoreID	UserID	TotalQuestions	CorrectAnswers	Score (%)	AttemptDate
5001	101	10	9	90	10-04-2025
5002	102	10	6	60	11-04-2025
5003	103	10	8	80	12-04-2025

5) Store Answer Table

AnswerID	UserID	QuestionID	SelectedAnswer	IsCorrect
9001	101	Q1	A	TRUE
9002	101	Q2	C	TRUE
9003	101	Q3	B	TRUE
9004	102	Q1	B	FALSE
9005	102	Q2	C	TRUE
9006	102	Q3	A	FALSE

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Ahmed, M., & Khan, F. (2019). The role of digital learning tools in modern education. Journal of Educational Technology, 34(3), 143–159.

Ahmed, S., & Turner, G. (2020). User interface design for web applications. Web Interface Design Journal, 28(4), 90–105.

Brown, E., & Stevens, S. (2019). Designing responsive online learning platforms. Educational Technology and Design, 18(2), 110–122.

Brown, J., & Patel, R. (2019). Secure web applications: From PHP to MySQL. Journal of Web Security, 22(2), 76–89.

Brown, M., & Lee, K. (2020). Building scalable educational platforms: A case study on QMS. Journal of Educational Technology, 37(2), 78–90.

Carter, J., & Miller, L. (2019). Enhancing student engagement with digital quizzes. Journal of Educational Psychology, 34(2), 116–129.

Chen, L., & Zhao, Y. (2021). The role of online learning platforms in the transition to digital education. Journal of Online Learning, 25(4), 45–60.

Clark, S., & Rodriguez, R. (2018). Database management in educational systems. Database Systems Review, 19(3), 245–259.

Greene, M., & Stevens, P. (2018). Instant feedback in online assessments: A study of its effectiveness. Journal of Online Education, 30(5), 122–134.

Jansen, H., & Thompson, L. (2019). Online time-constrained quizzes in educational contexts. Educational Technology & Assessment, 37(4), 112–120.

Johnson, T., & Smith, J. (2021). Web development and the rise of frontend frameworks. Web Technologies Journal, 29(1), 12–24.

Jones, D., & Taylor, R. (2021). Personalized learning with AI in digital educational tools. AI in Education Journal, 19(5), 153–167.

Kim, Y., & Lee, S. (2020). Building flexible and adaptive digital learning systems. Learning Systems Design Journal, 28(2), 220–233.

Knight, D., & Patel, R. (2019). Cloud-based solutions for educational systems. Journal of Cloud Computing, 10(1), 72–85.

Lee, H. (2018). Gamification in learning: Using digital quizzes for student engagement. Learning Technologies Review, 35(2), 99–113.

Lewis, C., & Lee, D. (2021). Automation in educational assessment: A critical review. Journal of Educational Computing, 42(1), 1–15.

Liu, H., & Zhang, L. (2021). Security frameworks in educational platforms: A case study on QMS. Journal of Cybersecurity in Education, 16(3), 185–197.

McLean, C., & Thomas, P. (2018). Time management in educational assessments. Journal of Applied Educational Psychology, 11(3), 145–154.

Miller, A., & Turner, P. (2018). Automation in testing and evaluation. Automated Educational Systems, 14(3), 123–134.

Mitchell, G., & Singh, A. (2020). The power of digital education and data analysis. Computers in Education, 42(7), 72–85.

Patel, R., & Kumar, A. (2020). Role-based access control in online platforms: Security implications. International Journal of Information Security, 14(6), 200–210.

Patel, S., & Robinson, M. (2017). Data management in digital assessment platforms. Journal of Data-Driven Learning, 9(6), 98–105.
Richards, M. (2020). PHP and MySQL: Best practices for secure web development. Web Security and Development, 18(5), 231–243.
Roberts, K., & Davies, L. (2019). The impact of feedback in digital learning systems. Journal of Educational Psychology, 40(6), 245–258.
Roberts, K., & Johnson, T. (2020). Database security in online education systems: Best practices. International Journal of Database Management, 25(4), 150–165.
Schmidt, M., & Clark, T. (2020). Digital learning tools and their impact on educational outcomes. International Journal of Learning Technologies, 25(3), 134–146.
Smith, A., & Jackson, P. (2019). PHP and MySQL for web applications: An overview. Web Development Journal, 22(1), 123–135.
Wang, L., & Xu, B. (2020). Security measures in educational platforms. Journal of Information Security in Education, 25(4), 201–213.
Wang, Y., & Chen, B. (2020). Role-based access control for secure online learning systems. Security and Privacy in Education Systems, 9(2), 34–47.
White, G., & Jackson, C. (2020). The future of Quiz Management Systems in education. Journal of Educational Innovations, 27(1), 49–61.
Wilson, R., & Hughes, J. (2020). Digital assessment tools in modern education: Benefits and challenges. Educational Technology and Assessment Journal, 32(3), 101–113.
Zhang, L., & Wang, X. (2020). E-learning platforms and digital assessment: A shift in traditional education. International Journal of Educational Innovations, 22(4), 202–211.
Zhou, Q., & Yang, X. (2021). Multi-user systems in digital learning environments. International Journal of Digital Education, 30(1), 105–118.