iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72545

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Sangam: Smart Crowd Management and Assistance Platform

Hema Verma[1], Vaishali Dewangan[2], Ms. Sonia Wadhwa[3]
Department Of Computer Science, Government Engineering College, Bilaspur

Abstract: Sangam is a scalable, web-based platform built to manage massive religious congregations like the Kumbh Mela [15]. It leverages modern web technologies to digitize and streamline essential services such as crowd monitoring, ghat allotment, hotel booking, prasad distribution, multilingual assistance, and a lost & found mechanism. Using real-time communication[12] and intelligent automation[3], Sangam enhances safety, improves visitor experience, and simplifies administrative efforts.
Keywords: Event Management, Crowd Monitoring, Web Application, Socket.IO, Multilingual Assistant, Real-Time System, React.js, Node.js, MongoDB

I. INTRODUCTION

Mass religious gatherings like the Kumbh Mela, which attract millions of devotees from across the globe, represent one of the most complex event management challenges [15]. These events stretch the limits of public infrastructure and require the coordination of numerous services such as accommodation, transportation, crowd control, health care, and security [6]. The scale and diversity of the attendees, coupled with the spiritual and cultural significance of the event, make seamless management not just a logistical necessity but also a moral imperative. Despite advancements in digital technologies and smart city initiatives [6], the majority of Indian religious festivals continue to rely heavily on manual, paper-based systems and traditional communication methods such as loudspeaker announcements and physical signboards. These approaches are inadequate for handling real-time issues like crowd surges, lost individuals, or fluctuating resource availability [4]. As a result, inefficiencies, overcrowding, delayed response times, and safety hazards are common occurrences [11]. To address these limitations, Sangam introduces a smart, web-based platform that centralizes and digitizes essential services, including crowd monitoring, ghat allotment, prasad management, hotel booking, multilingual assistance, and lost & found [5]. Built with modern technologies like React.js [8], Node.js [9], MongoDB [10], and Socket.IO [12], the platform offers real-time updates, intuitive UI, and AI-powered multilingual interaction through OpenRouter [14]. By automating key processes and enhancing accessibility, Sangam not only improves visitor experience but also empowers administrators with data-driven insights [3], ensuring safer, smoother, and more efficient festival management.

II. LITERATURE SURVEY

A. Background and Related Work

The management of large-scale public gatherings has long been a focus of research, particularly in contexts such as religious events, concerts, marathons, and political rallies [11]. The complexity of these events lies in ensuring safety, coordinating services, and maintaining an optimal experience for attendees. Various studies have explored the integration of Internet of Things (IoT) [2], Artificial Intelligence (AI) [3], Geospatial Analytics, and Crowd Simulation Models [11] to address such challenges. In the context of religious gatherings, significant work has been done in places like Mecca for Hajj management [1]. Systems implemented there rely on RFID-based tracking, automated crowd estimation, and surveillance analytics. These approaches emphasize real-time monitoring and scalable infrastructure [1]. However, such systems are often expensive and require highly specialized infrastructure, limiting their applicability in decentralized, diverse environments like Indian religious festivals [5].

B. Gaps in Existing Systems

Despite the progress in crowd analytics and IoT integration[2][3], there is a noticeable gap when it comes to systems specifically designed for Indian religious gatherings[6]. Most local events still depend on:

- Manual ghat and accommodation allotment
- Physical announcement systems
- Limited crowd estimation tools[5]
- No multilingual digital assistant or centralized digital portal[7]

Moreover, existing crowd control systems often fail to support **real-time communication**, **user personalization**, and **modular design**, which are crucial for dynamic and diverse settings.

C. Innovations by Sangam

Sangam bridges these gaps by adopting a web-centric, modular, and user-accessible approach. Unlike static or hardware-heavy systems[5], Sangam:

- Leverages React.js for component-based front-end development[8]
- Utilizes Node.js and MongoDB for scalable backend operations[9][10]
- Implements Socket.IO for real-time updates such as ghat status and crowd alerts[12]
- Incorporates AI-powered multilingual assistants via OpenRouter to support diverse linguistic backgrounds[14]
- Provides a QR-based lost & found system for efficient identity verification and recovery[2]

This blend of technologies provides flexibility, real-time responsiveness, and broader inclusivity, making Sangam uniquely suited to Indian cultural and infrastructural contexts.

D. Positioning Sangam in Current Research

While many smart city models and event management platforms emphasize urban infrastructure or entertainment-based crowd control[6], Sangam focuses on spiritual gatherings where the cultural, emotional, and linguistic diversity of visitors is far greater. As a result, it contributes a domain-specific innovation to the literature by offering:

- A practical model for Indian festival digitization[5]
- A blueprint for real-time public service automation
- A scalable template for other mass events like Rath Yatra or Ganesh Utsav

III. PROBLEM STATEMENT

Despite advancements in smart infrastructure, Indian religious events continue to face several pressing challenges due to the absence of integrated, real-time, and user-friendly systems.

The key problems addressed by this project are:

1) Overcrowding and Traffic Congestion: Massive gatherings often lead to unmanaged crowds and chaotic traffic, posing safety risks and delays in movement.
2) Inadequate Emergency Planning: Emergency responses are delayed due to poor coordination, lack of data, and absence of real-time alert systems.
3) Difficulty Accessing Food and Accommodation: Visitors struggle to locate hygienic and affordable food or shelter due to lack of guidance and organized distribution.
4) Missing Persons and Ineffective Lost & Found Systems: With large crowds, people often get separated, and there is no streamlined process to report, trace, or reunite them.
5) Lack of Smart Crowd Control Technologies: Current event management practices rely heavily on manual operations, lacking AI-based tools for crowd flow analysis and control.
6) Manual Ghat and Accommodation Allocation: Assignments for ghats (ritual bathing spots) and lodging are handled manually, leading to disputes, delays, and inefficiency.
7) No Centralized Registration or Visitor Records: The absence of a unified database hinders planning, tracking, and analytics for better service delivery.
8) Lack of Multilingual Digital Communication: Visitors from diverse linguistic backgrounds face difficulties due to non-inclusive communication systems.
9) Delayed Response to Overcrowding or Lost Persons: Due to lack of real-time monitoring and digital reporting, critical situations often escalate before action is taken.

IV. METHODOLOGY

The methodology of Mansha follows an Agile IoT development approach, combining iterative software development with structured hardware prototyping and real-time system integration to enable seamless brain-to-device interaction.

A. Sangam Methodology: Agile

1) Requirements – Understanding Sangam's Purpose

In the initial stage, we identified the core objectives of Sangam by interacting with stakeholders and understanding the event's operational needs.

We gathered detailed requirements around:

- Crowd managemen
- Lost & found system
- Accommodation tracking
- Food distribution

Agile allowed us to keep the scope flexible, so as we discovered new event requirements or user expectations, we could easily update our backlog.

2) Design – Planning User-Centered Interfaces

With the requirements in place, we began designing the structure and flow of Sangam.

This involved:

Sketching wireframes for each page (Home, Services, Login, etc.)

Designing the UI with a modern, responsive layout

Selecting fonts, color palettes, and iconography to reflect a clean, professional look

We kept the design modular and adaptable, so components could evolve as user needs became clearer during development.

3) Development – Building Feature by Feature

Development was carried out in **iterations (sprints)**. Each sprint focused on specific features:

- First sprint: Homepage and hero section
- Second sprint: Individual service pages like Crowd Management
- Third sprint: Navbar and responsive design
- Fourth sprint: Authentication (Login/Signup)

We used reusable components in React and followed clean code practices. Agile's iterative approach helped us integrate features gradually and refine them with user input.

4) Testing – Ensuring a Smooth Experience

At the end of each sprint, we performed **continuous testing**:

- Functional testing for buttons, navigation, and forms
- Responsiveness testing across devices
- Cross-browser testing
- Feedback from peers and users for usability

Any bugs or UI inconsistencies were logged and addressed in the next sprint, which is central to Agile's principle of ongoing improvement.

5) Deployment – Going Live in Stages

Deployment was handled incrementally. Rather than waiting to finish the entire site, we deployed major sections as they were completed:

- Initial static version for review
- Later versions with interactivity and backend-ready structure

This allowed stakeholders to **see progress early**, offer suggestions, and ensure the project was aligned with expectations.

6) Review – Gathering Feedback & Improving

After each release, we conducted internal reviews and collected feedback from mentors, team members, and potential users. This review cycle helped us:

- Identify areas for UX/UI improvements

- Add new sections like FAQs, animations, and dark mode
- Improve responsiveness and accessibility

Each review informed our next sprint, truly embodying **Agile's cycle of continuous feedback and enhancement**.

.

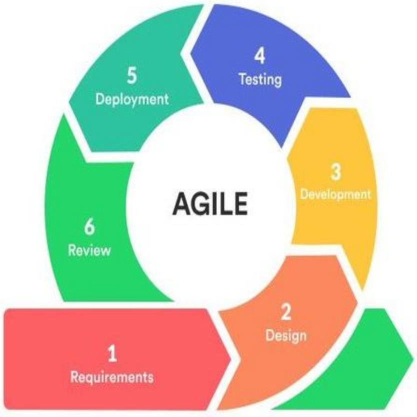

Fig 1:- Agile Methodology

B. Stakeholder Engagement and System Design

Sangam is built on a dynamic and user-centric system design that integrates modern web technologies to streamline large-scale event management. The methodology behind Sangam emphasizes real-time responsiveness, intuitive navigation, and modular scalability—ensuring an efficient experience for both attendees and organizers. By combining smart crowd control, seamless accommodation mapping, and an intelligent lost & found system, Sangam delivers a holistic solution tailored to the complex needs of cultural festivals and gatherings. Its architecture reflects a perfect blend of accessibility, performance, and innovation, aimed at transforming how events are planned and managed..

1) *Requirement Analysis – Identifying Key Festival Services*- We started by identifying core services needed for festival management—such as crowd control, lost & found, accommodation, food tracking, and navigation. These were prioritized based on inputs from organizers, volunteers, and past participants. This step shaped the system's goals and informed the design and development strategy.
2) *UI/UX Design – Using Bootstrap and Custom StylesSelection* - We designed a clean, responsive interface using Bootstrap for layout and custom CSS for branding. The design featured a light theme with cream and brown tones, a hero section with a call-to-action, structured service pages, and a mobile-friendly navbar with dark mode support.
3) *Component-Based Frontend – Built in React.js* - Using React.js, we created modular components like the Navbar, Forms, and Service Cards. Each section (e.g., Lost & Found, Login) was a separate component, allowing easy reuse and scalability. Routing was managed with React Router for smooth navigation.
4) *RESTful APIs & State Management – Via Node.js and React Context*- We built RESTful APIs with Node.js and Express to connect the frontend to MongoDB. On the frontend, React Context API was used to manage global state—like login status and theme preferences—ensuring smooth data flow across components.
5) *Real-Time Features – Implemented Using Socket.IO* - Socket.IO enabled real-time updates for modules like Lost & Found and planned crowd alerts. It allowed immediate data sharing, improving user experience and responsiveness, especially in dynamic festival environments.
6) *Database Design – Structured with MongoDB* - We used MongoDB to define schemas for users, items, feedback, and bookings. This setup supported data validation, easy querying, and seamless backend integration for scalable data handling

V. SYSTEM HARDWARE SPECIFICATION

A. Hardware Module

The Sangam Hardware Module is a comprehensive setup designed to facilitate efficient crowd management, attendee identification, and real-time event monitoring. It integrates components such as QR code-enabled wristbands, 2D barcode scanners, CCTV surveillance systems, edge computing units, and networking infrastructure.

The module ensures seamless data acquisition, real-time crowd analytics, and secure identification through scalable and robust hardware tailored for high-density environments. Power backup solutions and mobile access devices support uninterrupted field operations and system control across the venue.

1) *QR Code Wristbands*- Sangam uses waterproof, tamper-proof QR code wristbands made from Tyvek or silicone for attendee identification, tracking, and Lost & Found management. These wristbands are compatible with 2D QR scanners and smartphones, ensuring quick and reliable access across event checkpoints.
2) *QR Code Scanners* - To scan wristbands at entry/exit points, accommodation areas, and food counters, Sangam employs 2D QR code scanners available in both wired and wireless variants. These scanners support USB and Bluetooth connectivity, making them suitable for both fixed stations and mobile operations.
3) *CCTV / Surveillance Cameras*- For real-time crowd monitoring and AI-based analytics, Sangam uses IP cameras with Wide Dynamic Range, offering a minimum resolution of 1080p, with 4K preferred for enhanced clarity. These cameras feature a 90°–120° field of view, motion detection.
4) *Edge Computing Device / Server (Optional for Local AI Processing)* - For real-time video analytics and offline operations, Sangam utilizes edge computing units equipped with at least an Intel i5 or Ryzen 5 processor, 8GB RAM, 256GB SSD, and an NVIDIA GTX 1650 GPU or higher. This setup enables efficient processing of surveillance data and supports basic AI model execution on-site.
5) *Wi-Fi Routers and Network Setup*- To ensure real-time data syncing across devices and service stations, Sangam relies on dual-band or tri-band routers, with a mesh network setup recommended for large venues. This provides stable, high-speed connectivity essential for smooth coordination and data flow throughout the event area.

VI. SYSTEM SOFTWARE SPECIFICATION

The Sangam Software Framework is a comprehensive suite of tools and technologies designed to manage event operations, including crowd monitoring, attendee tracking, and real-time data processing. The system begins with QR code wristbands and scanners for attendee identification and tracking, which feed data to a central Node.js backend. Python is employed for real-time crowd analysis using OpenCV and AI-based models to predict crowd behavior and density. The frontend is built with React.js for a dynamic, responsive user interface, while RouterAI facilitates multilingual chatbot assistance. Socket.IO ensures seamless real-time communication between clients and the backend, and external devices are controlled via RESTful APIs for enhanced event management.

1) *Frontend*- HTML, CSS, Bootstrap, and React.js build a responsive, modern, and dynamic user interface. React.js ensures efficient, scalable development with real-time updates, while Bootstrap ensures a mobile-first design with ready-to-use components for faster development.
2) *Backend*- Node.js serves as the backend framework, providing a lightweight and efficient environment for handling multiple requests simultaneously. MongoDB is used for database management, offering a flexible, scalable, and document-oriented approach to store attendee and event data.
3) *Data Processing* - OpenCV is utilized for real-time crowd analysis, processing video feeds from surveillance cameras to track crowd density, movement, and behavior, feeding this data into the AI-based analytics system for decision-making.
4) *Real-time Communication* - Socket.IO is employed to enable real-time communication between clients (attendees, staff) and the backend, ensuring instant updates and notifications, such as entry/exit alerts and crowd density warnings.
5) *Chatbot Assistance* - RouterAI is integrated for multilingual chatbot assistance, offering attendees support in various languages and ensuring a seamless experience for users from different regions.

VII. LIMITATIONS

While the **Sangam** event management system offers advanced features for real-time crowd monitoring and attendee tracking, there are certain limitations to consider in its deployment and operation. These challenges must be addressed to enhance system performance and user experience.

A. *Data Accuracy and Siganl Reliability*

- Environmental Factors: External factors like poor lighting or camera angle can impact the accuracy of CCTV surveillance and crowd analysis.
- Sensor Sensitivity: QR code wristband scanners might experience occasional misreads in high-traffic areas, affecting tracking precision.

B. Scalability in Large Venues

- Network Load: Large venues with thousands of attendees may experience network congestion, slowing down data syncing and causing delays in real-time updates.
- Hardware Limitations: Continuous real-time video feed and data processing require robust edge computing devices, which may face performance bottlenecks during peak load times.

C. System Integration and Maintenance

- Multilingual Assistance: While RouterAI provides multilingual chatbot support, it may not fully cover all regional dialects, limiting the system's accessibility for certain users.
- Third-Party Device Compatibility: Ensuring compatibility between the QR code scanners, CCTV systems, and other third-party devices can be challenging, requiring constant updates and testing.
- Maintenance Overhead: Frequent software and hardware updates are necessary to maintain system security, performance, and to accommodate new features, which may require significant effort for ongoing support.

VIII. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to everyone who contributed to the successful development and implementation of the Sangam event management system. First and foremost, we thank our mentors and advisors for their invaluable guidance and support throughout this project. We also extend our appreciation to the Sangam team for their dedication and hard work in bringing this vision to life.

Our deepest thanks go to the developers and technical partners who helped integrate cutting-edge technologies, such as QR code wristband tracking, CCTV-based crowd analysis, and AI-driven analytics, ensuring the system’s functionality and scalability. We are also grateful to RouterAI for their support in providing multilingual chatbot assistance, enabling a seamless user experience for attendees.

Finally, we would like to acknowledge our families and friends for their encouragement and patience during the course of this research.

REFERENCES

[1] IEEE Smart Cities, “Smart Crowd Management Technologies for Large-Scale Events,” *IEEE Xplore*, 2021. Available: https://ieeexplore.ieee.org

[2] Journal of Emerging Technologies, “IoT-based Lost & Found Systems Using RFID and QR Code,” *J. Emerging Technol.*, vol. 12, no. 4, 2020.

[3] Springer AI in Public Safety, “AI-driven Real-time Crowd Density Analysis for Public Events,” *SpringerLink*, 2019. Available: https://link.springer.com.

[4] Ministry of Home Affairs, Government of India, “Smart Event Planning & Safety Solutions for Mass Gatherings,” 2022.

[5] API Setu, Government of India, “Real-time API Access for Emergency Alerts and Public Information,” Available: https://www.apisetu.gov.in.

[6] NITI Aayog, “Smart City Initiatives for Crowd Control and Public Safety,” 2021.

[7] A. Kasinathan, A. Mustapha, and C. K. Bin, “A Customizable Multilingual Chatbot System for Customer Support,” *Proc. 2021 ACM Conf. on Human-Computer Interaction*, 2021. Available: https://pdfs.semanticscholar.org/7e70/0993405810ce2b571d02a533688a9ea88aaa.pdf.

[8] React.js Documentation, “Getting Started – React,” [Online]. Available: https://react.dev/.

[9] Node.js Documentation, “About Node.js,” [Online]. Available: https://nodejs.org/en/docs/.

[10] MongoDB Documentation, “MongoDB Manual,” [Online]. Available: https://www.mongodb.com/docs/.

[11] Mongoose Documentation, “Mongoose ODM Documentation,” [Online]. Available: https://mongoosejs.com/docs/.

[12] Socket.IO Documentation, “Socket.IO – Real-time Engine,” [Online]. Available: https://socket.io/docs/.

[13] Bootstrap Documentation, “Introduction to Bootstrap 5,” [Online]. Available: https://getbootstrap.com/docs/5.0/getting-started/introduction/.

[14] OpenRouter AI Documentation, “OpenRouter API Documentation,” [Online]. Available: https://openrouter.ai/docs.

[15] Harvard University South Asia Institute, “Managing the Kumbh Mela: Mapping the Largest Gathering on Earth,” South Asia Institute, Harvard University, 2015.

[16] S. Darianian and M. P. Michael, “Smart Event Management using IoT Technologies,” in *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 3, pp. 494-500, 2017, doi: 10.1109/TETC.2017.2679165.

[17] G. K. Still, “Crowd Management: Past and Contemporary Research and Future Directions,” *Safety Science*, vol. 72, pp. 65–68, 2014, doi: 10.1016/j.ssci.2014.08.006.