



Mindarena: A Real-Time Multiplayer Quiz Battle Game

GS Farhana, M Ankitha, M Aslam Basha, K Kedharnath, T Gopichandh

Department of Computer Science and Engineering Gates Institute of Technology , Andhra Pradesh, India.

Correspondence

GS Farhana

Department of Computer Science and Engineering, Gates Institute of Technology, Andhra Pradesh, India.

- Received Date: 11 Feb 2026
- Accepted Date: 02 Apr 2026
- Publication Date: 25 Apr 2026

Keywords

Dynamic Scoring, Quiz Battle Game, Real Time Synchronization, User Authentication, Web Sockets.

Copyright

© 2026 Authors. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International license.

Abstract

The Quiz Battle Game is a Java-based real-time quiz application designed to deliver an interactive and competitive learning experience. It allows two players to compete simultaneously in timed quiz battles, answering questions in real time while aiming for higher scores. Real-time synchronization ensures instant updates of player actions, timers, scores, and power-ups, creating a smooth multiplayer experience. The back-end is developed using Spring Boot and Servlet, handling business logic, user authentication, game sessions, and data processing. Web Sockets enable low-latency, bidirectional communication between players and the server. The user interface is built with Spring Boot, managing the overall multiplayer quiz flow. A MySQL database is used for persistent storage of user profiles, quiz questions, match history, scores, and leader boards. An Admin module allows authorized users to dynamically create, update, and manage quizzes and questions. Features such as dynamic scoring, power-ups, and leader boards enhance engagement, motivation, and replay value. Overall the application showcases the practical use of object-oriented programming, real-time systems, database management, and client-server architecture, making it a scalable solution for interactive educational gaming.

Introduction

The motivation behind developing the Quiz Battle Game (Multiplayer / Advanced) is to make learning more interactive, engaging, and competitive. Traditional quiz systems are often static and single-user, which reduces user interest and participation over time. There is a need for a system that not only tests knowledge but also encourages real-time interaction and quick thinking.

This project aims to address these limitations by introducing a real-time multiplayer quiz platform where users can compete with each other. The inclusion of features such as live score updates, time-based questions, and leaderboards enhances user motivation and involvement. Additionally, the project provides an opportunity to apply practical concepts like object-oriented programming, real-time communication, and database management, making it both educational and technically valuable. Traditional quiz applications are mostly single-user and static, lacking real-time interaction and competitive features. These systems do not provide instant feedback, live score comparison, or engaging gameplay, which reduces user interest and limits the overall learning experience.

There is a need for a system that enables real-time multiplayer participation, where users can compete simultaneously, receive instant updates, and stay actively engaged.

The challenge is to design and develop a scalable application that ensures synchronized gameplay, accurate scoring, and efficient data management while maintaining a smooth and interactive user experience.

Related Work

The literature survey presents an overview of quiz systems, e-learning platforms, and the technologies used in their development. It focuses on understanding various approaches, along with their features and limitations, to improve user engagement and system performance.

Many quiz applications are designed for single-user environments and lack real-time interaction and competitive elements. They offer limited feedback and do not support dynamic participation, which reduces their effectiveness as engaging learning tools. With advancements in technologies such as Java frameworks, Web Sockets, and MySQL, it is possible to develop scalable and real-time applications support the development of the proposed Quiz Battle Game (Multiplayer / Advanced), which aim to provide an interactive and competitive learning experience.

Lack of Real-Time Interaction: The system does not support live multiplayer gameplay, which limits user engagement and reduces the excitement of competition. **Single-User Experience:** Most applications are designed for individual use, preventing users from

Citation: Farhana GS, Ankitha M, Basha MA, Kedharnath K, Gopichandh T. Mindarena: A Real-Time Multiplayer Quiz Battle Game. GJEIIR. 2026;6(4):239.

competing or interacting with other. Basic Scoring Mechanism: Scoring is only based on correct answers and does not consider response time or advanced features, making it less dynamic.

Existing quiz applications face several limitations that reduce their overall effectiveness and user engagement. One of the major issues is the lack of real-time interaction, as most systems do not support live multiplayer gameplay, which decreases the excitement of competing with others. In addition, many platforms provide only a single-user experience, preventing users from interacting or competing with other participants. Another limitation is the basic scoring mechanism, where scores are calculated only based on correct answers and do not consider response time or performance-based evaluation.

Proposed Methodology

The goal of the proposed system is to develop a real-time multiplayer quiz application that provides an interactive and competitive learning environment. The system aims to enhance user engagement by allowing players to participate simultaneously and receive feedback during gameplay. It focuses on implementing accurate and fair scoring mechanisms based on response time and correctness, while ensuring smooth real-time communication using web sockets.

Based on the study, it is evident that integrating real-time communication, dynamic scoring, and competitive features can significantly enhance user participation and effectiveness. These insights provide a strong foundation for the development of the proposed Quiz Battle Game.

The proposed system focuses on developing a real-time multiplayer quiz application that creates an engaging and competitive learning environment for users. It enables multiple players to participate simultaneously, promoting interaction and healthy competition. By incorporating instant feedback mechanisms during gameplay, users can quickly understand their performance and improve their knowledge. The system is designed to be user-friendly and efficient, ensuring smooth navigation and accessibility. Overall, the application aims to make learning more dynamic, interactive, and enjoyable through technology-driven solutions that encourage continuous participation and knowledge enhancement.

A key objective of the system is to enhance user engagement through real-time participation and communication. Players can compete with others simultaneously, which increases excitement and motivation. The application ensures that feedback is delivered instantly, allowing users to track their progress and performance effectively. This immediate response system helps users identify strengths and areas for improvement. Additionally, the competitive nature of the platform encourages repeated participation, thereby improving learning outcomes. The system design emphasizes responsiveness and interactivity, ensuring that users remain actively involved throughout the quiz sessions without delays or interruptions.

The system also focuses on implementing accurate and fair scoring mechanisms based on response time and correctness of answers. This ensures that all participants are evaluated transparently and consistently. By combining speed and accuracy, the scoring model rewards both knowledge and quick thinking. Real-time communication is achieved using web sockets, which enable seamless data exchange between users and the server. This technology ensures minimal latency and smooth gameplay experience. As a result, players can enjoy uninterrupted quiz sessions, making the application reliable and efficient for real-time competitive learning environments.

Based on the study, integrating real-time communication, dynamic scoring, and competitive features significantly

enhances user participation and learning effectiveness. These insights form a strong foundation for developing the proposed Quiz Battle Game. The system will be supported by a server environment using Apache Tomcat, ensuring efficient handling of requests and smooth backend operations. It will operate on a Windows-based operating system, providing compatibility and ease of deployment. Together, these components contribute to building a scalable, reliable, and high-performance application that meets the requirements of modern interactive learning systems.

Hardware Requirements

The selection of hardware is very important in the existence and performance of any software. The size and capacity are the main requirements. The typical web server must have the following specifications for good performance.

Server : Apache Tomcat
OS : Windows

which aims to overcome existing limitations and deliver a more interactive and engaging quiz platform. The analysis phase is an essential step in the development of any software project. It focuses on understanding the requirements, features, and overall functionality needed to build an effective system. This phase helps in identifying user needs and ensures that the system is designed to meet its intended objectives.

In this project, the analysis helps in defining the structure and working of the quiz application, including user interaction, gameplay flow, scoring mechanism, and data management. It provides a clear foundation for designing and implementing a reliable and efficient system.

Software Requirements Specification:

User requirements describe the functionalities and features that the system must provide from the user's perspective. These requirements define how users interact with the system and what services they expect.

They also require access to monitor user activities and maintain the overall system efficiently.

- 1) Hardware requirements.
- 2) Software requirements.

The software requirements specification is produced at the end of the analysis task.

Software requirement is a difficult task, only decided after testing whether it fits the requirements.

Frontend : Swing
Backend : Spring Boot
Communication : Web Sockets
Database : MySQL
ORM : Hibernate

In addition to the processor, the system should have at least 4 GB of RAM. This amount of memory is necessary to support the smooth functioning of the application, especially when multiple processes such as the web server, database operations, and browser-based client interfaces are running simultaneously. Adequate RAM ensures better responsiveness and reduces delays during gameplay.

Feasibility analysis helps in identifying potential problems before development begins and ensures that the project can be completed with the available resources, time, and technology. It also supports decision-making by determining whether adding new modules and improving the existing system will be successful.

Players can easily join quiz battles, answer questions, and view results, while administrators can manage quiz content, update questions, and monitor user performance efficiently.

Feasibility Conclusion

Based on the feasibility analysis, the proposed Quiz Battle Game system is practical and suitable for successful implementation. The system is technically feasible, as it uses reliability.

Efficiency

The system is optimized to provide fast response time especially for real-time quiz battles between multiple players.

Reliability

The system ensures accurate data processing, stable performance, and minimal errors during operation.

Usability

The user interface is designed to be simple, interactive, and easy to navigate for both players and administrators

Design

The design phase is an important stage in software development where the system's structure, architecture, and workflow are planned. In this phase, the requirements gathered during analysis are converted into a clear blueprint for development. It defines how different components of the system are organized and how they interact to perform the required functions

Design Principles Modularity:

The application is divided into smaller modules such as user management, quiz management, scoring system, and communication module for easier development and maintenance.

Scalability:

The system is designed to handle an increasing number of users and data without affecting performance.

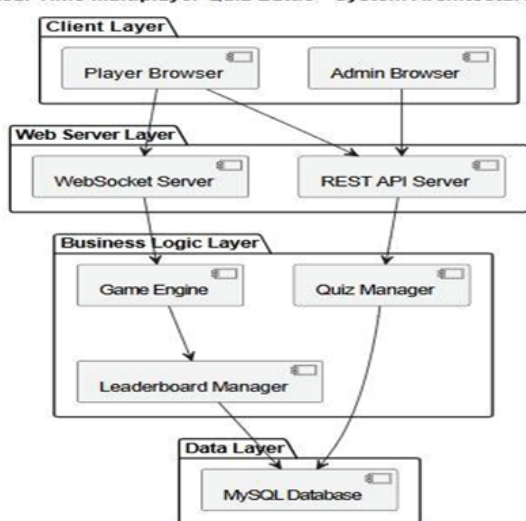
Maintainability:

The design allows developers to update, modify, or debug the system easily without affecting other modules.

Architecture Diagram

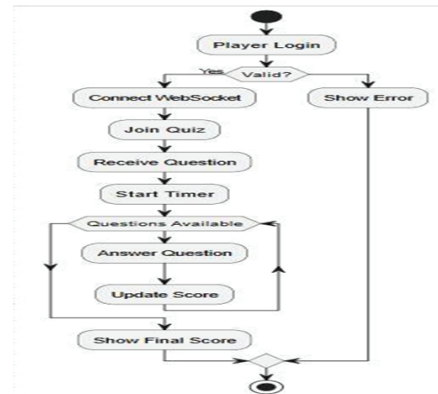
The architecture diagram represents the overall structure of the system and shows how different modules interact with each other. It helps in understanding the flow of data and communication between users, application modules, and the database.

Real-Time Multiplayer Quiz Battle - System Architecture



This diagram helps in visualizing the overall functioning of the system without going into detailed internal

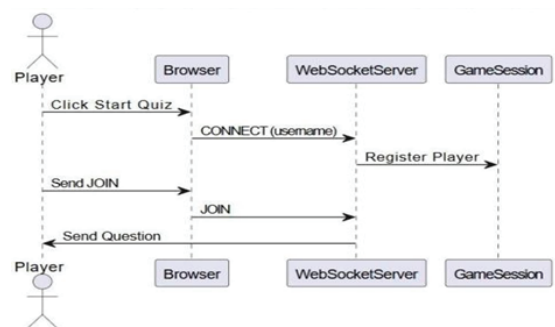
Flowchart



The flow chart of the system represents the step-by-step working process of the quiz application, starting from user login/registration, accessing quizzes, answering questions, score calculation, and displaying results. It helps in understanding the overall flow and decision-making process of the system.

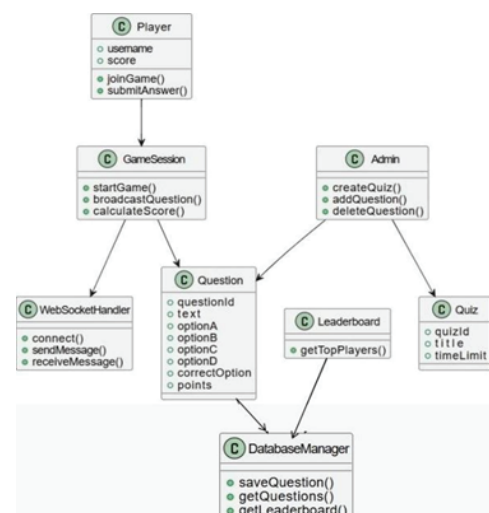
Sequence Diagram

The sequence diagram shows the interaction between the user, system, modules, and database in a step-by-step order for functions such as login, quiz participation, score calculation, and result display.



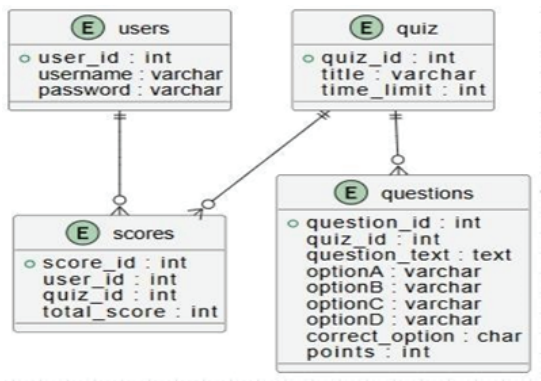
Class Diagram

The class diagram represents the static structure of the quiz application by showing the main classes such as User, Quiz, Question, Score, and Admin, along with their attributes, methods, and relationships.



ER diagram

The ER (Entity Relationship) Diagram represents the database structure of the quiz application by showing the main entities such as User, Quiz, Question, Score, and Admin and the relationships between them.



Implementation And Results:

Implementation is the phase where the system design is converted into a working application. It involves planning, analyzing the existing system, and developing methods for smooth system transition. During this stage, the modules are developed, integrated, and tested using selected technologies.

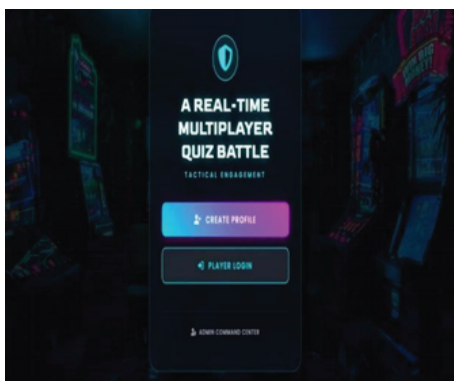
MODULES:

- User Module
- Quiz Management Module
- Multiplayer Communication Module
- Score Module
- Admin Module

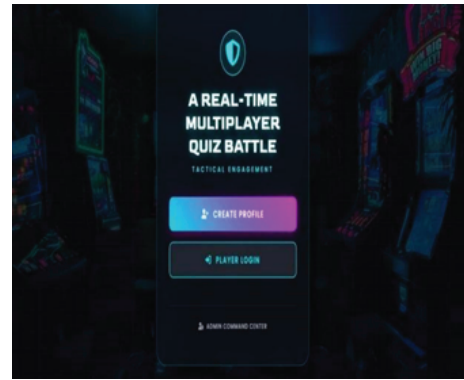
JAVA TECHNOLOGY:

Java is used as the primary programming language for developing the Quiz Battle Game due to its platform independence, security, and object-oriented features. The system uses Swing for the user interface, Spring Boot for backend development, WebSocket for real-time multiplayer communication, and MySQL for data storage. These technologies together support user management, quiz execution, score updates, and efficient system performance.

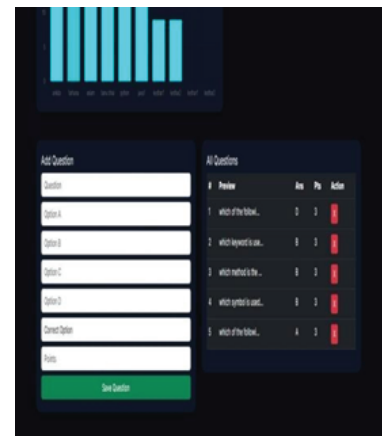
OUTPUT SCREENS:



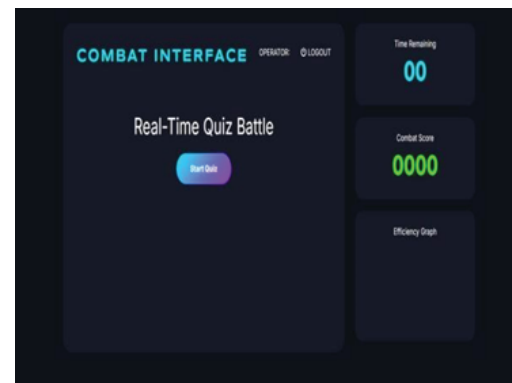
: Home Page of Quiz Battle Game



Login Admin Page



Admin Panel for Adding and Managing



Combat Interface

Testing And Validation

Testing is performed to identify errors and ensure that the system functions according to user requirements. It verifies that each component of the Quiz Battle Game works correctly and that the final application performs reliably. The main goal of testing is to detect defects in requirements, design, documentation, and code at an early stage. Proper testing ensures that the system provides secure user authentication, correct quiz execution, accurate score calculation, smooth real-time communication, and stable performance.

Several testing activities were conducted during development. User input and authentication testing ensures that registration and login work correctly with both valid and invalid credentials. Quiz execution testing verifies the proper display of questions,

timer functionality, answer submission, and synchronized question delivery to both players. Real-time communication testing checks the WebSocket connection and ensures instant updates of scores and gameplay events. Score calculation testing confirms that scores are calculated correctly and displayed in real time. User interface testing ensures smooth navigation and a user-friendly experience. System performance testing evaluates stability when multiple users access the application simultaneously.

Different types of testing were used, including Unit Testing, Integration Testing, Functional Testing, and System Testing. Additional methods such as Black Box Testing, White Box Testing, and Grey Box Testing were applied to validate system functionality, internal logic, and module interaction.

The overall testing strategy focuses on verifying login and registration, quiz functionality, real-time communication, score calculation, and admin operations, ensuring that the system performs efficiently and meets user expectations.

Conclusion

The Testing and Validation phase ensures that the Quiz Battle Game system functions correctly and meets all specified requirements. Various testing methods such as unit testing, integration testing, system testing, and functional testing were performed to verify the accuracy and performance of all modules. The results confirm that the system handles both valid and invalid inputs effectively, ensuring proper validation and error handling. It provides accurate score calculation, smooth quiz execution, and reliable real-time communication between players. Key features such as user authentication, quiz management, and leader board updates work efficiently. Overall, the system demonstrates good performance, stability, and usability, delivering a reliable and engaging multiplayer quiz experience.

The Testing and Validation phase plays a crucial role in ensuring that the Quiz Battle Game system operates correctly and fulfills all specified requirements. This phase focuses on identifying errors, verifying functionalities, and validating the overall performance of the application. By conducting thorough testing, developers can ensure that each component of the system works as intended. It also helps in improving the quality and reliability of the application. As a result, the system becomes more stable and user-friendly, providing a seamless and efficient experience for users during gameplay sessions.

Various testing methods were implemented to evaluate the system comprehensively, including unit testing, integration testing, system testing, and functional testing. Unit testing ensures that individual modules perform correctly, while integration testing verifies that different components work together without issues. System testing evaluates the complete application to ensure it meets the specified requirements. Functional testing focuses on validating the expected behavior of features. These combined testing approaches help in detecting and resolving potential issues, ensuring that the system functions accurately and efficiently under different conditions and usage scenarios.

The testing results confirm that the system effectively handles both valid and invalid inputs, ensuring strong validation and

proper error handling mechanisms. This prevents unexpected crashes and enhances system reliability. Users are guided with appropriate messages when incorrect inputs are provided, improving usability and user experience. The system also maintains data integrity and consistency during operations. By handling different input scenarios efficiently, the application ensures smooth functioning without interruptions. This level of validation is essential for maintaining the overall robustness and dependability of the Quiz Battle Game system.

Additionally, the system demonstrates accurate score calculation, smooth quiz execution, and reliable real-time communication between players. The scoring mechanism is designed to consider both correctness and response time, ensuring fairness and competitiveness. Real-time communication features enable players to interact and compete without delays, enhancing engagement. The quiz flow is seamless, allowing users to participate without disruptions. These functionalities contribute to a highly interactive and enjoyable gaming experience. The system's ability to maintain performance during real-time operations highlights its efficiency and technical strength.

The system was also evaluated under different load conditions to ensure its scalability and performance during peak usage. Multiple users were simulated to participate in quiz sessions simultaneously, testing the system's ability to handle concurrent requests efficiently. The results indicate that the application maintains consistent performance without significant delays or failures. This demonstrates that the backend architecture and real-time communication mechanisms are capable of supporting multiple users at once. Such performance testing ensures that the Quiz Battle Game remains reliable and responsive even during high user activity.

Furthermore, usability testing was conducted to assess the overall user experience and interface design of the application. Feedback from users helped identify areas for improvement in navigation, layout, and accessibility. The system was found to be intuitive and easy to use, allowing users to participate in quizzes without confusion. Minor enhancements were incorporated to improve clarity and interaction. These improvements contribute to a more engaging and satisfying experience for users. Overall, the application successfully meets user expectations by combining functionality, performance, and ease of use.

Finally, compatibility testing was performed to verify that the system works effectively across different environments and configurations. The application was tested on various system setups to ensure consistent performance and functionality. It was found to operate smoothly without compatibility issues, providing the same user experience across platforms. This ensures that users can access the system without technical difficulties. The overall testing process confirms that the application is robust, adaptable, and ready for deployment. These results highlight the system's capability to deliver a high-quality and consistent multiplayer quiz experience.

References

1. Oracle Corporation, Java SE Documentation. Available at: Java SE Documentation
2. Oracle Corporation, JDBC API Documentation. Available

- through the Java SE documentation.
3. Spring Team, Spring Boot Documentation. Available at: Spring Boot Documentation
4. Spring Team, Spring Framework Documentation. Available at: Spring Framework Reference.
5. Hibernate Team, Hibernate ORM Documentation. Official documentation for object-relational mapping in Java.
6. OMG (Object Management Group), Unified Modeling Language (UML) Specification. Official reference for UML diagrams.
7. JSON.org, Introducing JSON. Reference for structured data representation used in APIs and communication.
8. Spring Team, Spring Boot Reference Documentation. Available at: Spring Boot Docs
9. Object Management Group (OMG), Unified Modeling Language (UML) Specification. Available at: UML Specification
10. Gabriel L. Muller, HTML5 WebSocket Protocol and its Application to Distributed Computing, arXiv, 2014. .
11. Nagesh, C., Chaganti, K.R. , Chaganti, S. , Khaleelullah, S., Naresh, P. and Hussan, M. 2023. Leveraging Machine Learning based Ensemble Time Series Prediction Model for Rainfall Using SVM, KNN and Advanced ARIMA+ E-GARCH. International Journal on Recent and Innovation Trends in Computing and Communication. 11, 7s (Jul. 2023), 353–358. DOI:<https://doi.org/10.17762/ijritcc.v11i7s.7010>.
12. S. Khaleelullah, P. Marry, P. Naresh, P. Srilatha, G. Sirisha and C. Nagesh, "A Framework for Design and Development of Message sharing using Open-Source Software," 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2023, pp. 639-646, doi:10.1109/ICSCDS56580.2023.10104679.
13. Ramesh Kumar Ramaswamy, Pannangi Naresh, Chilamakuru Nagesh, Santhosh Kumar Balan, Multilevel thresholding technique with Archery Gold Rush Optimization and PCNN-based childhood medulloblastoma classification using microscopic images, Biomedical Signal Processing and Control, Volume 107, 2025,107801, ISSN 1746-8094, <https://doi.org/10.1016/j.bspc.2025.107801>.
14. K. R. Chaganti, B. N. Kumar, P. K. Gutta, S. L. Reddy Elicherla, C. Nagesh and K. Raghavendar, "Blockchain Anchored Federated Learning and Tokenized Traceability for Sustainable Food Supply Chains," 2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS), Gobichettipalayam, India, 2024, pp. 1532-1538, doi: 10.1109/ICUIS64676.2024.10866271.
15. Mohammed Inayathulla and Karthikeyan C, "Image Caption Generation using Deep Learning For Video Summarization Applications" International Journal of Advanced Computer Science and Applications(IJACSA), 15(1), 2024. [http:// dx.doi.org/10.14569/IJACSA.2024.0150155](http://dx.doi.org/10.14569/IJACSA.2024.0150155).
16. Mohammed, I., Chalichalamala, S. (2015). TERA: A Test Effort Reduction Approach by Using Fault Prediction Models. In: Satapathy, S., Govardhan, A., Raju, K., Mandal, J. (eds) Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1. Advances in Intelligent Systems and Computing, vol 337. Springer, Cham. https://doi.org/10.1007/978-3-319-13728-5_25
17. Inayathulla, M., Karthikeyan, C. (2022). Supervised Deep Learning Approach for Generating Dynamic Summary of the Video. In: Suma, V., Baig, Z., Kolandapalayam Shanmugam, S., Lorenz, P. (eds) Inventive Systems and Control. Lecture Notes in Networks and Systems, vol 436. Springer, Singapore. https://doi.org/10.1007/978-981-19-1012-8_18.
18. Mohammed and C. Karthikeyan, "Object detection in video summarization for video surveillance applications," Yugoslav Journal of Operations Research, vol. 35, no. 3, pp. 523–546, 2025. doi:10.2298/YJOR240615010I.
19. Inayathulla Mohammed and K. R. Rao, "Enhancing real-time violence detection in video surveillance using hybrid deep learning model," Journal of Web and User Applications, vol. 16, no. 1, 2025. doi:10.58346/JOWUA.2025.11.021.
20. G. Raju, K. Sushmitha, M. Priyanka, E. Sai Leela, M. Vani Chowdary, and A. Yashwantha, "Real Time Hand Gesture Recognition Using CNN," Global Journal of Engineering Innovations & Interdisciplinary Research (GJEIIR), vol. 5, no. 2, 2025, doi: 10.33425/3066-1226.1101.