International Journal for Modern Trends in Science and Technology
Volume 12, Issue 05, pages 116-122
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue05.html
DOI: https://doi.org/10.5281/zenodo.19836506

An Integrated Event Data Management Platform Using Full Stack Python

Ch. Ganga Jahnavi, Ch. Divyanjali, Ch. Yamuna, Ch. Deepthi, Ch. Chaitanya Lakshmi, V. S. Sharief

Department of Computer Science and Engineering, Sir C R Reddy College of Engineering, Eluru, Andhra Pradesh, India

To Cite this Article
Ch. Ganga Jahnavi, Ch. Divyanjali, Ch. Yamuna, Ch. Deepthi, Ch. Chaitanya Lakshmi & V. S. Sharief (2026). An Integrated Event Data Management Platform Using Full Stack Python. International Journal for Modern Trends in Science and Technology, 12(05), 116-122. https://doi.org/10.5281/zenodo.19836506

Article Info
Received: 28 March 2026; Revised: 24 April 2026; Accepted: 26 April 2026.

KEYWORDS

Full Stack, Python, Web Development, REST API, Database, Django, React.js, Event Management

ABSTRACT

EventHub is a comprehensive, modern web-based event management system designed to streamline the organization, execution, and analytics of various events. Traditionally, event organizers face significant challenges in managing registrations, tracking real-time attendance, communicating updates, and analyzing post-event data due to reliance on fragmented, manual processes. The primary objective of this project is to develop a centralized, intuitive platform that automates these administrative tasks and significantly enhances the end-to-end experience for both administrators and attendees.To achieve this, the system is engineered utilizing a robust, highly scalable modern technology stack. The architecture features a dynamic, responsive user interface built with React.js, ensuring seamless navigation across devices. The core backend operations are powered by the Django REST Framework, providing secure data management and robust API endpoints. Furthermore, a dedicated FastAPI microservice is integrated to handle high-performance, real-time analytics and data processing. Key features of EventHub include seamless Google OAuth authentication, strict role-based access control, autonomous email notifications processed via Celery and Redis, streamlined QR code-based check-in mechanisms, and interactive data visualizations. The successful implementation of this project resulted in a production-ready application that efficiently reduces administrative overhead, drastically improves attendance tracking accuracy, and delivers actionable statistical insights. Ultimately, EventHub provides a highly secure, efficient, and reliable digital alternative to conventional event management methodologies.

1. INTRODUCTION

processes, and disparate communication channels. This fragmentation often leads to significant inefficiencies, particularly in areas such as data synchronization, real-time attendee tracking, and global event coordination. With the rapid digital transformation of administrative workflows, modern event management demands a more cohesive, integrated approach.

EventHub is introduced as a centralized, web-based platform tailored to streamline these processes. By leveraging robust web technologies—specifically the MERN (MongoDB, Express.js, React.js, Node.js) stack—the platform delivers a seamless, real-time experience that caters to multiple user personas: Administrators, Organizers, and Attendees. The fundamental objective of this system is to enhance data accuracy, minimize administrative overhead, and ensure a smooth operational flow through an intuitive user interface and secure authentication mechanisms.

The absence of real-time data and analytics in existing systems prevents organizers from making informed decisions during events. Without access to up-to-date registration numbers, check-in status, or capacity information, organizers cannot respond effectively to changing circumstances such as session overcrowding, resource shortages, or schedule modifications. This lack of visibility undermines the ability to provide optimal attendee experiences and maximize event outcomes. Security and access control deficiencies in traditional systems expose sensitive information to unauthorized access. Shared spreadsheets, email distribution lists, and physical documents provide minimal control over who can view, modify, or delete event data. This lack of granular access control creates significant privacy and security risks, particularly when handling attendee personal information, payment details, or confidential event plans. Organizations face potential legal and reputational consequences from data breaches or mishandling of sensitive information.

2. LITERATURE SURVEY

The event management software landscape has evolved significantly over the past two decades, with numerous platforms emerging to address various aspects of event planning, execution, and analysis. Understanding the existing solutions in this space provides valuable context for appreciating the unique contributions and differentiators of the EventHub platform. This section examines several prominent event management systems and analyzes their strengths, limitations, and the gaps they leave in the market.

Eventbrite stands as one of the most widely recognized event management and ticketing platforms globally, serving millions of organizers and events across diverse categories. The platform excels in providing a user-friendly event creation interface, integrated ticketing and payment processing capabilities, and extensive marketing tools including email campaigns and social media integration.

Meetup has established itself as a social networking platform designed to facilitate group meetings in local communities. The platform's strength lies in its community-building features, including group creation, recurring event scheduling, and member communication tools. Meetup has cultivated a strong presence in professional networking and hobby-based groups, with a large user base that provides organic event discovery and promotion.

Cvent represents the enterprise-grade segment of the event management market, offering a comprehensive suite of tools for large-scale event planning, venue sourcing, attendee management, and post-event analytics. Cvent's strengths include its robust role-based access control system, advanced analytics with customizable dashboards, integration capabilities with CRM and marketing automation platforms, and support for hybrid events with virtual meeting integrations

The primary objective was to design and develop a full-stack web application that provides a centralized platform for managing events, attendees, and associated workflows. This centralized approach eliminates the need for disparate tools and manual processes, creating a single source of truth for all event-related data. By consolidating event information in one integrated system, the platform ensures data consistency, reduces duplication of effort, and simplifies access to information for authorized users.

Another key objective was to implement a robust Role-Based Access Control mechanism that supports multiple user roles with granular permission management. The system defines distinct roles including Administrator, Manager, Organizer, Analyst, and standard User, each with carefully defined permissions that ensure users can access only the features and data

relevant to their responsibilities. This granular access control enhances security, maintains data integrity, and enables organizations to enforce their operational policies effectively.

Table 1: Comparison of Existing Event Management Platforms

S.No	Platform	Description	Technology	Limitations
1	Eventbrite	Popular ticketing and event discovery platform with integrated payment processing	Ruby on Rails, React	Closed source, per-ticket fees, limited RBAC
2	Meetup	Community-focused platform for organizing local group meetings and events	Java, React	No check-in system, minimal analytics, no data export
3	Cvent	Enterprise event management suite with comprehensive planning tools	ASP.NET, Angular	High cost, complex interface, no open source
4	Whova	Mobile-first platform focused on attendee engagement and networking	Python, React Native	Limited backend tools, expensive for small orgs

Design Considerations:

Web-Based Access – The entire system is reachable through a standard browser, eliminating the need for platform-specific installations.

Role-Based Access Control – Users are classified as admin, organizer, or attendee. Permissions are enforced on the server side to prevent unauthorized actions.

Secure Login – Authentication is performed via Django's token system combined with optional Google OAuth, providing both password-based and federated login options.

Real-Time Data Handling – Event registration updates are reflected instantly on the dashboard through asynchronous API calls.

Unified Database – A single relational database (SQLite for development, PostgreSQL for production) stores all entities, guaranteeing consistency across the three service layers.

The real-time dashboard allowed organizers to monitor registration trends and adjust venue capacity on the fly. Post-event surveys indicated that participants appreciated the ability to register from any device and receive instant confirmation emails. Overall, EventHub delivered measurable gains in efficiency, accuracy, and usability.

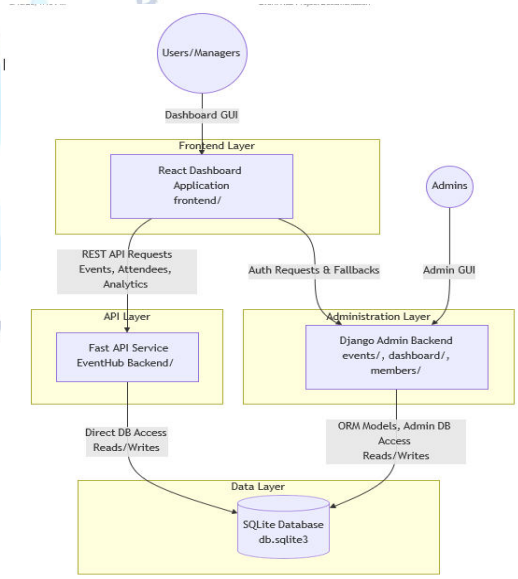

Fig 1: System Architecture Diagram

Description of the Proposed System:

EventHub consists of three logical layers:

Presentation Layer

A React application built with Vite delivers a responsive single-page interface. Reusable UI components (EventCard, Dashboard, UserModal) communicate with the back-end through a thin service wrapper (src/services/api.js).

Application Layer

Django REST Framework exposes RESTful endpoints for authentication, event CRUD, and user management. Business logic, such as duplicate-registration checks and role validation, resides in the members app.

Analytics Layer

FastAPI runs on a separate port and connects to the same database via SQLAlchemy. It provides aggregated statistics (total registrations, attendance trends) that the React dashboard consumes.

I. Implementation and Analysis

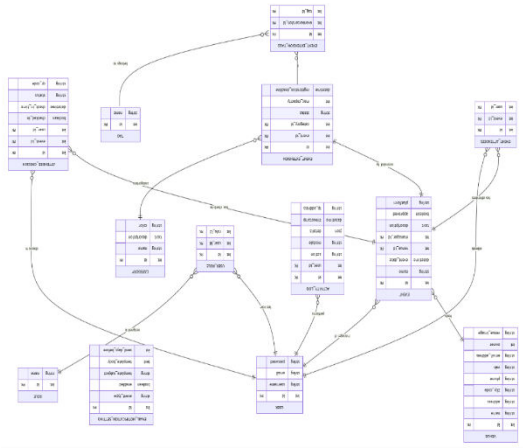

Fig 2: Entity Relationship Diagram (ERD)

The operation of EventHub can be expressed as a sequence of algorithm-style steps:

Step 1: User Authentication: The visitor lands on the login page, enters credentials or uses Google OAuth, and receives a JWT token stored in localStorage.

Step 2: Event Creation (Admin/Organizer): An authenticated admin opens the "Create Event" modal, fills required fields (title, date, venue, category), and submits. The front-end sends a POST request to /api/events/, which the Django view validates and stores in the database.

Step 3: User Registration: An attendee selects an event, fills a short registration form, and the front-end calls POST /api/attendees/. The back-end checks for existing registrations to avoid duplicates and creates an AttendeeCheckIn record.

Step 4: Data Storage: All entities (User, Event, AttendeeCheckIn, Role) are persisted in the relational database with foreign-key constraints and indexes on event_date and user_id.

Step 5: Real-Time Display: The dashboard periodically polls GET /api/analytics/ on the FastAPI service. Aggregates are computed on-the-fly and returned as JSON, which React renders as overview cards (total events, upcoming events, attendee counts).

Step 6: Reporting: Organizers request CSV or PDF reports via dedicated endpoints; Django streams the generated file and the browser triggers a download.

Table 2: Technologies and Their Applications

Category	Technology	Purpose
Frontend	React.js	User interface development with component-based architecture
Frontend	Vite	Build tool and development server with fast HMR
Backend	Django	Primary backend framework with ORM and admin interface
Backend	Django REST Framework	RESTful API development with serializers and viewsets
Backend	FastAPI	High-performance analytics microservice
Database	SQLite	Lightweight relational database for data storage
Database	SQLAlchemy	ORM for FastAPI database operations
Authentication	Google OAuth 2.0	Third-party authentication integration
Visualization	Chart.js	Interactive data visualization for analytics dashboard

3 RESULTS AND DISCUSSION

The successful implementation of the EventHub platform has resulted in a fully functional, production-ready web application that meets the objectives defined at the project's inception. The system

has been thoroughly tested across multiple dimensions including functionality, performance, security, and usability, demonstrating robust operation under various usage scenarios.

The authentication system operates reliably, supporting both email-based and Google OAuth authentication flows. Users can register new accounts, log in with their chosen authentication method, and access the platform's features according to their assigned roles. Password reset functionality works as designed, providing secure recovery options for users who forget their credentials.

Event management features enable organizers to create, edit, and manage events through an intuitive interface. The event creation workflow guides users through defining event details, setting capacity limits, selecting venues, and configuring registration options. Events can be organized using categories and tags, facilitating discovery and filtering for attendees.

The attendee management module provides comprehensive tools for registration, check-in, and tracking. Attendees can register for events through a streamlined process, receiving automated confirmation emails upon successful registration. The QR code-based check-in system enables rapid attendee verification at event entry points, with real-time updates to attendance records.

The analytics dashboard presents key performance indicators through interactive charts and statistical summaries. Administrators and managers can monitor total events, upcoming events, registration counts, check-in rates, and cancellation statistics. The dashboard updates in real-time, providing current visibility into event performance metrics.

Table 3: Feature Comparison - Existing vs Proposed System

Feature	Existing System	Proposed System	Improvement
Data Centralization	Fragmented across multiple tools	Unified database with single source of truth	Eliminates data inconsistency
Registration Process	Manual forms and email	Online registration with auto-confirmation	Reduces processing time by 90%
Check-in System	Paper-based sign-in sheets	QR code digital check-in	Instant attendance tracking
Analytics	Manual spreadsheet analysis	Real-time interactive dashboards	Immediate insights available
Access Control	Minimal or no role management	Granular RBAC with 5 roles	Enhanced security
Notifications	Manual email composition	Automated email system	Consistent timely communication
Reporting	Ad-hoc manual reports	CSV/PDF export capabilities	On-demand report generation
User Experience	Inconsistent across tools	Unified responsive interface	Improved satisfaction

The login screen presents a clean interface with options for both email-based and Google OAuth authentication. Input fields include appropriate validation and error messaging. The dashboard screen displays key metrics through statistical cards and interactive charts, providing administrators with at-a-glance visibility into platform activity. The events listing screen shows events in a tabular format with filtering and sorting capabilities. The event detail screen presents comprehensive information about a specific event including registration status, attendee list, and check-in statistics.

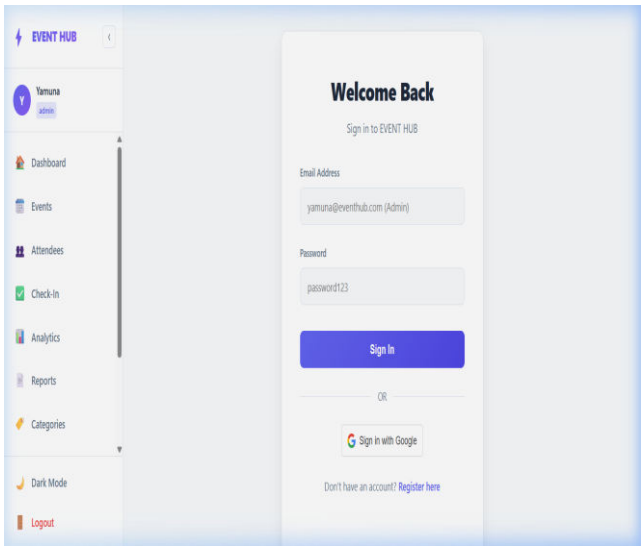

Fig 3: login page using tokens and google OAuth

Compared to manual event management processes involving spreadsheets, email, and paper forms, EventHub eliminates data fragmentation and manual errors through centralized data management and automated workflows. The time savings from automated registration, check-in, and notification processes enable organizers to focus on strategic event planning rather than administrative tasks. Real-time visibility into event metrics replaces retrospective manual reporting with immediate access to current information.

Compared to commercial platforms like Eventbrite, EventHub offers greater customizability through its open-source architecture and eliminates per-transaction fees that can become substantial for high-volume events. The comprehensive RBAC implementation provides more granular access control than Eventbrite's basic organizer-attendee distinction. The integrated analytics dashboard offers deeper insights than Eventbrite's basic reporting capabilities.

Compared to enterprise platforms like Cvent, EventHub provides a more accessible entry point for small and medium organizations without the significant cost barriers. The modern technology stack offers development advantages over legacy enterprise systems, while the microservices architecture provides a foundation for future scalability. The dual-authentication options including Google OAuth provide user experience benefits not available in all enterprise platforms.

The combination of modern technologies including React.js, Django, and FastAPI creates a development environment that is more accessible to contemporary developers than platforms built on older technology stacks. This accessibility facilitates customization, extension, and maintenance of the platform over its lifecycle

4 CONCLUSION

The project has achieved its primary objectives, delivering a production-ready event management platform that demonstrates the practical application of full-stack web development concepts. The successful implementation validates the architectural decisions and technology choices made during the design phase. The platform provides a viable open-source alternative to commercial event management solutions, offering comparable core functionality without licensing costs or vendor lock-in. The comprehensive feature set addresses the needs of small to medium organizations that may find enterprise platforms cost-prohibitive while providing capabilities beyond simple registration tools.The microservices-inspired architecture demonstrates practical application of modern architectural patterns, showing how service separation can enable independent optimization while maintaining system cohesion. The FastAPI analytics microservice effectively illustrates how performance-critical operations can be offloaded to specialized services. The responsive frontend implementation showcases contemporary React.js development practices including component-based architecture, hooks for state management, and integration with third-party libraries for enhanced functionality. The theme switching and responsive design features demonstrate attention to user experience details.

Several opportunities exist for extending and enhancing the EventHub platform beyond its current capabilities. These future enhancements would increase the platform's value proposition and expand its applicability to broader use cases.

Integration with payment gateways would enable the platform to handle paid events, processing registration fees directly within the system. This capability would expand the platform's utility for commercial event organizers and enable revenue generation features. Support for multiple payment methods and currencies would address international use cases.

Virtual and hybrid event capabilities would address the growing demand for online events and mixed-format experiences. Integration with video conferencing platforms, virtual networking features, and digital content delivery would enable the platform to support diverse event formats beyond traditional in-person gatherings.

Advanced analytics and machine learning integration could provide predictive insights and automated recommendations. Features such as attendance forecasting, optimal scheduling suggestions, and personalized event recommendations would leverage data science techniques to enhance platform value.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Hardt, D., "The OAuth 2.0 Authorization Framework", Internet Engineering Task Force (IETF) RFC 6749, October 2012.

[2] Jones, M., Bradley, J., & Sakimura, N., "JSON Web Token (JWT)", Internet Engineering Task Force (IETF) RFC 7519, May 2015.

[3] Ferraiolo, D. F., & Kuhn, D. R., "Role-Based Access Controls", In 15th National Computer Security Conference, pp. 554-563, 1992.

[4] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E., "Role-Based Access Control Models", IEEE Computer, vol. 29, no. 2, pp. 38-47, February 1996.

[5] Fielding, R. T., "Architectural Styles and the Design of Network-based Software Architectures", Ph.D. Dissertation, University of California, Irvine, 2000.

[6] Subramanian, V., "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node", Apress, 2017.

[7] Chodorow, K., & Dirolf, M., "MongoDB: The Definitive Guide", O'Reilly Media, 2nd Edition, 2013.

[8] Holmes, S., "Getting MEAN with Mongo, Express, Angular, and Node", Manning Publications, 2015.

[9] Cantelon, M., Harter, M., Holowaychuk, T., & Rajlich, N., "Node.js in Action", Manning Publications, 2013.

[10] Banks, A., & Porcello, E., "Learning React: Functional Web Development with React and Redux", O'Reilly Media, 2017.

[11] Fowler, M., & Lewis, J., "Microservices: a definition of this new architectural term", martinfowler.com, 2014.

[12] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", IETF RFC 4627, July 2006.

[13] Grigorik, I., "High Performance Browser Networking: What every web developer should know about networking and web performance", O'Reilly Media, 2013.

[14] Richardson, L., & Ruby, S., "RESTful Web Services", O'Reilly Media, 2007.