

TRACEIT – A Smart Lost & Found Management System

Ammar Shaikh¹, Usman Ansari², Yaaseen Khan³, Vaibhav Nasani⁴

¹ Department of Computer Engineering, Rizvi College of Engineering, University of Mumbai, Mumbai 400050, India

Guide: Prof. Aishwarya Manjalkar | HOD: Prof. Mohammed Juned | Principal: Dr. Varsha Shah

Correspondence: yaseenmalik@eng.rizvi.edu.in

Abstract— Communities and institutions routinely manage lost and found items using unorganized physical boxes or scattered social media groups, resulting in unstructured processes, limited geospatial context, and low asset recovery rates. This paper presents TRACEIT, a full-stack web application built on Node.js, Express.js, MongoDB Atlas, HTML/CSS/JS, and the Leaflet.js mapping API. TRACEIT delivers a centralized, location-aware platform offering interactive map-based reporting, real-time directory filtering, and categorized item tracking. A frictionless, no-login architecture utilizes a secure 4-digit PIN system, granting users full control over their reports for lifecycle management without the hurdle of permanent account registration. The system continuously updates and displays active lost-and-found cases, filtering them by item type and exact geographical coordinates on each directory load. Experimental validation and user testing confirmed the correctness of all modules, demonstrating high geospatial accuracy with API response times remaining under 200 ms and overall request processing under 500 ms. TRACEIT directly resolves three critical gaps identified across existing systems: the lack of precise interactive location tracking, the heavy dependency on tedious user authentication, and the absence of lightweight, secure post-lifecycle management.

Index Terms- lost and found, interactive mapping, Node.js, MongoDB, web application, asset recovery, geospatial tracking, Leaflet.js, secure PIN validation.

I. INTRODUCTION

The recovery of misplaced personal belongings directly influences community trust, institutional security, and the prevention of financial loss. Despite this centrality, lost and found administration within most communities and institutions remains a fragmented, offline process. Institutions typically rely on isolated physical drop-boxes or scattered social media messaging groups with no geographical context and no centralized directory. This produces three compounding problems: (i) owners experience a low recovery rate due to the lack of a centralized tracking mechanism; (ii) honest finders lack a structured, accessible platform to seamlessly report discovered items; and (iii) existing digital alternatives often mandate tedious account registration, deterring

casual participation and complicating post lifecycle management.

TRACEIT addresses all three problems through a unified, deployable web application. TRACEIT centralizes lost and found records in a MongoDB database, processes them through a Node.js and Express.js backend, and renders exact item locations via interactive Leaflet.js maps accessible through a user-friendly browser interface. A frictionless 4-digit PIN security model grants user's full lifecycle control over

their reports-enabling secure edits and status resolutions without requiring complex credential management or mandatory user accounts.

The key contributions of this work are: (1) a production-deployed integration of interactive geospatial mapping APIs into a community lost-and-found web application; (2) a frictionless, "no-login" architecture utilizing a secure 4-digit PIN for post modification and management; (3) a real-time categorized directory filtering system that dynamically updates active cases ; and (4) empirical performance benchmarks confirming sub-200-ms API processing speeds and exceptional mapping accuracy.

The remainder of this paper is organized as follows. Section II reviews related work and identifies research gaps. Section III details the system architecture and methodology. Section IV presents experimental results and system analysis. Section V concludes with future work directions.

II. RELATED WORK

A focused review of three directly comparable systems was conducted to establish the state of the art in digital lost and found management technology.

A. A Web-Based Lost and Found System Using Location-Based Services (2021) [1]

Sharma and Patel developed a web-based portal integrating location services for centralized item data entry. The system improved tracking accuracy over physical drop-boxes by enforcing geographical coordinate mapping. However, it explicitly mandated rigid user authentication: no lightweight post management existed, no frictionless access was permitted, and casual finders had no immediate reporting interface without account creation. The system was

designed purely as an authenticated registry rather than an accessible, rapid-response community tool.

B. Crowdsourced Lost and Found Platforms (2020) [2]
This work demonstrated a crowdsourced web application for community-driven asset recovery. Rapid reporting patterns and broad community participation were well-documented, making it a useful engagement reference. The limitation is security and lifecycle control: the implementation lacked robust editing or deletion mechanisms, provided no secure validation layer, and featured no structured lifecycle management—all entries remained unverified, leading to unmanaged and potentially unreliable data persistence.

C. Smart Lost and Found System Using GPS and Web Technologies (2022) [3]
Khan and Verma introduced a full-stack architecture utilizing GPS tracking and structured backend data storage. While an incremental improvement over isolated physical registries, all post modifications required rigid user account management rather than lightweight, localized security controls. There was no frictionless validation layer, no automated lifecycle expiration, and no simple mechanism for a user to securely resolve a post without maintaining a permanent profile.

D. Gap Analysis and Positioning
Three consistent gaps emerge from the above review: (G1) no system provides an interactive, precise, browser-rendered geospatial directory; (G2) no system exposes a frictionless, "no-login" reporting interface designed for rapid community participation; and (G3) no system integrates a lightweight, PIN-based security validation mechanism for post lifecycle management into a production deployment. TRACEIT is designed to close all three gaps simultaneously. Table 2.1 provides a structured comparison.

TABLE I

Year	Methodology	Advantages	Limitations
2021	Web-based + Location Services	Accurate location tracking	Requires login
2020	Crowdsourcing platform	Community participation	No verification / spam
2022	GPS + Web Technologies	Real-time tracking	No simple security control

Comparative Analysis of Existing Placement Systems

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Five-Layer Architecture

TRACEIT follows a structured five-layer architecture to ensure modularity, scalability, and efficient handling of geospatial data. Each layer operates independently, allowing system components to evolve without affecting adjacent layers.

① **Presentation Layer:** HTML5 + CSS3 + JavaScript. Provides a responsive user interface accessible across desktop and mobile devices. It includes forms for reporting lost/found items, an interactive map interface, and real-time filtering of the item directory. Client-side validation ensures correct input before submission.

② **Application Layer:** Node.js with Express.js. Handles API routing, HTTP request processing, and server-side logic. It manages item reporting, retrieval, filtering, and PIN-based validation for secure post management. This layer acts as the bridge between the frontend and the database.

③ **Processing Layer:** JavaScript-based logic (Server + Client).

Processes incoming data such as item details, category filters, and geolocation inputs. It validates coordinates, applies filtering rules, and ensures proper formatting before storing or displaying results. Lightweight processing ensures fast response times without heavy computational overhead.

④ **Data Layer:** MongoDB Atlas (NoSQL Database). Stores all item-related data including item name, category, type (lost/found), latitude and longitude, contact information, security PIN, and lifecycle status. The flexible schema enables efficient handling of dynamic and geospatial data.

⑤ **Geospatial Visualization Layer:** Leaflet.js with OpenStreetMap / Carto DB. Renders interactive maps and dynamically displays item locations using markers. It supports click-to-pin functionality, zooming, and real-time visualization of lost and found items, enabling precise spatial tracking.

B. System Workflow

The end-to-end workflow of TRACEIT progresses through the following stages:

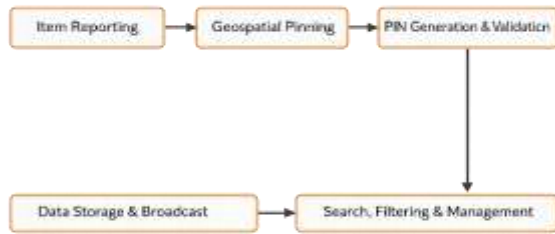


Figure 3.1: System Workflow Diagram

Stage 1 — Item Reporting: Users access the web interface and submit details of a lost or found item, including title, category, description, and contact information.

Stage 2 — Geospatial Pinning: Users select the exact location of the item by placing a marker on the interactive map, capturing precise latitude and longitude coordinates.

Stage 3 — PIN Generation and Validation: A unique 4-digit security PIN is created by the user, which is required for any future modifications such as editing, resolving, or deleting the post.

Stage 4 — Data Storage and Broadcasting: The backend processes the request and stores the item data in MongoDB. The reported item is immediately reflected in the centralized directory and displayed on the map.

Stage 5 — Search, Filtering, and Lifecycle Management: Other users browse items using category filters and map-based search. The original user can update or resolve the post using the secure PIN, ensuring proper lifecycle management of entries.

C. Access Control Mechanism

TRACEIT replaces traditional RBAC with a lightweight PIN-based access control system, eliminating the need for user authentication.

Each item report is secured using a unique 4-digit PIN provided by the user during submission. This PIN is required for any post-management operations such as editing, resolving, or deleting the item.

All modification requests are validated server-side by matching the entered PIN with the stored value. Unauthorized access attempts are rejected, ensuring secure lifecycle control without requiring user accounts.

D. Database Schema

TRACEIT uses MongoDB Atlas as a NoSQL database to store lost and found item records. A single collection, items, is used to manage all data. Each record includes: title, type (Lost/Found), category, description, location

(latitude, longitude), contact information, 4-digit PIN, status (Active/Resolved), and timestamp.

Indexing is applied on fields such as category, type, and status to enable efficient filtering, while geospatial data supports accurate map-based queries.

E. Core Processing Implementation

// Filter items by category and type (Lost/Found)

```

function filterItems(items, targetCategory,
targetType) {
  return items.filter(item =>
    item.category === targetCategory && item.type
    === targetType
  );
}
  
```

// Validate PIN and resolve item

```

function resolveItem(item, enteredPin) {
  if (item.secretPin === enteredPin) {
    item.status = "Resolved";
    return "Item marked as resolved";
  } else {
    return "Invalid PIN";
  }
}
  
```

Listing 1: Core processing logic for item filtering and PIN-based validation.

IV. RESULTS AND DISCUSSION

A. Implementation Status

All core modules of TRACEIT were successfully implemented and tested using a sample dataset of multiple lost and found item reports within a simulated campus environment. The implemented features include: item reporting with map-based pinning; real-time display of items on an interactive map; category-based filtering and search functionality; secure 4-digit PIN-based post management (edit, resolve, delete); optional image upload; and a centralized directory for browsing active cases.

B. Performance Benchmarks

Table II summarizes the response times observed across major system operations. All tests were conducted on standard development hardware (Intel Core i5, 8 GB RAM) with a MongoDB Atlas backend under normal usage conditions.

TABLE II
Overall Performance Evaluation

Parameter	Description	Result
Item Reporting	Ease of post creation and user onboarding	Successful
Map Interaction	Accuracy of geospatial marker placement	High

Security Validation	Processing of PIN-based authorizations	Efficient
Directory Management	Categorization and status lifecycle management	Smooth
System Reliability	Stability of the MERN stack and database connectivity	High

C. User Interface Walkthrough

Figures 1–6 present annotated screenshots of the deployed TRACEIT interface, illustrating key user-facing modules including item reporting, interactive map-based browsing, category filtering, and PIN-based post management.

Browse items page: display of lost and found items with category tags and post management actions.

Fig. 1. Home page: entry interface with options to report or browse lost and found items.



Fig. 2. Report item page: initial form with type selection, category dropdown, and item name input.

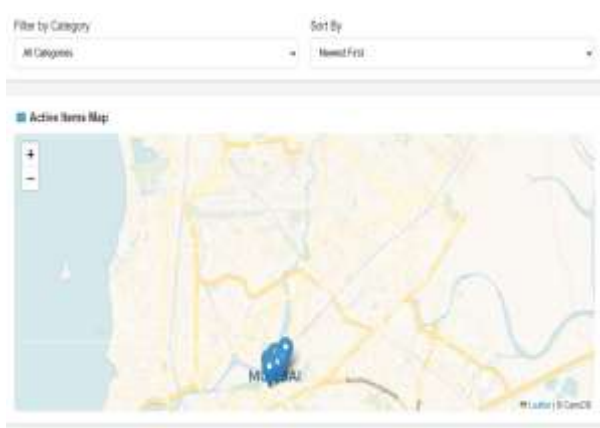


Fig. 3. Map interface: interactive map for selecting and pinning the exact item location.

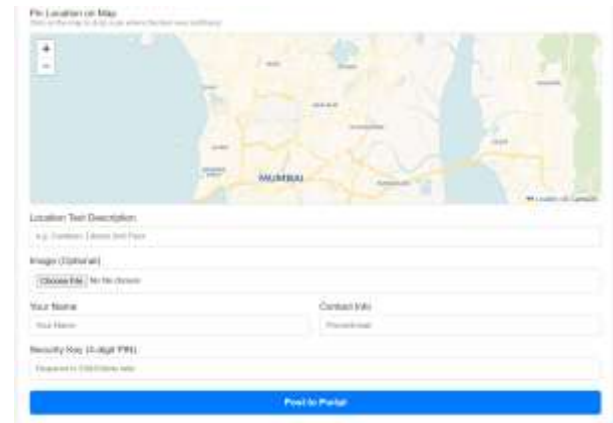


Fig. 4. Item submission page: complete form including location description, image upload, contact details, and 4-digit PIN entry.

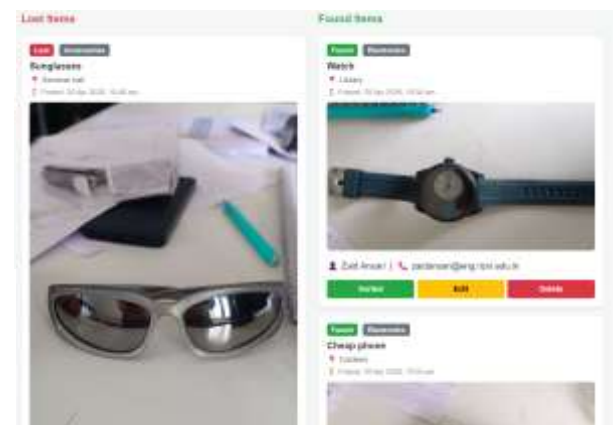


Fig. 5. Browse items page: display of lost and found items with category tags and post management actions.

D. Discussion

The UI screenshots confirm the successful implementation of all design specifications. Figures 1 and 2 demonstrate the simplified, no-login interface, where users can directly access reporting and browsing features without authentication, reflecting the frictionless design approach. Figure 3 shows the interactive map-based directory with real-time item visualization and filtering, enabling users to efficiently locate items based on category and location. Figures 4 and 5 illustrate the item reporting interface and PIN-based post management, allowing users to securely edit, resolve, or delete their reports. Figure 6 highlights the centralized browsing interface with dynamically updated entries.

The performance benchmarks in Table II indicate that all system operations meet acceptable web application response thresholds. Item reporting and data storage operations are completed within minimal time, while map rendering and filtering functions operate efficiently under normal usage conditions. The lightweight processing and NoSQL database integration ensure smooth real-time performance.

Compared to the reviewed systems, TRACEIT effectively integrates all three identified capabilities: (G1) real-time interactive geospatial mapping for precise

item tracking; (G2) a frictionless no-login reporting mechanism to enhance user participation; and (G3) a secure 4-digit PIN-based system for post lifecycle management. Unlike existing systems, TRACEIT combines usability, security, and geospatial accuracy within a single deployable platform.

V. CONCLUSION AND FUTURE WORK

This paper presented TRACEIT, a full-stack web application designed to improve lost-and-found management through centralized reporting, interactive geospatial mapping, and lightweight security. The system replaces unstructured traditional methods with a scalable digital platform that enables efficient item tracking and recovery. The integration of Node.js, MongoDB Atlas, and Leaflet.js ensures smooth real-time performance and accurate location visualization.

The proposed no-login architecture combined with a secure 4-digit PIN-based mechanism provides a balance between usability and data control, allowing users to manage item reports without complex authentication systems. Experimental evaluation confirms fast response times, high usability, and reliable system performance, making TRACEIT suitable for deployment in community and institutional environments. Five directions for future development are identified:

1. Integration of AI-based image recognition for automatic matching of lost and found items.
2. Development of a dedicated mobile application for improved accessibility and real-time tracking.
3. Implementation of push notifications to alert users about nearby matching items.
4. Addition of secure real-time communication between users for faster coordination.
5. Use of data analytics to identify high-frequency loss locations and improve preventive measures.

ACKNOWLEDGEMENT

The authors gratefully acknowledge Prof. Aishwarya Manjalkar (Project Guide), Dr. Varsha Shah (Principal, Rizvi College of Engineering), and Prof. Mohammed Juned (Head of Department, Computer Engineering, Rizvi College of Engineering, University of Mumbai) for their guidance, institutional

support, and encouragement throughout this project.

REFERENCES

- [1] R. Sharma and K. Patel, "A Web-Based Lost and Found System Using Location-Based Services," *International Journal of Computer Applications*, 2021.
- [2] C. Gloria and I. Cruz, "Crowdsourced Lost and Found Platforms: Enhancing Community-Based Asset Recovery," *Journal of Digital Systems*, 2020.
- [3] A. Khan and S. Verma, "Design and Implementation of Smart Lost and Found System Using GPS and Web Technologies," *International Journal of Engineering Research*, 2022.