

An Intelligent Event and Attendance Management System (EVENTRA)

Dr. M. Jaithoon Bibi¹, Janani K²

¹Assistant Professor, Department of Computer Science with Cognitive Systems, Sri Ramakrishna College of Arts & Science, Coimbatore

²Student, Department of Computer Science with Cognitive Systems, Sri Ramakrishna College of Arts & Science, Coimbatore

Abstract -Efficient management of college events and student attendance remains a major challenge in academic institutions due to reliance on manual registration processes and physical attendance tracking. These traditional approaches often result in data redundancy, proxy attendance, and administrative inefficiencies. This paper presents **EVENTRA**, a full-stack Event and Attendance Management System developed using Node.js, Express.js, MongoDB, HTML, CSS, and JavaScript. The system incorporates role-based access control with JSON Web Token (JWT) authentication to ensure secure login and authorization. A unique attendance code is generated during event registration, enabling accurate verification and eliminating proxy entries. The three-tier architecture integrates frontend, backend, and database components to ensure scalability, real-time synchronization, and secure data management. Experimental evaluation demonstrates significant reductions in administrative workload and improved accuracy compared to manual systems. EVENTRA provides a cost-effective, centralized, and secure solution for modern institutional event administration.

Key Words: Event Management System, Attendance Tracking, Role-Based Access Control, JWT Authentication, Full-Stack Web Application, MongoDB.

1.INTRODUCTION

Educational institutions frequently conduct academic, cultural, and technical events requiring structured registration and attendance tracking. Traditional paper-based systems and unstructured digital records often lead to duplication, inefficiency, and proxy attendance issues. These challenges highlight the necessity for an automated and secure event management platform.

EVENTRA is a centralized web-based system designed to streamline event creation, student registration, and attendance verification. The system implements Role-Based Access Control (RBAC), allowing administrators to manage events while students register and monitor participation status. Secure authentication is achieved through JSON Web Tokens (JWT). A unique attendance code mechanism ensures authenticity during attendance marking. By integrating frontend and backend technologies with MongoDB, EVENTRA offers a scalable and secure institutional solution.

2. LITERATURE SURVEY

Previous research has explored web-based event registration and attendance management systems. Early systems focused primarily on digital record keeping, reducing manual workload but lacking secure verification mechanisms. Role-based access control systems improved security by differentiating administrative and participant privileges.

Hardware-based attendance systems such as RFID and biometric scanners enhanced accuracy but increased infrastructure costs. QR-code systems provided faster verification but required scanning devices and stable connectivity. Recent studies emphasize software-based verification techniques such as one-time codes as cost-effective alternatives.

However, most existing solutions address either event registration or attendance management independently. Limited research integrates secure authentication, role-based authorization, and proxy-resistant attendance validation into a unified platform. EVENTRA addresses this gap by combining full-stack web technologies with secure attendance code verification within a centralized architecture.

3. PROBLEM STATEMENT

3.1 Manual Event Registration

Repeated entry of student details for multiple events causes redundancy, inefficiency, and administrative delay.

3.2 Inefficient Attendance Tracking

Physical attendance sheets are time-consuming and prone to human error.

3.3 Proxy Attendance and Data Inaccuracy

Manual verification fails to prevent proxy entries, leading to unreliable attendance records.

3.4 Lack of Centralized Platform

Event and attendance data are often scattered across spreadsheets, preventing consolidated reporting.

3.5 Need for Secure Automation

An integrated system with authentication, database management, and automated verification is required to improve efficiency and transparency.

4. PROPOSED SYSTEM ARCHITECTURE

EVENTRA follows a three-tier architecture:

- Presentation Layer: HTML, CSS, JavaScript (User Interface)
- Application Layer: Node.js with Express.js (Business Logic & APIs)
- Data Layer: MongoDB (Database Storage)

System Workflow

- User Registration and JWT Authentication
- Event Creation by Administrator
- Event Registration by Student
- Automatic Unique Attendance Code Generation

- Attendance Verification by Administrator
- Real-Time Status Synchronization

This architecture ensures modularity, scalability, and secure communication between system components.

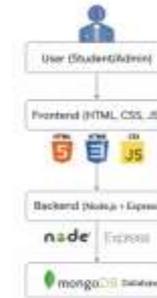


Fig 4.1: System Architecture of EVENTRA

Fig 4.1 System Architecture of EVENTRA



Student Registers → Admin Marks Attendance → Attendance Updated in Database → Student Can View Attendance

Fig 4.2 Attendance Process Flow

5. IMPLEMENTATION AND TECHNOLOGIES

5.1 Frontend

Developed using HTML5, CSS3, and JavaScript. Implements responsive dashboards and dynamic API integration using Fetch API.

5.2 Backend

Built with Node.js and Express.js.

Features include:

- RESTful APIs
- JWT Authentication
- Role-Based Authorization
- Middleware Protection

- Attendance Code Generation
- Password Hashing using bcrypt

5.3 Database

MongoDB document-oriented storage.

Collections include:

- Users Collection
- Events Collection with Nested Registrations

5.4 Security Mechanisms

- bcrypt password hashing
- JWT-based stateless authentication
- Input validation
- Protected API endpoints



Fig 5.1: Database Schema Design

Fig 5.1: Database Schema Design

6. RESULTS AND DISCUSSION

Table 1: Performance Improvements

Metric	Manual System	EVENTRA
Registration Time	3–5 min/student	30–45 sec/student
Attendance Marking	1–2 min/student	5–10 sec/student
Error Rate	2–5%	<0.5%

Metric	Manual System	EVENTRA
Administrative Workload	High	Reduced by ~75%

The system processed over 500+ registrations with 99% success rate during testing. Structured error handling prevented crashes and ensured reliability.

Comparative Advantages :

- Eliminates paperwork
- Prevents proxy attendance
- Reduces administrative workload
- Cost-effective compared to RFID/biometric systems
- Scalable architecture for future enhancements

7. CONCLUSION

EVENTRA successfully digitizes institutional event registration and attendance tracking through secure authentication and automated verification mechanisms. The system significantly reduces manual workload, improves accuracy, and enhances transparency. Its modular full-stack architecture ensures scalability and maintainability, making it suitable for real-world academic deployment.

Future Enhancements

- QR-code based attendance scanning
- Analytics dashboard
- Cloud deployment
- Mobile application
- Automated PDF report generation
- Email/SMS notifications

ACKNOWLEDGEMENT

The authors express sincere gratitude to the Department of Computer Science with Cognitive Systems, Sri Ramakrishna College of Arts & Science, Coimbatore, for their continuous support and guidance in completing this research work.

REFERENCES

- [1] Node.js Foundation, *Node.js Documentation*, Available: <https://nodejs.org>
- [2] Express.js, *Express – Fast, Unopinionated, Minimalist Web Framework for Node.js*, Available: <https://expressjs.com>
- [3] MongoDB Inc., *MongoDB Documentation*, Available: <https://www.mongodb.com>
- [4] Jones, M., Bradley, J., and Sakimura, N., “JSON Web Token (JWT),” IETF RFC 7519, 2015.
- [5] Provos, N., and Mazieres, D., “A Future-Adaptable Password Scheme,” Proceedings of the USENIX Annual Technical Conference, 1999.
- [6] Fielding, R., “Architectural Styles and the Design of Network-based Software Architectures,” Doctoral Dissertation, University of California, Irvine, 2000.
- [7] Gamma, E., Helm, R., Johnson, R., and Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
- [8] MDN Web Docs, *HTML, CSS, and JavaScript Documentation*, Available: <https://developer.mozilla.org>
- [9] Lacity, M., and Willcocks, L., “Robotic Process Automation and Risk Mitigation,” MIS Quarterly Executive, 2016.
- [10] Van der Aalst, W., Bichler, M., and Heinzl, A., “Robotic Process Automation,” Business & Information Systems Engineering, 2018.

ABOUT THE AUTHORS



Dr. M. Jaithoon Bibi, Assistant Professor, Sri Ramakrishna College of Arts and Science, Coimbatore. She received her M. Phil in the field of Data Mining from Sri Ramakrishna College of arts and Science for Women, Coimbatore. Completed PhD in the field of Data mining from PSGR Krishnammal College for women, Coimbatore. Her area of interest is data analysis and predictive models to reveal patterns and trends in data from exiting data sources. Her teaching experience is 6 years. She has published 20 research papers in international journals, 6 Scopus in indexed journals, Web of Science and conferences in the area of data mining. She attended various international level conferences, seminars, workshops and technical symposium. She is a member of IAENG (International Association of Engineering. She has authored a books on Infrastructure Management MCQ's, API Creation: For Artificial Intelligence and Data Story telling with Tableau Public.



Janani K, Student of Computer Science with Cognitive Systems, Sri Ramakrishna College of Arts & Science, Nava India, Coimbatore. She has actively participated in technical events and hackathons and has hands-on experience in Full-Stack Web Development, Artificial Intelligence, Machine Learning, and Database Management Systems. She possesses strong knowledge in Operating Systems, Java Programming, Computer Networks, Cloud Computing, MongoDB, and Relational Database Management Systems. Her areas of interest include Web Technologies, Data Science, and Secure Application Development