TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52032

AI-Powered Lost and Found Portal for College Campus

Sumit Kumar Verma[1], Angad Singh[2], Abhay Sahu[3], Asmit Kumar Rauniyar[4], Pankaj Kumar Gupta[5]

[1,2,3,4]Scholar, Department of Computer Science & Engineering, Goel Institute of Technology & Management, Lucknow, Uttar Pradesh, India

[5]Assistant Professor, Department of Computer Science & Engineering, Goel Institute of Technology & Management, Lucknow, Uttar Pradesh, India

KEYWORD

Lost and found system, AI item matching, convolutional neural networks, real-time notifications, QR code, geolocation

ABSTRACT

Every year, countless students, faculty, and staff across college campuses lose personal belongings — and most of them never get those items back. Traditional lost-and-found systems, whether physical registers, notice boards, or informal WhatsApp groups, simply were not built for the pace and scale of modern campus life. They are slow, hard to search, and offer no intelligent way to connect lost items with found ones. This paper introduces Lost and Found — a full-stack, AI-powered web portal designed to completely rethink how institutions handle lost and found items. Built on the MERN stack (MongoDB, Express.js, React.js, and Node.js), the platform uses a TensorFlow.js convolutional neural network (CNN) to automatically match photos of lost items with found ones, achieving an 87.3% matching accuracy in real-world trials. Beyond AI matching, the system includes QR code tagging for physical items, GPS-based location tracking, role-based access for different types of users, real-time push notifications via WebSockets, and a live analytics dashboard for administrators. During a 60-day pilot at the Institute of Technology, Lucknow with 200 participants, the portal reduced average item recovery time by 74% compared to the paper-based system, and earned a System Usability Scale (SUS) score of 82.4 — placing it firmly in the "Excellent" category.

I. Introduction

Losing something on a busy university campus is frustratingly common. In an internal survey conducted at the Institute of Technology, Lucknow, nearly 68% of students admitted to losing at least one valuable item during the academic year — yet fewer than 30% ever got it back. The core problem? There is no good system in place. Lost-and-found cabinets managed by security staff, broadcast WhatsApp messages, and social media posts all fall short in similar ways: they are hard to search, they do not automatically match items to owners, and they can even expose personal contact information to strangers.

The good news is that modern technology makes a much better solution possible. Advances in deep learning, cloud infrastructure, and real-time browser communication have opened the door to a truly intelligent lost-and-found ecosystem. That is exactly what this paper presents: Lost and Found, a web-based portal that takes a human problem and applies smart, practical technology to solve it.

Corresponding Author: Sumit Kumar Verma, Scholar, Department of Computer Science & Engineering, Goel Institute of Technology & Management, Lucknow, Uttar Pradesh, India
Email: vermasumit8400@gmail.com

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52032

The platform automates item classification, uses AI to compare images of lost and found items, sends instant notifications across devices, and gives administrators a real-time view of what is happening across campus. Here is a summary of what makes it distinctive:

- A modular, scalable full-stack architecture purpose-built for institutional lost-and-found management.
- A TensorFlow.js CNN model that automatically matches lost and found item images by visual similarity.
- A QR code tagging system that links physical found items to their digital database records.
- Geolocation-powered item discovery with heatmap analytics to help administrators identify high-loss zones.
- A thorough usability and performance evaluation based on a live campus pilot.

The sections that follow cover related work (Section II), system architecture (Section III), implementation details (Section IV), evaluation results (Section V), and future directions (Section VI).

II. Literature Review

Research into digital lost-and-found systems has evolved considerably over the years, moving from basic web and mobile apps toward AI-driven and IoT-enabled solutions. Here is a look at where the field stands and where the gaps still lie.

A. Web and Mobile-Based Systems

Tan and Chong [1] built one of the earlier digital lost-and-found systems for Universiti Kebangsaan Malaysia, combining a web interface with an Android app. Students and staff could report and search for items through structured forms, which meaningfully reduced recovery times. The catch was that item matching was purely keyword-based — there was no way to identify an item from a photo.

Ibrahim et al. [2] took a similar approach for educational institutions, automating report and retrieval workflows, but users — especially first-timers — found the system hard to navigate, and there was no support for multiple languages. Joshi et al. [3] introduced Foundify, which added real-time notifications and a smarter matching algorithm, bringing recovery rates up compared to unstructured alternatives.

B. IoT and RFID Approaches

Alshammari et al. [4] showed that RFID technology could automatically detect and log items at campus checkpoints with impressive accuracy. The problem is that RFID requires significant hardware investment and only works with pre-tagged items — making it impractical for most real-world scenarios. Prashanth et al. [5] explored a mobile GIS-based approach using GPS coordinates to narrow down item locations, a concept that directly inspired the geolocation features in Lost and Found.

C. AI and Machine Learning Approaches

Kanjalkar et al. [6] built a Next.js/MongoDB application that let users list items with photos, but stopped short of any automated image analysis. The JETIR 2024 system [7] introduced CNN-based image categorization for item-type recognition, reporting improved search precision. Singh et al. [8] focused on security, using Django's ORM with encrypted audit logs and role-based access control. Lost and Found pulls together the best ideas from all of these works — structured data capture, image AI, real-time communication, and secure access — into one cohesive platform.

D. Where the Gaps Are

Despite the progress made by previous research, several important gaps remain unaddressed:

- Most systems rely on text-only search, without any image-to-image similarity matching.
- Very few provide real-time push notifications across both web and mobile devices.
- QR code-based physical tagging linked to digital item profiles has barely been explored.
- Geolocation heatmaps for administrative planning are essentially absent from existing systems.

Lost and Found was built specifically to close all four of these gaps.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52032

III. System Architecture & Design

A. How the System Is Structured

Lost and Found uses a classic three-tier client-server architecture, but with a few modern twists. The front end is a React.js Progressive Web Application (PWA) — meaning it can be installed and used like a native app. It communicates with a Node.js/Express.js RESTful API, which in turn reads from and writes to a MongoDB Atlas cloud database.

What sets the architecture apart are the two additional layers running alongside: a Socket.io WebSocket server for real-time notifications, and a TensorFlow.js AI engine for image matching. Images themselves are stored and served from Cloudinary's CDN, keeping load times fast.

B. How Data Is Organized

The MongoDB database uses four main collections. Users stores profile information, roles, and contact details. Items holds each report (type: lost or found), along with category, description, image URL, GPS location, timestamp, and current status. Matches links lost and found items together along with a similarity score. Notifications keeps track of alerts sent to users. A separate AuditLogs collection records all sensitive operations for compliance purposes.

C. Who Can Do What: Role-Based Access

The system defines three levels of access. Regular users can report items, view potential matches, and file claims. Staff verifiers can physically verify items at the lost-and-found counter and update their status in the system. Administrators have full access to everything — all item records, the analytics dashboard, and system settings. Each user receives a JWT token at login that carries their role, and every protected route checks that token before allowing access.

IV. Implementation

A. Reporting an Item

The item reporting form was designed to make the process as painless as possible. Users choose from a taxonomy of 24 categories, write a description, upload up to five photos, and specify when and where the item was lost or found — either by allowing the browser to access their location, or by dropping a pin on a map. Before photos are uploaded to Cloudinary, they are automatically compressed on the user's device to keep things snappy.

B. The AI Image Matching Engine

The AI matching engine is the heart of the platform. Every time a new item is submitted, a background worker thread kicks off a matching pipeline. First, the uploaded image is passed through a pre-trained MobileNetV2 model to extract a 1,280-dimensional feature vector — essentially a numerical fingerprint of the image. Then, the system computes the cosine similarity between that fingerprint and those of all unmatched items in the opposite category (lost vs. found):

$$\textbf{Similarity}(A, B) = (A \cdot B) / (\|A\| \times \|B\|)$$

Any pair scoring 0.82 or higher gets flagged as a potential match and logged in the Matches collection. Both the person who reported the item as lost and the one who found it receive an instant notification. The whole process takes about 120 milliseconds per image — fast enough that users never notice the wait.

MobileNetV2 was chosen because it strikes an excellent balance between accuracy and speed. It does not need a GPU to run well, making it practical for standard server hardware.

C. QR Code Tagging

When a found item is submitted, the system generates a unique QR code that encodes the item's database ID and a verification URL. Staff print this label and attach it to the physical item at the lost-and-found counter. Anyone who scans the QR code with a smartphone camera is immediately taken to that item's status page — no app required, no searching needed.

D. Real-Time Notifications

Notifications are delivered the moment something important happens. Using Socket.io, the server maintains a persistent WebSocket connection with every logged-in user. When a potential match is found, a status is updated, or

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52032

a claim is approved or rejected, the relevant user sees a notification in real time. For users who happen to be offline, a fallback email is dispatched via Nodemailer and SendGrid so they do not miss anything.

E. Geolocation and Heatmaps

Every item report stores GPS coordinates as a GeoJSON point in MongoDB. This allows efficient proximity queries — for example, finding all items reported within 500 meters of a given location. On the front end, a Leaflet.js interactive map shows item markers. Administrators get an extra layer: a heatmap that shows where items are most frequently lost across campus, helping them focus security efforts or launch awareness campaigns in hotspot areas.

F. Admin Analytics Dashboard

The dashboard gives administrators a live view of what is happening across the system. Using MongoDB aggregation pipelines, it surfaces key metrics like the number of active cases, average resolution time, category breakdowns, daily report volumes, and how often AI matches lead to successful recoveries. Charts are rendered with Chart.js. Automated cron jobs archive resolved cases after 90 days and generate monthly PDF summary reports without any manual effort.

Table I: Technology Stack Summary

Layer	Technology / Library	Purpose	Version
Front-End	React.js, Tailwind CSS, Leaflet.js, Chart.js	UI rendering, interactive maps, analytics charts	React 18.2, Leaflet 1.9
Back-End	Node.js, Express.js, Socket.io	REST API and real-time WebSocket server	Node 20 LTS, Express 4.18
AI Engine	TensorFlow.js, MobileNetV2	Feature extraction and cosine similarity matching	TF.js 4.12
Database	MongoDB Atlas, Mongoose ODM	Document storage with GeoJSON indexing	MongoDB 7.0
Authentication	JWT, bcryptjs	Stateless auth and password hashing	JWT 9.0, bcrypt 5.1
Image Storage	Cloudinary CDN	Image hosting, transformation, and delivery	SDK 2.4
QR Code	qrcode (npm)	Physical item label generation	1.5.3
Notifications	Socket.io, Nodemailer, SendGrid	Real-time and email fallback notifications	Socket.io 4.7
DevOps	Docker, GitHub Actions, Vercel, Render	Containerization, CI/CD, and deployment	Docker 25.0

V. Evaluation & Results

A. How the Study Was Set Up

The pilot ran on the Institute of Technology, Lucknow campus from January to March 2026. Two hundred participants took part — 148 undergraduate students, 32 postgraduate students, and 20 faculty and staff members. Half were assigned to use the traditional paper-based system; the other half used Lost and Found. Over 60 days, 312 unique lost-item events were recorded across both groups.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52032

B. AI Matching Performance

To rigorously evaluate the matching engine, the team assembled 500 verified lost-found image pairs spanning 12 item categories. The results were encouraging:

- Precision: 89.1%
- Recall: 85.6%
- F1-Score: 87.3%
- Average inference latency: 118 milliseconds

Performance was strongest for electronics (93.2% precision), which tend to look distinctive, and weakest for clothing (79.4%), which is expected given how variable textiles look even within the same category. A threshold sensitivity analysis confirmed that a cosine similarity cutoff of 0.82 gave the best balance between catching true matches and avoiding false positives.

C. Recovery Time: A Dramatic Improvement

Under the conventional system, it took an average of 6.8 days to recover a lost item. With Lost and Found, that dropped to just 1.8 days — a 73.5% improvement. Even more striking: 41% of recovered items were reunited with their owners within 24 hours, largely because the AI match notification prompted users to act immediately.

D. What Users Thought

Usability was assessed using the System Usability Scale (SUS), a well-validated 10-question survey. Lost and Found earned a mean SUS score of 82.4 (standard deviation 7.1), which puts it solidly in the "Excellent" category (anything above 80 qualifies). In open-ended feedback, users most frequently praised QR code scanning and real-time notifications as features they genuinely found useful. The most common requests for improvement were offline functionality and support for Hindi.

E. Technical Performance

The system was load-tested with Apache JMeter simulating 500 concurrent users. Even under that load, the 95th-percentile API response time stayed below 320 milliseconds. Database queries averaged just 45 milliseconds, thanks to compound indexes on category, status, and geolocation fields. Socket.io notification delivery averaged 62 milliseconds end-to-end — fast enough to feel instantaneous.

Table II: Comparative Analysis — Lost and Found vs. Existing Systems

Feature	SmartFind	Tan & Chong [1]	Joshi et al. [3]	Kanjalkar [6]
AI Image Matching	✓ CNN (87.3%)	✗	Partial	✗
Real-Time Notifications	✓ WebSocket	✗	✓	✗
QR Code Tagging	✓	✗	✗	✗
Geolocation + Heatmap	✓	✗	✗	✗
Role-Based Access Control	✓ 3-tier	✗	Partial	✓
Admin Analytics Dashboard	✓ Full Analytics	✗	Basic	✗
Recovery Time Reduction	73.5%	~40%	~55%	~35%
SUS Score	82.4 (Excellent)	74.1	79.0	71.5

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52032

VI. Conclusion & Future Work

Lost and Found demonstrates that a thoughtfully designed, AI-powered platform can make a meaningful, measurable difference in how institutions handle lost items. By combining CNN-based image matching, QR code physical tagging, real-time WebSocket notifications, and location-aware analytics, the system reduced item recovery time by 73.5% and earned an excellent usability rating in a real-world campus pilot.

The modular MERN stack architecture means that any component — the AI engine, the notification service, the data layer — can be independently upgraded or scaled as institutional needs grow. The open API design also makes it straightforward to integrate with existing campus ERP platforms and access-control systems.

Looking ahead, several enhancements are planned:

- Fine-tuning the MobileNetV2 model on a domain-specific dataset of lost items to improve recognition accuracy for clothing and accessories.
- Building offline-first PWA capabilities using IndexedDB and service workers, so the app works even in areas with poor connectivity.
- Adding Hindi and other regional language support to make the platform accessible to a wider audience.
- Exploring a federated deployment model that lets multiple campuses share a single cross-campus item database.
- Integrating blockchain-based provenance logging for high-value items like laptops and identity documents.

Acknowledgements

The authors sincerely thank the administration and student community of the Institute of Technology, Lucknow for their cooperation and participation during the pilot study. This work was conducted without any external funding.

References

[1] S. Y. Tan and C. R. Chong, "An Effective Lost and Found System in University Campus," Journal of Information and Technology Management (JISTM), vol. 8, no. 32, pp. 99-112, Sep. 2023.

[2] E. Ibrahim, F. Brot, and I. Fofana, "Online Web-Based Lost and Found System for Educational Institutions," International Journal of Advanced Computer Science and Applications, vol. 9, no. 6, pp. 215-222, 2018.

[3] K. Joshi et al., "Foundify: A Lost and Found Ticket System," International Journal of Innovative Research in Technology (IJIRT), vol. 12, no. 5, pp. 113-119, Oct. 2025.

[4] A. A. Alshammari, M. A. Razzaque, and M. A. Hossain, "A Novel Approach for Lost and Found System Using RFID Technology," Proc. IEEE Int. Conf. Consumer Electronics (ICCE), Las Vegas, NV, Jan. 2018, pp. 1-4.

[5] G. K. Prashanth et al., "Mobile Technology for Efficient Lost and Found Item Retrieval Using GIS-Based Approach," Proc. 3rd Int. Conf. Smart Systems for Electrical Sciences (ICSSES), Mar. 2025, pp. 1-5.

[6] P. Kanjalkar et al., "Lost and Found Web Application," International Journal for Research in Applied Science and Engineering Technology (IJRASET), vol. 12, no. 11, Nov. 2024. doi: 10.22214/ijraset.2024.65573.

[7] P. Desai and A. Mehta, "Automated Lost and Found System with Peer-to-Peer Communication," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 11, no. 5, May 2024, pp. 1-9.

[8] A. Singh, R. Pandey, and V. Joshi, "Secure Lost and Found Item Recovery System with Role-Based Access Control," IJRASET, vol. 12, no. 8, 2024.

[9] S. Kumar et al., "Web-Based Lost and Found System," MIT International Journal of Computer Science and Information Technology, vol. 11, no. 1, Dec. 2022.

[10] H. Zhao et al., "Design and Implementation of the Lost-and-Found System," Proc. 3rd IEEE IAEAC Conf., Chongqing, China, 2018, pp. 2442-2446.

[11] S. Khan and T. Rane, "Design of Role-Based Access Systems Using Django Framework," International Journal of Web Engineering (IJWE), vol. 5, no. 2, 2023.

[12] P. Hutson and J. Hutson, "Bridging the Digital Divide: Innovative Uses of QR Codes and NFC," ISRG Journal of Arts, Humanities and Social Sciences, vol. 2, no. 2, 2024.