IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Leveraging Real-Time Systems For Efficient Event Planning And Feedback In Academic Settings

[1]Alsha Thomas, [2]Ambily Mohan, [3]Sarath Murukan

[1]Assistant Professor, [2]Assistant Professor, [3]Student

[1]Computer Science,

[1]Adi Shankara Institue of Engineering and Technology, Kalady, India

Abstract: A comprehensive platform that digitizes educational organization's event planning, administration, and participation is the Real Time Notification and Feedback Management System. In order to keep students informed and involved, its main objective is to consolidate all event-related procedures, from registration and announcements to updates and feedback. For students, the system serves as a one-stop solution for discovering and participating in college events. Students can register for events, track their registration status, and receive real-time notifications about updates or reminders. A personalized dashboard allows them to view their registered events, event details such as dates, venues, and schedules. Admins are provided with powerful tools to manage events effectively.

Index Terms **- Sentiment, Vue.js, Node.js, OAuth, RBAC**

I. INTRODUCTION

In today's dynamic and rapidly evolving business environment, project management plays a pivotal role in achieving strategic objectives, driving innovation, and promoting the sustainable growth of organizations. Across various sectors—including business, technology, research, and development—effective project management is essential for guiding initiatives from project initiation through to successful completion. This discipline encompasses project planning, execution, and performance evaluation, ensuring that projects are delivered on time, within budget, and according to quality standards. The significance of structured project planning and resource optimization has increased markedly due to rapid technological advancements, heightened market competition, stringent regulatory compliance, and shifting customer expectations. Employing robust project management methodologies, tools, and techniques enables organizations to enhance efficiency, control costs, mitigate risks, and improve overall productivity. Additionally, project management facilitates clear communication and collaboration among diverse teams, fostering a unified approach toward common goals. The integration of modern digital tools, such as real-time data analytics and cloud-based platforms, further strengthens decision-making processes and responsiveness to changes. Ultimately, mastering project management empowers organizations to remain adaptable, competitive, and resilient in an ever-changing global landscape, positioning them to capitalize on emerging opportunities and navigate challenges effectively.

II. OBJECTIVE

The primary objective is to analyze, develop, and implement an optimized system or process that ensures operational efficiency, effectiveness, and innovation within the targeted domain. This involves addressing the critical challenges and leveraging the key opportunities identified during the system analysis phase. A comprehensive understanding of these challenges—including technical limitations, user requirements, and environmental constraints—enables the formulation of the most effective and sustainable solutions. It also focuses on designing a structured methodology to establish clear and measurable goals, enabling precise strategy formulation for project execution and continuous performance monitoring. Emphasis is placed on utilizing best practices in project management, including risk assessment, resource allocation, and quality assurance, to maximize productivity and achieve desired outcomes. By integrating advanced tools and techniques, the system aims to enhance decision-making processes, streamline workflow automation, and improve overall system reliability and scalability. The ultimate goal is to deliver a robust and adaptable solution that meets stakeholder expectations while fostering continuous improvement and driving technological innovation.

III. PROBLEM DEFINITION

A well-defined problem statement is essential for understanding the key challenges and formulating effective solutions. The system currently in place faces several inefficiencies that hinder performance, productivity, and user satisfaction. These inefficiencies may include manual processes, redundant workflows, lack of automation, data inconsistencies, and difficulties in integration with modern technologies. The existing system does not meet the evolving needs of users, leading to delays, errors, and resource wastage. Furthermore, the scalability of the current system is limited, restricting its ability to accommodate growing demands. To address these issues, a comprehensive system analysis is required to assess the existing system, identify limitations, and propose an optimized solution that enhances efficiency, accuracy, and overall performance.

3.1 Existing System

The current system relies heavily on conventional methods and outdated technologies, which may include manual, semi-automated, or standalone digital processes depending on the project context. Typical practices involve the use of paperwork, spreadsheets, and isolated software tools that lack integration and centralized data management. This fragmented structure often results in data duplication, inconsistencies, and increased security vulnerabilities. Due to the absence of automated workflows and a unified platform, the system demands significant human intervention, leading to frequent errors, reduced efficiency, and operational delays.

3.2 Limitations of the Existing System

The limitations of the existing system significantly impact its overall performance and usability. Manual processes introduce inefficiencies and are prone to human error. Fragmented communication across uncoordinated channels leads to miscommunication and missed updates. The lack of modern digital features such as real-time notifications, event scheduling, and feedback collection restricts user engagement and responsiveness. Additionally, weak or outdated security measures pose a risk to sensitive user data, making the system unreliable for secure and scalable operations.

IV. PROPOSED SYSTEM

The proposed system is a web-based solution designed to overcome the inefficiencies of the existing manual event management process. The current system lacks automation, making event registration, notifications, and coordination challenging. CEMS streamlines event management by providing secure authentication, automated event registration, real-time notifications, and an interactive dashboard for both students and administrators. The system ensures seamless communication, efficient event tracking, and improved student engagement. Key features include role-based access control, automatic event status updates, feedback collection, and advanced report generation, making event organization more structured and hassle-free. By integrating technology into event management, CEMS enhances efficiency, reduces errors, and provides a centralized platform for better coordination and participation.

4.1 Theoretical framework

The Admin Module is responsible for managing events. Administrators can create new events, modify existing ones, and delete outdated events. They also have control over
participant lists and event schedules. Once an event is created or updated, the system ensures that the information is stored in the database and notifications are sent to students. This ensures that students are always informed about upcoming events and any changes made to them.

The Backend Module, built using Node.js, serves as the central processing unit of the system. It handles all communication between the frontend and the database. It processes event creation requests from admins, retrieves event details for students, manages user authentication, and ensures that notifications are delivered through Firebase Cloud Messaging. This module ensures that all data is synchronized and accessible to users in real time.
Database Module

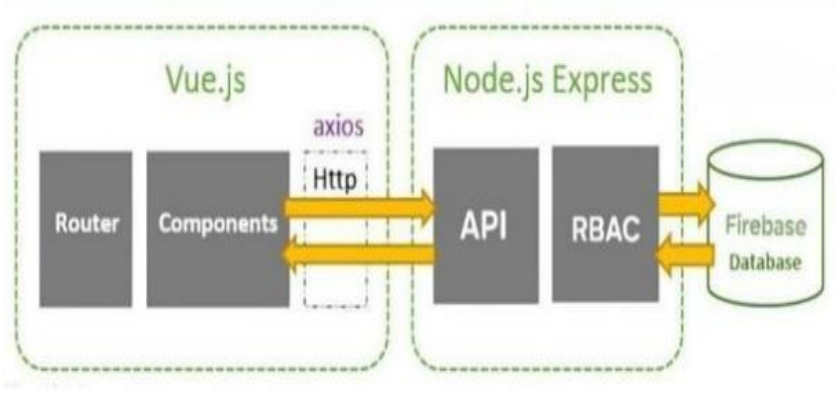
The Database Module is implemented using Firebase, a NoSQL cloud database. This module stores all event-related information, including event names, descriptions, dates, times, and locations. It also keeps track of student registrations and notification logs. The use of Firebase enables real-time data updates, ensuring that students and admins always have access to the latest event information. The database is designed to be highly scalable and secure, allowing it to handle a growing number of events and participants efficiently.

The User/Student Module provides students with an interface to interact with the system. Students can browse upcoming events, register for those they are interested in, and receive automatic notifications about event updates. Notifications help students stay informed about event changes, ensuring they do not miss out on important updates.

The Notification Module is designed to enhance user engagement and ensure timely communication through two core features: Push Notifications for Reminders and Email Confirmation. Push notifications deliver instant alerts to users about important tasks, upcoming deadlines, or system-related actions, helping to reduce missed activities while improving user productivity and responsiveness through real-time interaction. Meanwhile, the email confirmation feature automatically generates and sends confirmation messages following key user actions such as registrations, transactions, or bookings. This not only provides immediate acknowledgment but also reinforces transparency and builds user trust by confirming successful interactions within the platform.

The Feedback Module facilitates the collection and analysis of user input to enhance service quality and user satisfaction by implementing post-event feedback collection mechanisms. Users are prompted to provide feedback immediately after attending an event or interacting with the system. The module utilizes surveys, rating scales, and open comment fields to capture both qualitative and quantitative data, allowing for a comprehensive evaluation of user experiences. This structured feedback process supports data-driven decision-making and continuous improvement of future events and system interactions.

4.2 System Architecture



The system architecture is composed of three main layers: frontend, backend, and database, each designed to efficiently handle specific functionalities. The frontend is developed using Vue.js, featuring key components such as a router that manages navigation and page transitions between views like event listings, registrations, and user profiles. The user interface is built with modular Vue components that include forms for event creation, registration, notifications, and user management. Data communication on the frontend is handled using Axios, which sends HTTP requests to the backend API whenever users interact with the system, such as registering for events.

The backend is implemented with Node.js and Express.js, responsible for business logic, authentication, and API endpoint management. It exposes RESTful endpoints that process requests from the frontend, including event creation, user authentication, and notification delivery. A Role-Based Access Control (RBAC) system enforces user permissions, distinguishing between admins—who can create and manage events—and students, who can register for events and receive notifications. When API requests require authentication or authorization, the RBAC module validates user roles before granting access.

At the data layer, the system utilizes Firebase as its database, which securely stores and manages user information (such as names, emails, and roles), event details (including titles, dates, and descriptions), registrations linking users to events, and notification data like updates and reminders. The backend communicates with Firebase through the Firebase SDK, ensuring efficient and secure data retrieval and updates.

Overall, data flows seamlessly between the Vue.js frontend and the Node.js backend via Axios HTTP requests, while the backend interacts with Firebase to maintain consistent, real-time data management, supporting a robust and responsive event management platform

V. RESEARCH METHODOLOGY

The system adopts a modular approach, with the frontend developed in Vue.js to ensure a responsive and dynamic user interface, while the backend is implemented using Node.js and Express.js to manage business logic and API communication. Emphasis is placed on real-time interaction, secure data handling, and role-based access control to deliver a scalable, efficient, and user-friendly application.

4.1 Front end of the system

The frontend of the application is built using Vue.js, a progressive JavaScript framework. It utilizes Vue Router for managing navigation and page transitions, ensuring seamless access to different pages such as event listings, event registration, and user profiles. The user interface is composed of modular Vue components, including forms for event creation, event registration, notifications, and user management. These components communicate with the backend API using Axios, an HTTP client, to send and receive data.

When a user interacts with the system, for example, by registering for an event, an HTTP request is made to the backend. This request is processed by the backend API, which handles the necessary business logic and data operations.

4.2 Back end of the system

The backend is developed using Node.js and Express.js, a minimal and flexible Node.js web application framework. Express.js handles the routing and business logic of the application. The backend exposes RESTful API endpoints that the frontend communicates with. These endpoints handle various operations such as event creation, user authentication, and notifications.

A key feature of the backend is the implementation of Role-Based Access Control (RBAC). This module manages user roles, such as Admin and Student, to control access to different parts of the system. Admins have the ability to create events, while students can register for events and receive notifications. When a request is made to the backend, the RBAC module checks the user's permissions before granting access to the requested resource.

V. PERFORMANCE EVALUATION

Performance testing ensures that the system can handle expected loads without degradation. It evaluates speed, scalability, and stability under varying conditions. Load Testing is used to measure system response under normal and peak loads. Stress Testing is used to determine system behavior under extreme conditions. Scalability Testing is used to test how the system adapts to increased traffic. Database Performance Testing is used to ensures database queries run efficiently. Tools used are JMeter and Apache Bench for load and stress testing. Google Lighthouse are used for frontend performance analysis

Metric	Expected Result	Actual Result
Page Load Time	<3 sec	2.5 sec
Concurrent Users	1,000 users	1,200 users
API Response Time	<500 ms	450 ms
Database Query Execution	<100 ms	90 ms

.

V. CONCLUSION

The successful implementation of this system demonstrates the practical benefits of using modern web development technologies to build a secure, scalable, and efficient event management system. By integrating Vue.js for a responsive and user-friendly frontend, Node.js for a high-performance backend, and Firebase for real-time data synchronization, the system improves both workflow efficiency and user experience. Security has been a core focus, with features such as user authentication, data encryption, secure API communication, and role-based access control ensuring that sensitive information remains protected. The use of OAuth authentication and encrypted data storage reduces the risk of security breaches and builds user trust. Overall, this system highlights how combining front-end frameworks, server-side technologies, and cloud-based services can create a powerful solution for digital event management. It serves as a model for how technology can streamline processes, enhance communication, and improve engagement in educational and organizational environments

VI. FUTURE ENHANCEMENTS

To enhance the functionality and performance of the College Event Management System (CEMS), several advanced features and architectural upgrades are planned for future implementation:

1. AI-Powered Feedback Analysis: By integrating machine learning algorithms, the system will automatically analyze user feedback to detect trends, evaluate event effectiveness, and offer predictive insights for better future planning. Sentiment analysis will further categorize responses as positive, neutral, or negative, enabling event organizers to make informed decisions and improve event quality and relevance.

2. Personalized and Smart Notifications: Users will have the ability to customize their notification preferences, including the frequency, type, and priority of alerts. Additionally, AI-driven smart reminder systems will predict the optimal times to send notifications, maximizing user engagement and increasing participation rates.

3. Cloud-Based Architecture for Scalability and Reliability: Transitioning to a cloud-native infrastructure using services like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP) will offer improved scalability, dynamic load balancing, and high availability. This shift will ensure the system maintains real-time responsiveness during peak usage, while enhancing fault tolerance, disaster recovery, and data security through cloud-managed solutions.

REFERENCES

[1] V. Mishra, M. Dubey, P. Banarjee, A. Jumle, P. Raipure, and P. Wankhede, "Event Management Systems (EMS): A Web-Based Approach," International Journal of Trend in Research and Development (IJTRD), vol. 9, no. 4, pp. 50-57, 2024.
[2] S. M. Reza, M. R. Khosravi, and M. K. Moghimi, "A survey on security in web-based event management systems," International Journal of Computer Applications, vol. 180, no. 7, pp. 25–31, Jan. 2024.
[3] J. H. Wei, Y. Zhao, and Z. Zhao, "Automated notification systems in event management: A study of real-time communication," Computer Science Review, vol. 20, no. 5, pp. 101–110, 2023.
[4] A. Rahman, S. Gupta, and M. Singh, "Cloud-Based Event Management System with Real-Time Updates," Journal of Computer and Network Security, vol. 15, no. 3, pp. 123–135, 2023.
[5] M. Perez, A. C. Kot, and A. Rocha, "Detection of real-world event scheduling conflicts in online platforms," in Proc. IEEE Int. Conf. on Software Engineering, May 2023.