

Mern Event Management with Firebase

**M V Ramana¹, P. Mounika², T. Indu Priya³, P. Dora Babu⁴,
M. Jayendra Karthik⁵, D. Charan Raj⁶**

¹Assistant Professor, Department of Computer Science and Engineering, VSM College of Engineering, Ramachandrapuram, Andhra Pradesh, India

^{2,3,4,5,6} UG Students, Department of Computer Science and Engineering, VSM College of Engineering, Ramachandrapuram, Andhra Pradesh, India

ABSTRACT

In the digital era, event management has evolved with the integration of online platforms, enabling seamless organization, registration, and payment processes. This project introduces "Planorama," an online event management system built using React.js and Firebase, designed to streamline event creation, user participation, and secure transactions via QR-based UPI payments. Leveraging Firebase's real-time database and authentication services, Planorama addresses challenges such as manual event coordination and payment verification by offering an automated, scalable solution. The system employs a modular architecture with machine learning potential for future enhancements, such as event recommendation or fraud detection. This paper explores the system's design, implementation, and performance, demonstrating its efficacy in enhancing user experience and administrative efficiency.

1. INTRODUCTION

The rise of online social platforms has transformed how events are organized and attended. Traditional event management often involves labor-intensive processes, including manual registrations and cash-based payments, leading to inefficiencies and errors. Planorama aims to mitigate these issues by providing a centralized digital platform where users can explore events, register seamlessly, and pay securely. Built with React.js for a dynamic frontend and Firebase for backend services, the system ensures real-time data synchronization and robust user authentication. Key features include a user-friendly landing page, an admin panel for event oversight, and a payment module integrated with QR codes. This paper details the system's architecture, implemented functionalities, and potential for scalability, positioning it as a modern solution to event management challenges.

2. LITERATURE REVIEW

Throughout the development of our project, we were able to review some projects, Journals, Articles and books which are related to the title of our project. We believe that all the reviewed materials have been a good asset for the overall design and development of the project we have chosen. In this section some of the related projects we have reviewed are discussed.

2.1 Online Event Management System

Aditi Gupta, Piyush Yadav This paper presents a web-based solution for organizing events online, covering functionalities like user registration, event listing, and admin control. The system lacks advanced UI and integration, which PLANORAMA addresses with a modern stack and UPI payments. — Title: A

Review on Event Management Systems Using Web Applications Authors: S. Priya, M. Rajeshwari The authors discuss various techniques used in web-based event management. They emphasize the need for mobile responsiveness and efficient database handling. PLANORAMA builds upon these ideas by integrating MongoDB and React for real-time, responsive interaction. Aditi Gupta, Piyush Yadav This paper presents a web-based solution for organizing events online, covering functionalities like user registration, event listing, and admin control. The system lacks advanced UI and integration, which PLANORAMA addresses with a modern stack and UPI payments.

2.2 Project Overview

PLANORAMA is a modern web-based Event Management System developed using the Semi-MERN stack, including React.js, Node.js, Express.js, and Firebase Firestore, with Tailwind CSS for responsive design. The system addresses challenges found in traditional event management such as manual registrations, lack of accessibility, and inefficient payment methods. PLANORAMA allows administrators to create, update, and delete events, while users can easily browse events, register, and make secure UPI-based QR payments. It offers real-time updates, automated registration confirmations, and a digital platform for storing event history and images. The application is designed with scalability in mind, supporting multiple events and users simultaneously. It features both admin and user dashboards for smooth interaction. Security is ensured through encrypted passwords and role-based access controls. The system is accessible on any device, making it highly portable and user-friendly. PLANORAMA simplifies the event management process and enhances the experience for both organizers and participants. Overall, it serves as an efficient, reliable, and scalable solution for managing a wide range of events.

3. EXISTING METHOD

Current event management systems frequently lack integration with contemporary online platforms and instead rely on stand-alone apps or manual procedures. Online registration and payment options are offered by tools like Eventbrite and Meetup, but they might not provide flexible participant management or real-time admin control like Planorama does. Prior studies, like the one conducted by Conroy et al. (2015) draws attention to the shortcomings of simple web tools in managing intricate workflows, which frequently call for third-party integrations for notifications or payments. Additionally, these systems have trouble with real-time updates and scalability, which are essential in dynamic event environments. By integrating Firebase's real-time features and a personalized admin dashboard, Planorama expands upon these frameworks and overcomes the static nature of many other solutions.

Rather than being integrated with contemporary online platforms, existing event management systems frequently rely on stand-alone apps or manual procedures. Though they might not offer flexible participant management or real-time admin control like Planorama, tools like Eventbrite or Meetup offer online registration and payment options. Conroy et al.'s earlier study, for example. (2015) emphasizes how simple web tools are inadequate for managing intricate workflows, frequently necessitating third-party integrations for notifications or payments. Scalability and real-time updates, which are essential for dynamic event environments, are other issues with these systems. Beyond the static nature of many current solutions, Planorama expands upon these foundations by integrating Firebase's real-time capabilities and a custom admin dashboard.

3.1 Manual Registrations

Human error is more likely when participants register using paper forms.

3.2 Offline Promotions

Posters, banners, or word-of-mouth are used to advertise events, which restricts their audience.

3.3 Low Scalability

Managing attendees and resources becomes disorganized and ineffective as event sizes grow.

4. PROPOSED SOLUTION

PLANORAMA presents an entirely digital, centralized, and user-friendly platform to solve the inefficiencies in conventional event management. constructed with the Semi-MERN stack (Express.js, Firebase, React.js and Node.js) and Tailwind CSS, it guarantees seamless event management for both planners and attendees. Administrators can create, edit, and remove event listings using the system, which includes information about the date, time, venue, capacity, and description. A clear, responsive interface that works on PCs, tablets, and smartphones makes it simple for users to peruse upcoming events. One of the main features is the safe and easy UPI-based payment system that uses dynamic QR codes. Users can use apps like Paytm, Google Pay, and PhonePay to make payments. Third-party payment gateways are not necessary because users manually enter the transaction ID for admin verification after making a payment. Images, registration information, and event data are stored in Firebase Fire store, which guarantees scalability and effective data management. Tools for allocating volunteers and keeping track of registrations are also available to administrators. By giving immediate feedback on the status of payments and registrations, real-time updates and notifications improve the entire experience. For functionality and security reasons, the admin and user dashboards are clearly divided. Secure access is ensured by robust validation and authentication procedures, such as encrypted passwords and OTP-based login. Role-based access is supported by the system to stop unwanted access.

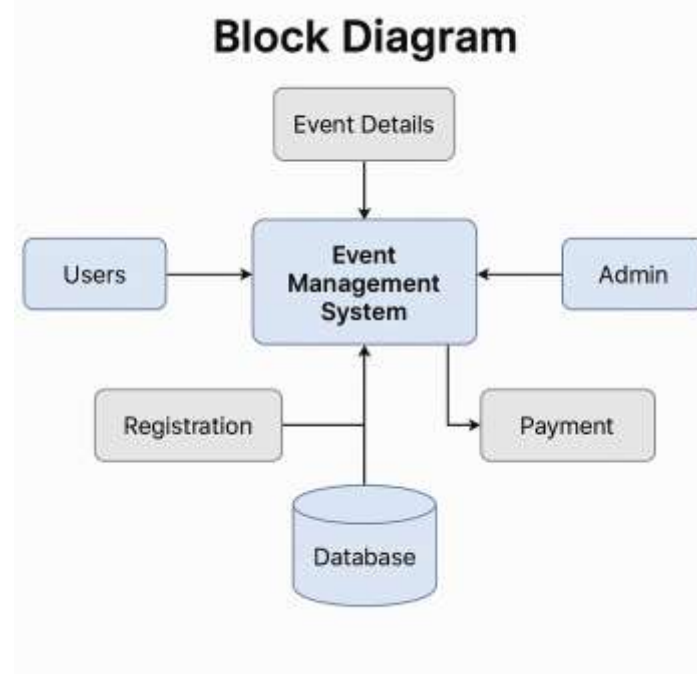


Fig.4.1.1: Block Diagram

5. Key Algorithms

5.1 Landing Page

// User Login

```
IF user selects Sign-in THEN
    INPUT email, password OR use Google Authentication
    IF credentials valid THEN
        REDIRECT to Home Page
    END IF
ELSE IF user selects Sign-up THEN
    CREATE new account with email, password OR Google Authentication
    IF account creation successful THEN
        REDIRECT to Home Page
    END IF
END IF
```

5.2 Home Page

```
IF user selects filter THEN
    UPDATE event list based on selected category
END IF
```

```
IF user clicks "View Details" THEN
    REDIRECT to Event Details Page with selected event ID
END IF
```

5.3 Event Details Page

```
IF user clicks "Register for Event" THEN
    DISPLAY Registration Form
    GENERATE input fields based on:
        - Minimum participants (pre-filled)
        - Maximum participants (allow add/remove)
    FOR each participant DO
        COLLECT:
            - Name
            - Email
    END FOR
END IF
```

```
IF user clicks "Preview" THEN
    DISPLAY summary of participant details
END IF
```

```
IF user clicks "Payment" THEN
    REDIRECT to Payment Page
END IF
```

5.4 Payment Page BEGIN DISPLAY

```
IF user scans QR code THEN
  PROCESS payment
END IF
```

```
IF user enters Reference ID THEN
  STORE Reference ID
END IF
```

```
IF user clicks "Payment Done" THEN
  VALIDATE Reference ID (optional)
  REDIRECT to Success Page OR Event Details Page
END IF
```

5.5 Success Page BEGIN DISPLAY

```
IF user clicks "Back to Home" THEN
  REDIRECT to Home Page
END IF
```

5.6 Admin Panel BEGIN

// Admin Panel Operations

DISPLAY:

- List of all events
- "Create Event" button

```
IF admin clicks "Create Event" THEN
  DISPLAY form with:
    - Event name
    - Description
    - Category (dropdown)
    - Minimum participants
    - Maximum participants
    - Amount
  IF form submitted THEN
    SAVE event to Firestore
    REFRESH event list
  END IF
END IF
```

```
IF admin clicks "Edit" on event THEN
  DISPLAY pre-filled form with event details
  IF form submitted THEN
    UPDATE event in Firestore
```

```
    REFRESH event list
  END IF
END IF

IF admin clicks "Delete" on event THEN
  DISPLAY confirmation dialog
  IF confirmed THEN
    DELETE event from Firestore
    REFRESH event list
  END IF
END IF
```

```
IF admin clicks "Logout" THEN
  CLEAR admin session
  REDIRECT to Admin Login Page
END IF
```

IMPLEMENTED MODULE

Frontend with React.js

React.js powers the system's UI, utilizing components like `Navbar.jsx` for navigation, `LandingPage.jsx` for public access, and `Home.jsx` for authenticated users. State management via hooks ensures dynamic rendering of event lists and forms.

BackendwithFirebase

`FirebaseConfig.js` initializes authentication (Google Sign-In, email/password) and Firestore for storing events, registrations, and payments. Real-time listeners in `Home.jsx` and `AdminPanel.jsx` fetch and update data instantly.

AuthenticationModule

`Login.jsx` and `Register.jsx` implement Firebase Authentication, supporting email/password and Google Sign-In, with `ProtectedRoute.js` ensuring secure access to user-specific pages.

PaymentIntegration

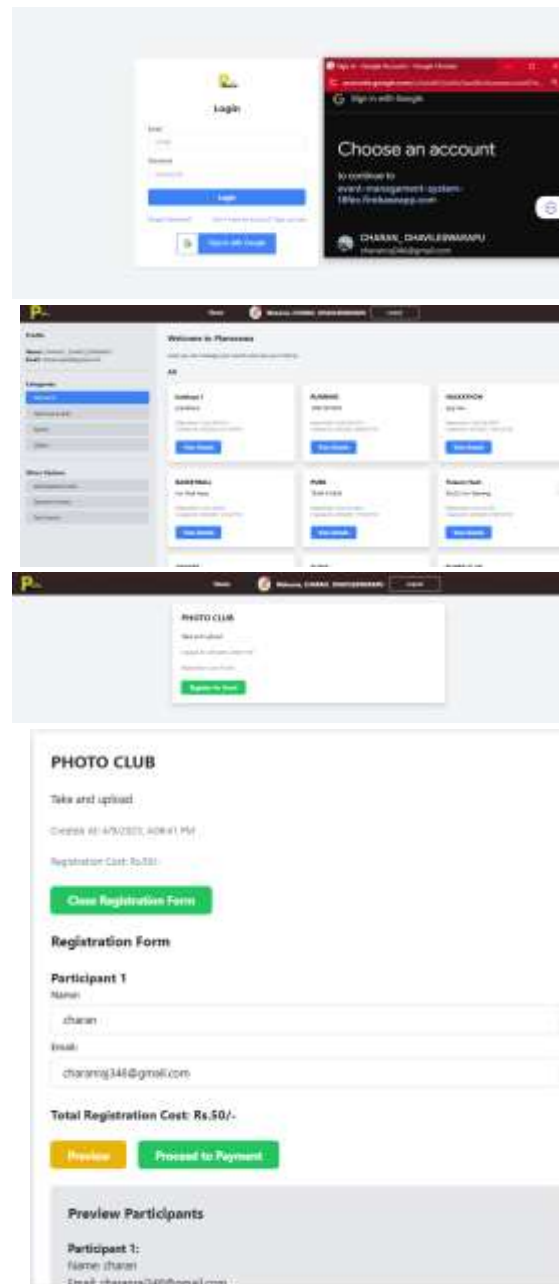
`PaymentPanel.jsx` facilitates QR-based UPI payments, storing transaction details in Firestore. `PaymentDetails.jsx` provides a history view, enhancing transparency.

SYSTEM ARCHITECTURE

1. **Frontend Layer:** React components (e.g., `LandingPage.jsx`, `EventCard.jsx`) render UI.
2. **Backend Layer:** Firebase handles authentication, database, and storage.
3. **Data Flow:** Events → Registrations → Payments, synced via Firestore.
4. **User Interaction:** Public users explore, authenticated users register, admins manage via `AdminDashboard.jsx`.

Fig: 1 System Architecture Diagram (Conceptual: React UI → Firebase → External Email Service)





The image displays three screenshots of the IJFMR event management system. The top screenshot shows a login page with a 'Login' button and a 'Forgot your password?' link. The middle screenshot shows a dashboard with a sidebar menu and a main content area with various event management options. The bottom screenshot shows a 'PHOTO CLUB' registration form with fields for Name, Email, and a 'Proceed to Payment' button.

PHOTO CLUB

Take and upload

Checkin At: 4/5/2023, 4:04:41 PM

Registration Cost: Rs.50/-

[Close Registration Form](#)

Registration Form

Participant 1

Name:

Email:

Total Registration Cost: Rs.50/-

[Review](#) [Proceed to Payment](#)

Preview Participants

Participant 1:

Name: charan

Email: charan344@gmail.com

Payment Page

Scan the QR code below to make the payment.

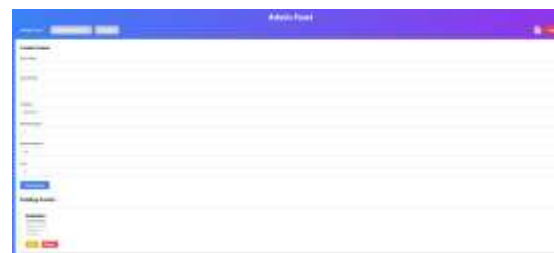


Total Registration Cost: Rs.50/-

Transaction ID:

T08643739368292

Complete Registration



CONCLUSION

Planorama offers a robust, user-centric solution for online event management, integrating real-time data, secure payments, and admin control. By addressing manual process inefficiencies, it enhances accessibility and trust for users and organizers alike. Future enhancements could include AI-driven event suggestions, multi-platform support (e.g., mobile apps), and advanced analytics, ensuring adaptability to evolving user needs.

REFERENCES

1. Conroy, N. J., et al. (2015). "Automatic deception detection: Methods for finding fake news." ASIS&T

Annual Meeting.

2. React.js Documentation. (2023). "Building User Interfaces."
3. Firebase Documentation. (2023). "Real-Time Database and Authentication."
4. Tacchini, E., et al. (2017). "Automated fake news detection in social networks." arXiv:1704.07506.