Sandip Foundations

Sandip Institute of Engineering and Management

Department of Computer Engineering

Academic Year 2025- 2026

B.E Project Synopsis

Trinetra :- A Smart, Secure and Scalable Pilgrimage Management System Using AI

PROJECT SYNOPSIS

Abstract:-

The Trinetra is an innovative crowd management and safety system designed specifically for pilgrimage places, aiming to ensure a secure, sustainable, and seamless experience for pilgrims. The system addresses challenges like overcrowding, lost individuals, and real-time communication between users, authorities, and administrators. It leverages modern technologies such as **Firebase**, **OpenCV with YOLOv5**, and **cloud-based analytics** to provide intelligent crowd monitoring, route planning, lost-and-found reporting, and emergency alerts. Key features include a **role-based authentication system** (User, Authority, Admin), **lost-and-found person reporting**, **real-time crowd detection** using cameras and AI models, and **LED screen broadcasting** of unresolved cases. The **Route Planner** and **Nearby Essentials Finder** help pilgrims navigate safely and efficiently, while the **Emergency SOS button** provides immediate access to authorities. Authorities can manage reports, view heatmaps, assign volunteers, and broadcast alerts through a dedicated dashboard. The system is multilingual (English, Hindi, Marathi) and integrates **text-to-speech** and **voice-input functionalities** for improved accessibility. By combining crowd data analysis, cloud notifications, and user-centric design, Trinetra ensures the safety and convenience of pilgrims, particularly during large gatherings. Future expansions may include offline caching, advanced analytics, and integration with IoT-based surveillance systems, making it a scalable and impactful solution for smart pilgrimage management.

Keywords: Crowd Management, Pilgrimage Safety, Lost & Found System, Real-Time Alerts, Multilingual Interface

Objectives:

- To develop a robust crowd detection and monitoring
- system for pilgrimage places to prevent overcrowding.
- To implement a lost-and-found reporting module with automated matching and LED broadcasting.
- To create role-based dashboards (User, Authority, Admin) for seamless communication and case management.
- To integrate real-time alerts, route planning, and multilingual support for enhanced user experience.

Literature Review: -

Paper 1 :- "Leveraging Machine Learning for Real-Time Crowd Control and Safety at Kumbh Mela"

This paper presents SAMRAKSH, a real-time AI-powered framework designed to improve crowd safety and management during massive events like the Kumbh Mela. By integrating YOLOv11 for crowd and weapon detection, ByteTrack for tracking and blockade identification, and DeepFace for lost person recognition, the system ensures rapid and automated responses to dynamic crowd scenarios. The backend, built using FastAPI and WebSockets, processes live video feeds with high accuracy and minimal latency. A companion mobile app delivers real-time alerts. Tested during simulated environments, the system achieves over 90% accuracy in crowd estimation and proves scalable for other high-density gatherings like concerts or religious festivals.

Paper 2 :- "Web-Based Lost and Found System"

The paper titled "Web-Based Lost and Found System" presents the design and development of a web-based application aimed at simplifying the process of reporting and recovering lost items. Traditional methods like lodging complaints at police stations or using social media and print media often prove inefficient and fragmented. To address this, the proposed system offers a unified digital platform where users can register and report lost or found items by submitting detailed descriptions and optional images. The system operates through three modules: the Citizen Module (for users to raise or respond to complaints), the Admin Module (for verifying user identities), and the Police Module (acting as a mediator to verify claims and facilitate item recovery). The application uses structured forms, data flow diagrams, and verification mechanisms to ensure a reliable and secure user experience. Ultimately, the system aims to enhance community collaboration and streamline the lost-and-found process through dynamic data handling and efficient information flow.

Paper 3 :- "Influence of Kumbh Mela on Indian Society: A Study"

The paper titled "Influence of Kumbh Mela on Indian Society: A Study" by Prof. Sanjay Prasad Sharma explores the profound cultural, spiritual, social, and economic impact of the Kumbh Mela, one of the world's largest religious gatherings. Rooted in ancient mythology and celebrated at four sacred riverbanks in India, the Kumbh Mela serves as a powerful symbol of India's unity in diversity. The study delves into its historical origins, spiritual significance, and its modern-day influence on arts, literature, media, tourism, and infrastructure. It highlights how the festival boosts local economies, fosters traditional craftsmanship, and provides a global platform for cultural exchange. While acknowledging challenges such as environmental concerns and crowd management, the paper emphasizes the Kumbh Mela's evolving nature and its global recognition, notably its inclusion in UNESCO's Intangible Cultural Heritage list. Ultimately, the study portrays the Kumbh Mela as more than a religious event—it is a celebration of India's enduring spiritual legacy, communal harmony, and cultural richness that resonates far beyond national borders.

Feasibility Study:

1. Technological Feasibility

Trinetra is built using modern and reliable technologies that ensure scalability, real-time data handling, and smooth user experience. The frontend is developed using Flutter, which supports cross-platform mobile development with high performance. Firebase is used as the backend for authentication, real-time database, cloud storage, and push notifications, ensuring seamless communication between users, authorities, and admins. For crowd management, OpenCV with YOLOv5 (or MobileNet-SSD) provides AI-powered object detection and crowd counting capabilities. The use of widely adopted frameworks and cloud services ensures minimal compatibility issues and simplifies future maintenance and upgrades.

2. **Algorithmic Feasibility**

The project employs efficient and well-proven algorithms to achieve its objectives:

- **Crowd Counting Algorithm:** Using YOLOv5 and OpenCV, real-time detection of people in camera feeds is achieved with high accuracy while maintaining low computational overhead.

- **Lost & Found Matching Algorithm:** A rule-based matching algorithm is implemented, which compares multiple parameters (age, clothing description, last seen location, and time) to identify potential matches without requiring resource-heavy facial recognition.

- **Route Planning Algorithm:** The system integrates **Google Maps API and dynamic routing** to suggest safe, low-crowd paths based on real-time crowd data. The combination of these algorithms ensures that the system operates efficiently even on mid-level hardware, without compromising performance or accuracy.

3. **Development and Deployment Feasibility**

The development of Trinetra is both cost-effective and time-efficient due to the availability of open-source tools, cloud-based backends, and pre-trained AI models. The project follows a modular development approach, which makes testing and integration easier. Deployment is cloud-based, where Firebase Hosting and Cloud Functions handle backend operations, while Android applications serve end users and authorities. The AI modules (crowd counting) can be deployed on a local laptop or server at the pilgrimage site, which sends live data to Firebase. This architecture ensures quick deployment with minimal infrastructure costs. Moreover, the solution can be scaled by simply adding more cameras, devices, and cloud resources as required.

Methodology/ Planning of work :-

1. **Research Type & Overall Approach**
   - **Research Type:** *Applied, solution-driven R&D* focused on building and validating a real-time, AI-assisted crowd management and safety platform for pilgrimage contexts.

   - **Methodological Approach: Iterative & incremental (Agile/Scrum)** with short sprints, continuous integration, and user feedback loops from authorities and pilot users.

   - **Units of Analysis:** (i) **Crowd density & flow** at locations (temples/ghats), (ii) **Lost/Found case lifecycle**, (iii) **System response time, accuracy & reliability**, (iv) **User engagement & usability metrics** (e.g., SOS use, report resolution time).

2. **High-Level Phases**

   I. **Problem Understanding & Requirement Engineering**
   - Stakeholder interviews (pilgrims, police/authorities, volunteers).
   - Define feature backlog (User, Authority, Admin).
   - Success metrics (e.g., alert latency < 5s, crowd counting accuracy > 85% on PoC footage).

   II. **System Architecture & Tech Stack Finalization**
   - Choose **Flutter** + **Firebase** stack for app/backend, **OpenCV** + **YOLOv5/MobileNet-SSD** for crowd counting.
   - Design **data models** (Firestore schemas for reports, roles, alerts, screens).
   - Define **RBAC (Role-Based Access Control)** and **security rules** (Firebase Rules).

   III. **PoC: Crowd Counting & Alerting Pipeline**
   - Build Python script for live detection → push counts to Firebase every 10s.
   - Implement **Cloud Function triggers** for threshold breaches → push FCM alerts.
   - Validate accuracy, speed, and false positives.

   IV. **MVP Mobile App (User Role)**
   - **Auth (Signup/Login with role selector)**, multilingual UI (EN/HI/MR).
   - **Lost/Found forms**, **My Reports**, **SOS**, **Nearby essentials**, **Route planner** (Maps API).
   - Accessibility: **TTS, Voice Input, Font Scaling**.

   V. **Authority Dashboard & Workflows**
   - **Real-time report list**, filters, **accept/reject/found** actions.
   - **Location mapping**, volunteer assignment, **notes/sightings**, **chat with user**.
   - **Heatmaps & analytics (basic)**.

   VI. **Admin Portal & System Governance**
   - **User/Authority management**, permissions & document verification.
   - **LED screen registration & control**, notification templates, override controls.
   - **Complete analytics dashboard** (resolution time, density by region/time).

   VII. **LED Broadcast & Blinking Photo Engine**
   - Rule engine to select unresolved lost persons, rotation logic, multilingual display.
   - Admin/Authority approval workflow.

   VIII. **Matching Logic (Non-FR, Rule-Based)**
   - Implement multi-parameter similarity scoring (age, gender, clothing, time/location proximity, keywords).
   - Threshold-based alerts (e.g., ≥70% similarity) → human verification.

IX. **Testing, Validation & Optimization**

- **Unit**, **integration**, **end-to-end**, **load**, **security**, **usability**, **a11y** testing.
- **Model validation** for crowd counting using annotated sample videos.
- Optimize latency & Firebase costs (indexes, batching, caching).

X. **Deployment, Monitoring & Documentation**

- **Staged rollout** (pilot temple / event).
- Monitoring with **Firebase Analytics, Crashlytics, Logs Explorer**.
- Create **SOPs** for authorities/admin & end-user manuals.
- Continuous improvement & feedback incorporation.

3. Data Collection & Analysis Methods

Data Sources:

- Camera feeds (entry/exit) for person counting.
- App telemetry (crowd status checks, SOS triggers, route searches).
- Firestore collections (reports, users, actions).

Analytics & Visualization:

- Crowd heatmaps (by time/location).
- Time-to-resolution stats for lost/ found.
- Alert frequency & response times.
- Tools: Firebase Analytics, BigQuery export (optional), Python/Colab for offline analysis.

Evaluation Metrics:

- Crowd counting accuracy, alert latency, match suggestion precision, system uptime, user satisfaction.

4. Tools, Frameworks & Infrastructure

- **Frontend:** Flutter (Android primary), Dart.
- **Backend & Infra:** Firebase Auth, Firestore / Realtime DB, Cloud Functions, FCM, Firebase Hosting.
- **AI / CV:** Python, OpenCV, YOLOv5 or MobileNet-SSD.
- **Maps & Routing:** Google Maps Platform.
- **DevOps & QA:** GitHub/GitLab CI, Postman, Jest/Flutter test, Locust/JMeter (load), OWASP ZAP (security).
- **Project Management:** Jira/Trello, Agile boards, weekly sprint reviews & demos.

5. Security, Privacy & Compliance Plan

- **Role-based access control** with strict Firebase Security Rules.
- **PII minimization** (no facial recognition, optional ID verification only for authorities/admins).
- **Encrypted communication** (HTTPS/TLS).
- **Audit logging** for sensitive actions (status overrides, LED broadcasts).
- **Data retention & deletion policies** (resolved cases archived/ anonymized).

Facilities required for proposed work:-

- **Software tools :**

For the development of Trinetra, the software requirements include Flutter for cross-platform mobile app development, Firebase for authentication, real-time database, cloud storage, and push notifications, and OpenCV with YOLOv5 and TensorFlow for AI-powered crowd detection and monitoring. Google Maps API is utilized for route planning and navigation, while Python is employed for the backend AI modules and crowd counting scripts. Node.js is required for implementing Firebase Cloud Functions to handle serverless backend logic, real-time alerts, and automated notifications.

- **Hardware tools :**

On the hardware side, the system requires Android smartphones or tablets for user and authority applications, along with LED display boards for broadcasting alerts and lost person details. For crowd detection, laptops or PCs equipped with webcams will be installed at entry and exit points to capture live video feeds. These hardware and software resources collectively ensure the seamless implementation, testing, and deployment of the proposed system.

Expected outcomes:-

The implementation of **TrinetraGuard – A Crowd Management System** is expected to significantly enhance safety, efficiency, and user experience at pilgrimage sites. The system will provide **real-time crowd monitoring and alerts**, helping authorities prevent overcrowding and manage crowd flow effectively. Through the **Lost-and-Found reporting module** and LED display broadcasting, missing persons can be identified and reunited with their families more quickly. Pilgrims will benefit from **smart route planning, emergency SOS features, and multilingual support**, ensuring convenience and accessibility for people of all age groups. Authorities and administrators will gain valuable insights through **analytics dashboards, heatmaps, and real-time reports**, enabling data-driven decision-making. Overall, the project aims to reduce safety risks, improve event management, and set a foundation for future smart city and smart pilgrimage solutions.

REFERENCES :-

1. Shukla, H., Sharma, A., Khandelwal, Y., Jain, S., Gupta, S., & Soni, C. (2025). Leveraging Machine Learning for Real-Time Crowd Control and Safety at Kumbh Mela.
2. Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi object tracking by associating every detection box. In European conference on computer vision, pages 1–21. Springer, 2022.
3. Xiaomin Wu. Fastapi as a backend framework. 2024.
4. Shabana Habib, Altaf Hussain, Waleed Albattah, Muhammad Islam, Sheroz Khan, Rehan Ullah Khan, and Khalil Khan. Abnormal activity recognition from surveillance videos using convolutional neural network. Sensors, 21(24):8291, 2021.
5. Sanam Narejo, Bishwajeet Pandey, Doris Esenarro Vargas, Ciro Ro driguez, and M Rizwan Anjum. Weapon detection using yolo v3 for smart surveillance system. Mathematical Problems in Engineering, 2021(1):9975700, 2021.

Submitted by:

1. Nisarga Santosh Lokhande (Roll no. 24)
2. Rushikesh Santosh Landge (Roll no.22)
3. Gayatri Sahebrao Wadage (Roll no. 49)
4. Dhanashri Rajendra Sonawane (Roll no. 44)