E-Vents: A Unified Mobile Platform for College Events Hosting and Registration

Sarvesh Sawant, Om Shinde, Swayam Naikude, Nikhil Patil, and Dr. Sheetal Patil

Vidyalankar Institute of Technology, Wadala, Mumbai-400037

DOI: https://doi.org/10.51244/IJRSI.2026.1306000316

Received: 18 June 2026; Accepted: 23 June 2026; Published: 09 July 2026

ABSTRACT

Event hosting in academic institutions often faces problems because of broken communication channels, manual registration processes, and a lack of real-time interaction. These issues lead to lower student participation, poor logistical management, and a heavier workload for organizers. To tackle these challenges, this paper introduces E-Vents, a mobile event management app that works across platforms. It is designed to simplify college event workflows, from creation and registration to notifications and tracking participation. E-vents has a combination of different technologies, which includes the usage of Flutter (Dart) at the frontend, Node.js at the backend that enhances the performance, MongoDB for efficient database management, and Firebase Cloud Messaging (FCM) for the latest event updates and notification alerts. This system provides three different roles: administrator, organizer, and user, each having unique access to the system, such as access to the entire system, hosting events, participation, etc. We have chosen Flutter because it provides a cross-platform operation with having a single codebase, which enables a user-friendly experience irrespective of the device. A pilot deployment involving several university departments was carried out to confirm the platform's performance and usability. Surveys and backend analytics were used to get input from the event planners and the students who attended. In comparison to traditional systems, 87% of users reported a more streamlined experience, and the results showed a 90% increase in registration efficiency. E-Vents provides a strong solution to update college event management by combining scalable backend infrastructure, real-time messaging, and mobile-first design. This paper adds to the larger conversation on digital transformation in educational event administration by describing the platform's system design, component integration, user feedback, and overall impact.

Keywords: Mobile Event Management, Cross-Platform Application, E-Vents, Flutter, Dart, NodeJS, MongoDB, Firebase Cloud Messaging (FCM), Academic Institutions, College Events, Event Workflow Automation, Real-time Notifications, Student Participation, Logistical Management, Digital Transformation, Educational Administration Hybrid Architecture, User Roles

INTRODUCTION

Academic institutions function as dynamic centers where numerous seminars, workshops, festivals, and clubs intersect to weave a complex tapestry of campus experiences. A stronger campus community emerges through these events, which boost student life and foster interdisciplinary collaboration. Even with digital learning advancements, our methods for organizing, sharing, and tracking participation remain outdated. A multitude of institutions depend on an intricate combination of tangible flyers, informal group chats, sporadic spreadsheet collections, and occasional Google Forms. These methods appear straightforward to deploy, yet they soon demonstrate their limitations by failing to match demand growth while leaving data vulnerable and not achieving the necessary modern event flow efficiency. The absence of unified management systems combined with slow communication channels and manual registration methods creates numerous challenges for students and event coordinators. A multitude of participants fail to seize opportunities because they remain unaware of announcements due to poor discoverability and insufficient reminder systems. Event organizers find themselves overwhelmed by the need to handle numerous disconnected responsibilities such as attendance tracking, participation verification, update dissemination, and turnout analysis without the aid of a unified system. The presence of these inefficiencies diminishes student engagement while simultaneously creating substantial logistical burdens, especially for large-scale events and those involving multiple departments. A variety of

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

commercial event management tools exist, but the majority remain excessively generalized or specifically designed for corporate functions and ticketed events. The systems frequently miss essential campus-specific functionalities, including distinct role-based access controls for students and faculty members alongside cross-device operability and institutional branding elements. The combination of financial limitations and technical restrictions renders these platforms impractical for use in student-led projects or smaller educational institutions. We have created this online unified mobile platform called E-Vents by keeping in mind the event hosting concerns that are faced by different colleges. This platform is built by using the Flutter (Dart) framework for the frontend to have both Android and iOS functionality in one single codebase, ensuring a smooth and consistent experience. Having Node.js and Express.js in the backend with RESTful APIs, MongoDB has been used as our cloud-based NoSQL to store user details, event details, and registration status. To keep all the participants updated, we have integrated Firebase Cloud Messaging (FCM) for instant and reliable notifications. Our system is designed with three different roles: administrator, organizer, and user. Each role has different access to the portal according to its needs. The user has access to their personal dashboard, where all the upcoming events will be listed. The user will also receive all real-time notifications regarding the event. Organizers will have access to the dashboard in such a way that enables them to create any event by entering all the details, which will be further verified, and also have access to push notifications and registration metrics. For the verification process we will have our major role, which is the administrator role, which will have access to the whole system with access to verification, managing permissions, and maintaining system integrity.

Related Work

The increasing demand for digitized campus operations has led to the development of various event management systems aimed at streamlining student participation and reducing administrative workload. However, most existing platforms either cater to corporate use cases or lack the real-time interactivity, scalability, and role-based workflows required in academic settings.

1. CampusConnect was proposed as a mobile-first event platform for academic institutions, providing basic event hosting and scheduling features (Nandakishore & Abraham, 2025). While the system addressed some communication issues, it did not integrate real-time messaging or push notification services and lacked the cross-platform flexibility provided by modern frameworks like Flutter.

2. Cardinal Connect focused on student organization events and offered basic registration and management features. However, it was implemented as a web-only solution, relying on static user interfaces without support for real-time user engagement or mobile app deployment (Blancaflor & Dela Cruz, 2021).

3. Eventlyst, centralized campus-wide event listings and notifications to improve visibility and participation. However, it did not offer QR-based check-ins, custom workflows per user role, or support for instant updates using cloud messaging protocols like Firebase Cloud Messaging (Bhaalkar et al., 2024).

4. Feliza and Ammadienta (2024) introduced an event platform that included integrated ticketing and limited networking features. Though innovative in some respects, their solution was designed primarily for broader public events rather than academic-specific use cases and lacked the institutional role control needed for college events (Feliza & Ammadienta, 2024).

Thinking from a technical perspective, there were many attempts made for the development of a single device platform that lacked cross-device functionality and real-time responsiveness. E-vents Make use of the Flutter (Dart) framework to bridge the gap between Android and iOS devices. Having NodeJS and MongoDB at the backend, it provides better performance and scalability. Firebase Cloud Messaging (FCM) provides an edge by providing real-time updates.

Implementation

The development of E-vents follows a full-stack architecture, which is designed in such a way that it provides a real-time cross-platform mobile experience for college event management. The system is built by using the

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Flutter (Dart) framework for the frontend, Node.js with Express.js for the backend and API services, MongoDB for database management, and Firebase Cloud Messaging (FCM) for real-time push notifications for user engagement. Each layer of application provides scalability, responsiveness, and efficient flow of data between user and system.

- **Cross-platform development (Flutter with Dart)**

The frontend interface of E-vents is made using Flutter (Dart), a UI toolkit provided by Google, which enables the development of cross-device platforms that function with both Android and iOS. This framework has the following features:

1. *Cross-Platform Compatibility*: One single code runs on both Android and iOS, ensuring the same user experience.

2. *User Role Interface*: The system is built for three different roles: administrator, organizer, and user, which can be implemented by distinct UI logic.

3. *Event Browsing and Registration*: The system provides the user with different filter options so that the user can efficiently register for an event based on the choice from the mobile device directly.

4. *QR Code Generation*: After registration the user will receive a dynamically generated QR Code, which will be further used for verification purposes.

5. *Firebase Integration*: The organizer will be able to push event alerts in the form of notifications with the help of Firebase Cloud Messaging (FCM).

- **Backend Implementation (Node.js + Express.js)**

The backend provides a RESTful API structure by the use of Node.js + Express.js. This structure is used to connect the mobile frontend with the MongoDB database.

1. *Authentication and Role-Based Access*: JWT (JSON Web Tokens) is used to authenticate users and secure endpoints between the three roles.

2. *Event Management API*: The organizers will have the access to create, update, or delete an event. The API will retrieve the event data for users and admins.

3. *QR Code Generation and Validation*: Using the npm package, a QR Code will be generated after successful registration of an event, which will be stored with registration details.

4. *Notification Triggers*: Backend logic will be implemented in such a way that it will be connected with Firebase Admin SDK, which will allow pushing notifications regarding the event.

- **Database Management (MongoDB)**

MongoDB is used to handle the database in the backend, which is hosted by using MongoDB Atlas, which provides cloud-based access. It used document-based schema for:

1. *User*: Stores all the user information, including role, profile, and login credentials.

2. *Events*: Stores all the event information, like title, date, venue, etc.

3. *Registrations*: It provides a linkage between the users and events and also the QR data for verification.

4. *Notifications*: Manages logs of notifications using Firebase authentication.

- **Real-Time Notification with Firebase**

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Firebase Cloud Messaging (FCM) is used in the system so that the organizer will be able to push real-time notifications and alerts.

1. *Push Notifications*: The organizer sends push notifications to the user regarding the events.

2. *Integration with Flutter*: The Firebase plugin enables the Flutter apps to receive notifications and messages even if the app is running in the background.

3. *Admin and Organizer*: Backen triggers notifications to the organizers for new registrations or cancellations.

- **Security, Hosting and Deployment**

1. *Security Measures*: To have a secured system, passwords are hashed using bcrypt, and the API will be secured by using middleware for authentication and role verification.

2. *Frontend Deployment*: This Flutter app will be deployed to both the Google Play Store and the App Store.

3. *Backend Deployment: The* backend server will be hosted either on Render or Heroku with MongoDB in MongoDB Atlas itself.

4. *Monitoring*: The monitoring will be handled using the Firebase Console and server logs for performance optimization and error tracking.

Architecture

- **Main Architecture**

The architecture of E-Vents is structured as a modular, service-oriented full-stack application designed for real-time event management in college environments. It adopts a client-server model, leveraging Flutter (Dart) for the cross-platform mobile frontend, Node.js with Express.js for backend API services, MongoDB Atlas as the cloud database, and Firebase Cloud Messaging (FCM) for instant user notifications.

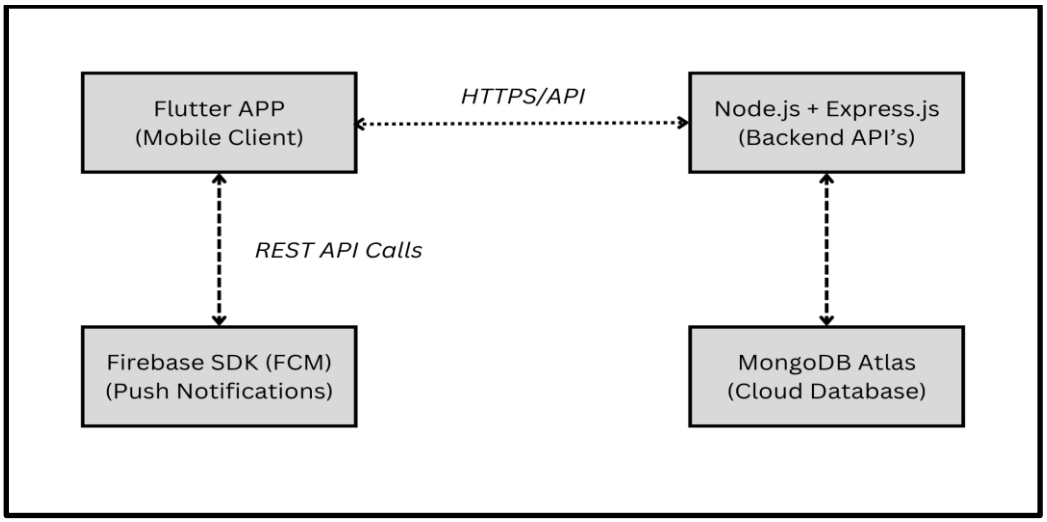

Figure 1: Main Architecture

Front-end Flow

High-Level Architecture

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

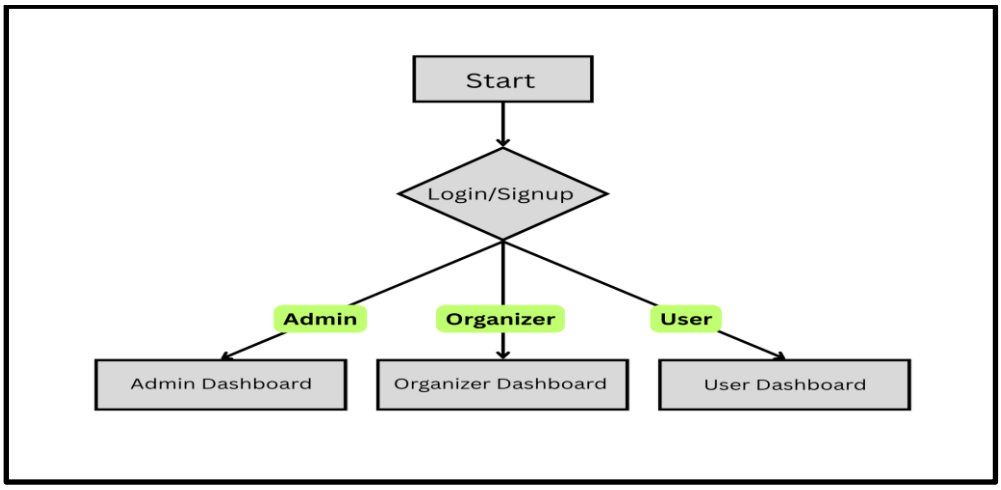

Figure 2: High-Level Overview

Administrator Architecture

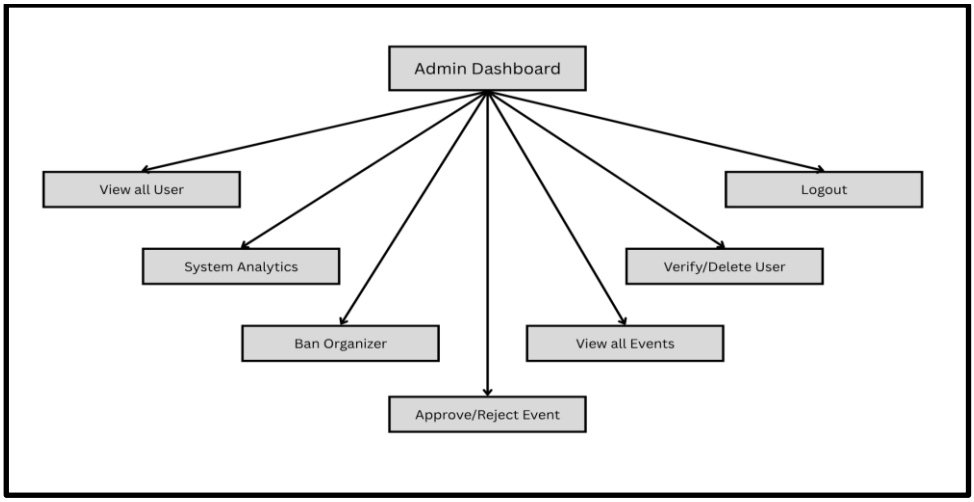

Figure 3: Administrator Overview

Organizer Architecture

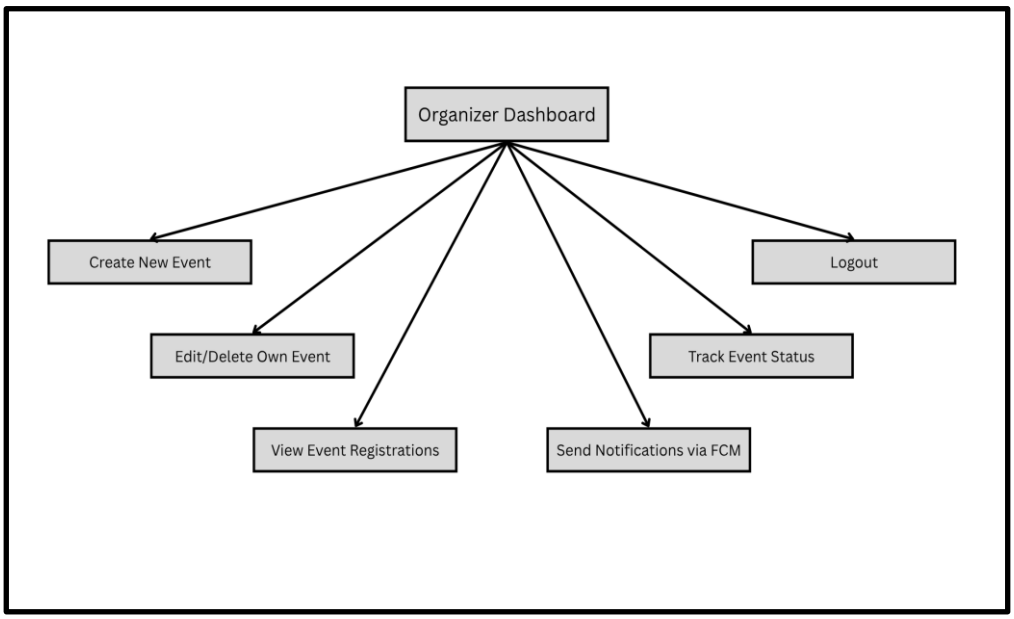

Figure 4: Organizer Overview

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

User Architecture

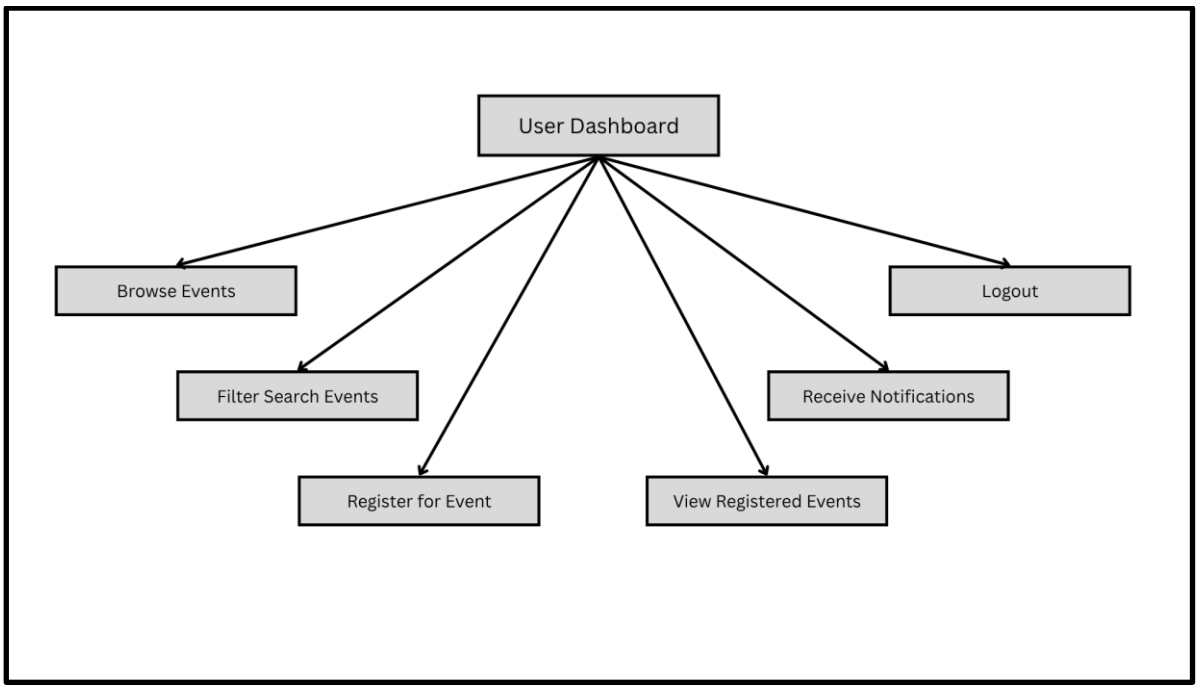

Figure 5: User Overview

Back-end Flow

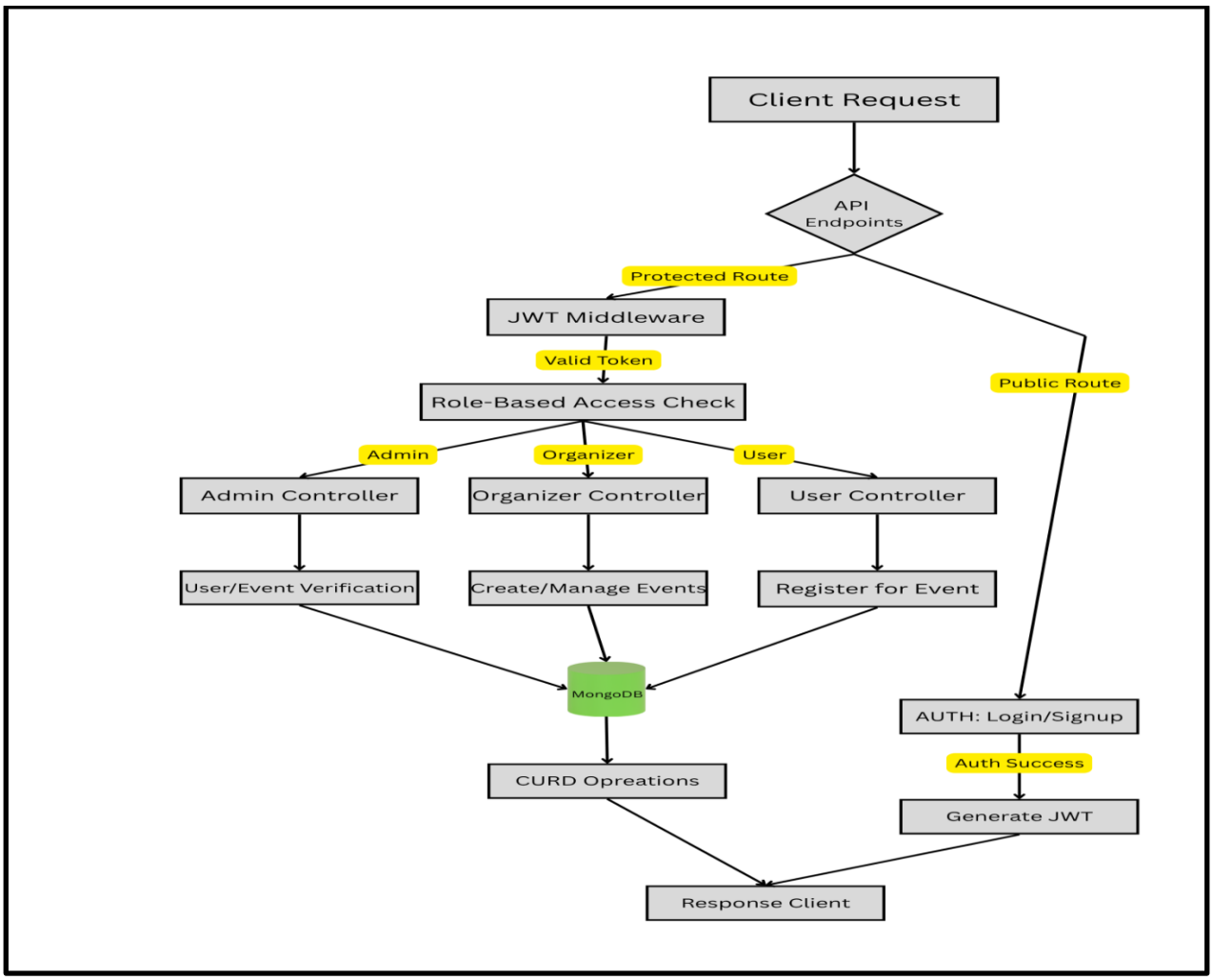

Figure 6: Backend Overview

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

MongoDB Flow

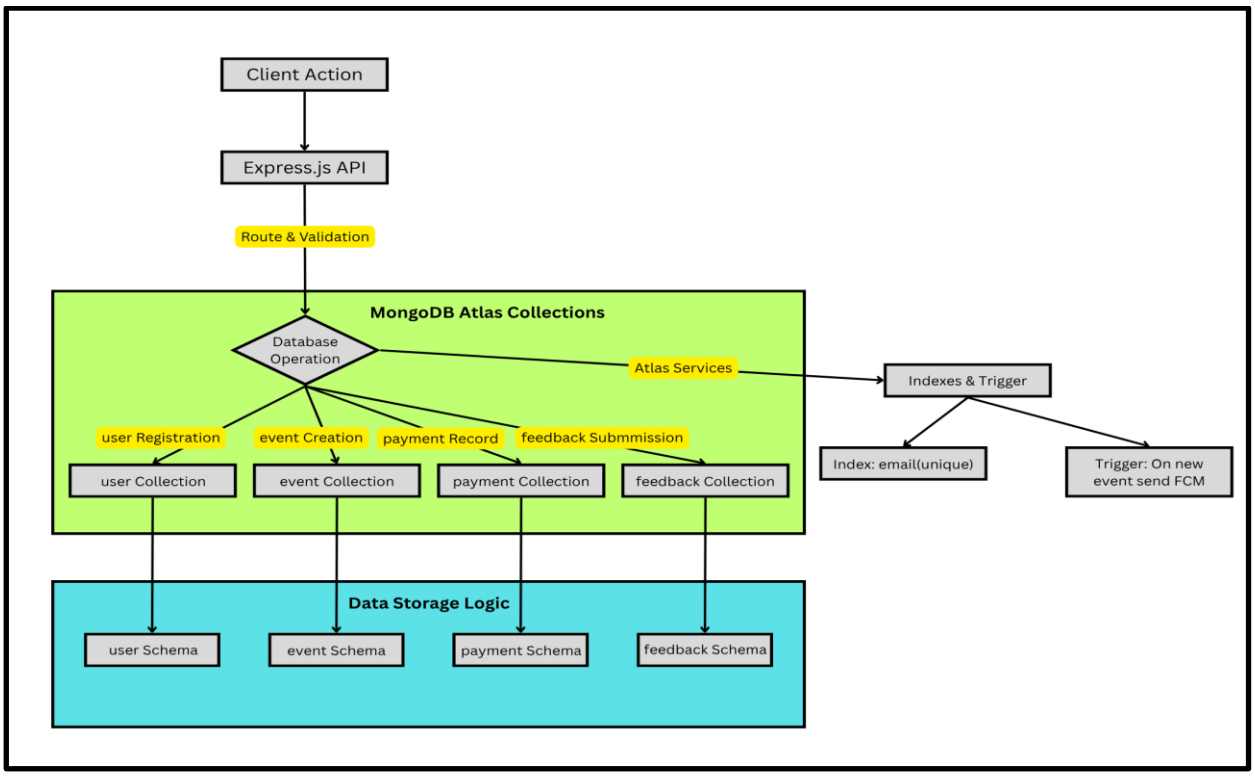

Figure 7: MongoDB Architecture

Firebase Integration

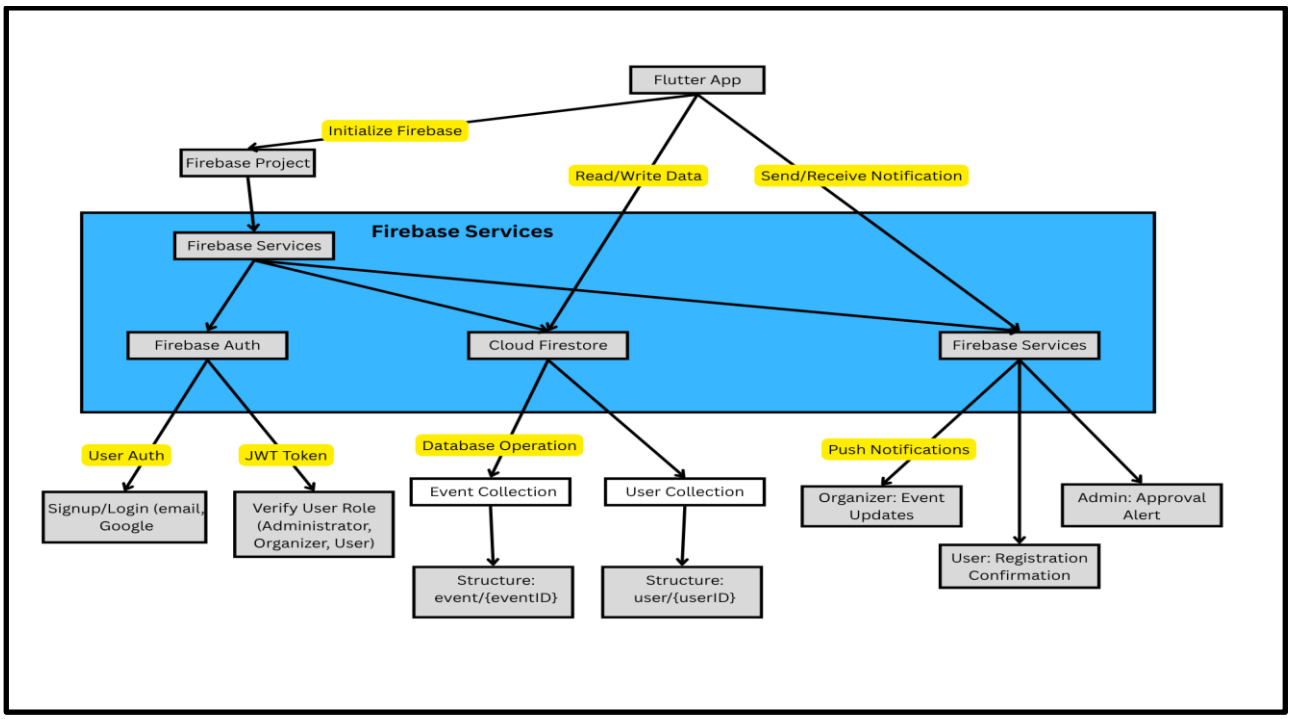

Figure 8: Firebase Architecture

Future Scope

The E-Vents platform, in its present working, has several areas of potential development in the future. A dedicated web portal for the administrator/organizer would increase user experience on larger displays, especially for the use of processing high-volume events, as well as the ability to see in-depth analysis. Merging

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

event analysis and insights would facilitate data-driven decisions by ascertaining the participants' trends, registration surges, and in-demand event types. Future releases can also interface with the likes of an SIS or an LMS' academic system in order to automate user authentication and facilitate credit-based tracking of events. Introducing machine learning would enable the user experience to be customized by suggesting events based on a participant's history of interests and of registrations. Including support for offline display of the QR codes and caching of events would add reliability during use on campus, where network access might be restricted. Conversely, the problem of scalability can be solved by transforming the E-Vents platform into a multi-institutional platform with subdomains, alongside permissions related to roles, university, or college. Further in the future, blockchain-related certification can be explored in order to offer tamper-proof digital professional or academic event attendees' credentials.

CONCLUSION

In this work, we present E-Vents, a multiplatform mobile application developed using Flutter (Dart) as the frontend and Node.js, MongoDB, and Firebase making up the backend stack. Designed with college life in view, E-Vents disrupts the chief limitations of typical event management systems by offering a centralized, real-time, and user-friendly solution for hosting, signing up, and managing events on the campus. The system provides a safe, easy-to-use experience for each stakeholder based on their specific needs. Real-time event listing, QR code check-in, and push notifications through Firebase are just a few of the features that make it easier to communicate, less work for administrators, and more participation from students. Pilot deployment demonstrated greatly enhanced registration speed and participant experience, highlighting the system's practical advantage. Organizers benefited from the utilization of automated messages and real-time participant observation, while the students preferred the friendly interface and direct accessibility of the events. With the aid of a modern technology stack and mobile-driven design, E-Vents becomes a scalable, streamlined, and impactful instrument of college events digitalization. Future evolution of the platform may take further strides in the fields of high-end analytics, machine learning, and multi-institution scale, further enhancing its potential status as a unified ecosystem of academic event engagement.

REFERENCES

1. A. Nandakishore and J. T. Abraham, “CampusConnect: A Digital Solution for Campus Event Management,” *Project Report*, 2025. [Online]. Available: http://136.232.36.98:8080/xmlui/bitstream/handle/123456789/6430/CampusConnect.pdf
2. E. B. Blancaflor and G. A. B. Dela Cruz, “Cardinal Connect: A Student Organization Events Management System,” *Proc. ACM*, 2021. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3483816.3483835
3. R. R. Bhaalkar, S. Kumari, and S. Vyas, “Eventlyst: A Centralized Platform for Improving College Event Awareness and Participation,” *Int. J. of Comp. Sci. & Mobile Comput.*, vol. 13, no. 2, 2024. [Online]. Available: https://search.ebscohost.com/login.aspx?AN=181714907
4. D. S. P. Feliza and M. Ammadienta, “Innovative Event Management Platform Design with Integrated Ticketing and Networking Features,” *JDBIM Journal*, 2024. [Online]. Available: https://ejournal.unesa.ac.id/index.php/jdbim/article/view/60424
5. P. Pabba, B. Viswadha, and P. Srivani, “Mobile Application for College Event Management,” *IRJET*, vol. 9, no. 12, 2022. [Online]. Available: https://www.academia.edu/download/96242777/IRJET_V9I1237.pdf
6. Firebase, “Firebase Cloud Messaging Documentation,” *Google Developers*, 2024. [Online]. Available: https://firebase.google.com/docs/cloud-messaging
7. MongoDB, “MongoDB Atlas: Cloud Database,” *MongoDB Inc.*, 2024. [Online]. Available: https://www.mongodb.com/atlas
8. Flutter, “Flutter: Beautiful Native Apps in Record Time,” *Flutter.dev*, 2024. [Online]. Available: https://flutter.dev/
9. Node.js, “Node.js Documentation,” *OpenJS Foundation*, 2024. [Online]. Available: https://nodejs.org/en/docs