Smart City Complaint Management System: A Web-Based Platform for Citizen Grievance Registration, Tracking and Resolution

Simhadri N, Shrishail
Department of Information Science and Engineering, Aditya College of Engineering and Technology, Bengaluru, India
Under the guidance of **Asst. Prof. Deepika Verma**, Department of Information Science and Engineering
Corresponding email: simhadrinsimhadrin02@gmail.com

Abstract—Urban areas frequently experience civic issues such as damaged roads, overflowing garbage, blocked drainage, water leakage, faulty streetlights and other public-service problems. When citizens do not have a convenient and transparent channel to report these issues, complaints may be delayed, difficult to monitor or repeatedly submitted through disconnected communication methods. The Smart City Complaint Management System is proposed as a web-based application that provides a centralized interface for citizens and municipal administrators to register, manage and monitor civic complaints. Citizens can create an account, submit a complaint with a description, category and supporting photograph, and view its progress using a complaint reference. An administrator can review submitted complaints, manage records, update complaint status and communicate progress through the system. The proposed application follows a modular client-server architecture using the MERN technology stack: MongoDB, Express.js, React.js and Node.js. Authentication, role-based access, complaint data management and image-upload handling are considered as core system functions. This paper describes the problem background, related grievance-management practices, methodology, proposed architecture, modules, security considerations and an evaluation approach. The effectiveness of the system is to be assessed through functional testing and usability evaluation after implementation.

Keywords—smart city, complaint management, e-governance, citizen grievance, MERN stack, complaint tracking

I. INTRODUCTION

The growth of urban populations increases the demand for reliable public services and effective civic administration. Municipal authorities are responsible for maintaining essential infrastructure and services, including roads, sanitation, drainage, water supply and street lighting. Citizens who identify a public issue need a clear way to report it and receive information about the action taken. In practice, complaints may be communicated through visits to offices, telephone calls, emails or separate online channels. These methods can make it difficult for citizens to follow the progress of a complaint and for administrators to maintain a consolidated record.

Digital grievance platforms provide an organized way to receive, record and monitor public complaints. For example, the Government of India's Centralised Public Grievance Redress and Monitoring System (CPGRAMS) allows citizens to lodge grievances online and track their status using a unique registration number [1]. This demonstrates the value of online submission and traceability in public grievance handling. The proposed Smart City Complaint Management System applies similar core principles to a project-level civic complaint workflow, with emphasis on a simple interface for reporting local urban issues.

The main objective is to design and develop a web application through which citizens can submit civic complaints, attach supporting images and monitor progress, while authorized administrators can review and update complaint records. The system is intended to reduce dependence on unstructured reporting channels and provide a consistent digital record of each complaint.

The remainder of the paper is organised as follows. Section II reviews related practices and states the research gap. Section III presents the methodology: requirements, architecture, modules, workflow, technologies and the testing approach. Section IV states the results available at this stage, Section V discusses expected outcomes and advantages, and Section VI concludes with limitations and future enhancements.

A. Objectives

- To provide a centralized online platform for registering civic complaints.
- To allow citizens to submit complaint details, select a category and attach a supporting image.

- To generate or maintain a unique complaint reference for tracking.
- To provide authorized administrators with tools to view, manage and update complaints.
- To display complaint status and relevant updates to the citizen.
- To maintain complaint information in a structured database and support future expansion.

B. Existing Approach and Its Limitations

In a conventional complaint-reporting process, citizens may contact municipal offices directly, make telephone calls, send emails or use different service-specific portals. Although these channels can be useful, the information may be distributed across multiple records and communication methods. Citizens may not always receive a single reference number or a convenient view of complaint progress. Administrators may also need to consolidate complaint details manually. The main limitations are the following.

- Complaint information may be distributed across different communication channels.
- Citizens may have difficulty checking the current status of a submitted issue.
- Photographic evidence may not be consistently attached to complaint records.
- Manual record keeping can increase administrative effort.
- Duplicate or incomplete complaints may be difficult to identify.
- Complaint history and service-level analysis may be difficult without centralized records.

II. LITERATURE REVIEW

Digital public grievance systems provide useful reference models for designing citizen-facing complaint applications. The following review summarizes relevant practices and technical guidance that inform the proposed system.

A. Online Public Grievance Platforms

The CPGRAMS portal provides an online channel for citizens to lodge service-related grievances with public authorities. It supports status tracking through a unique registration ID and includes feedback and appeal-related facilities [1], [5]. These features illustrate the importance of complaint identification, traceability and a defined grievance-handling workflow. The proposed project adopts the basic concepts of online submission and status tracking for local civic complaints.

B. Complaint Registration and Workflow Management

A complaint-management application generally requires structured information such as the complainant, issue category, description, date, location, evidence and current status. A consistent workflow helps administrators review incoming complaints and record progress. For a civic system, categories may include roads, garbage collection, drainage, water supply and streetlights. The exact categories and workflow should be configurable according to the needs of the intended municipality or project demonstration.

C. MERN Stack for Web Application Development

MongoDB, Express.js, React.js and Node.js form the MERN stack. MongoDB provides document-oriented data storage, React.js supports the user interface, and Express.js with Node.js supports server-side routes and application logic. MongoDB's documentation describes the use of the MERN stack for applications that combine a React interface with server-side services and database operations [2]. This separation supports modular development and maintenance.

D. Secure Handling of Uploaded Evidence

Complaint photographs can help administrators understand the reported issue. However, file uploads introduce security risks if file types, sizes and access permissions are not controlled. OWASP recommends validating allowed file types, limiting file size, generating safe filenames and restricting upload access to authorized users [3]. These recommendations are relevant to the evidence-upload module of the proposed application.

E. Research Gap and Proposed Approach

Existing grievance platforms demonstrate online complaint submission and tracking, while general web technologies provide reusable components for authentication, data storage and file handling. The proposed project focuses on integrating citizen registration, complaint submission with image evidence, administrator-side complaint management and status tracking in one project-level web application. It is designed as a modular system that can be extended with location mapping, notifications, analytics and department-wise assignment in future versions.

III. METHODOLOGY

The development methodology consists of requirement identification, system design, module development, database integration, testing and deployment preparation. The application is organized into a frontend, backend API and database layer.

A. Requirement Analysis

The primary users are citizens and administrators. Citizens require registration and login, complaint

submission, image attachment, complaint-history viewing and status tracking. Administrators require secure login, access to submitted complaints, complaint review and status management. The application also requires input validation, authentication and controlled access to complaint records.

B. Proposed System Overview

The Smart City Complaint Management System is proposed as a centralized web application for recording and monitoring civic complaints. A citizen can create an account, log in, submit a complaint by entering its category and description, attach an image where appropriate and view the complaint's status. An administrator can access the management dashboard, inspect submitted records and update their progress. Each complaint should have a unique reference so that it can be identified and tracked.

C. System Architecture

The proposed architecture follows a three-layer client-server model (Fig. 1). The React frontend collects user input and displays complaint information. The Node.js and Express.js backend exposes API endpoints, validates requests, applies authentication and processes complaint operations. MongoDB stores user accounts and complaint records. An image-upload middleware handles evidence files subject to file validation and size restrictions.

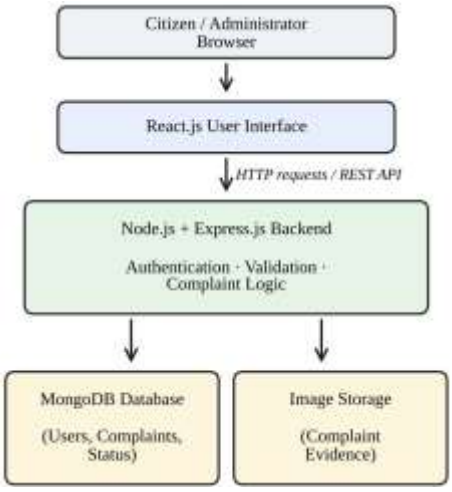


Fig. 1. High-level architecture of the proposed system.

D. Major Modules

1) *User registration and login.* Allows citizens to create accounts and authenticate before accessing personal complaint functions. Passwords must be stored using a secure password-hashing method.
2) *Citizen dashboard.* Displays the citizen's submitted complaints and their current statuses, helping the user review complaint history.
3) *Complaint submission.* Collects complaint category, title, description and relevant location details. The user may attach a supporting photograph.
4) *Complaint tracking.* Allows the citizen to view complaint details and status using the complaint reference or the authenticated complaint-history page.
5) *Administrator login.* Restricts administrative functions to authorized users through role-based access control.
6) *Admin dashboard.* Displays complaint records and supports filtering or reviewing complaints by category, date or status, depending on implementation.
7) *Complaint status management.* Allows an administrator to update a complaint through defined states such as Submitted, Under Review, In Progress, Resolved or Rejected. Status changes should be recorded with timestamps.
8) *Data management.* Stores user and complaint records in MongoDB and supports retrieval and updates through backend API endpoints.

E. Proposed Complaint Workflow

1) The citizen registers or logs in to the application.
2) The citizen opens the complaint form and enters the issue details.
3) The citizen selects a complaint category and optionally attaches an image.
4) The backend validates the submitted information and stores the complaint.
5) The system assigns or records a unique complaint reference.
6) The administrator reviews the complaint and updates its status as work progresses.
7) The citizen checks the complaint history or tracking view to see the latest status.

F. Requirements

1) Functional Requirements

- The system shall allow a citizen to register and log in.
- The system shall allow an authenticated citizen to submit a complaint.
- The system shall validate required complaint fields and uploaded evidence.
- The system shall store complaint records and associate them with the submitting user.
- The system shall allow citizens to view their own complaint records and statuses.
- The system shall allow authorized administrators to view and update complaint records.

2) Non-Functional Requirements

- *Security:* authentication, role-based authorization and safe file handling.
- *Usability:* clear navigation, readable forms and understandable status labels.
- *Reliability:* validation and error handling for failed or incomplete requests.
- *Maintainability:* modular frontend, backend routes and data models.
- *Scalability:* architecture that can be extended to more users, complaint categories and departments.

G. Design Details

1) User Interface

The frontend is planned using React.js to provide separate views for citizens and administrators. The citizen interface includes registration, login, complaint submission and complaint-history pages. The administrator interface provides a dashboard for viewing complaint records and updating their status. Forms should provide clear labels, validation messages and responsive layouts.

2) Backend and API

The backend is planned using Node.js and Express.js. API routes handle account registration, login, protected user operations, complaint creation, complaint retrieval and administrator updates. Authentication middleware verifies access credentials before protected operations are performed. Server-side validation is required even when frontend validation is also provided.

3) Database Design

MongoDB is used to store application data [4]. A User collection may contain a user's name, email, securely hashed password and role. A Complaint collection may contain a complaint reference, citizen identifier, category, title, description, image path, submission date, location details where available, status and update information. The database design should maintain a relationship between each complaint and its submitting user.

4) Complaint Image Upload

The complaint form may allow a citizen to attach a photograph that supports the report. The server should validate the file extension and actual file type, apply a maximum file-size limit, generate a safe stored filename and ensure that only permitted users can upload files. These controls follow the security principles described by OWASP [3].

H. Technologies

The project is planned around the MERN stack, which provides a JavaScript-based development environment across the frontend and backend [2]. The final technology versions and hosting configuration will be documented once implementation is complete. Table I summarises the technologies and their purposes.

TABLE I. TECHNOLOGY AND PURPOSE SUMMARY

Technology	Purpose
React.js	Frontend components, forms, dashboards and navigation.
Node.js	JavaScript runtime for backend execution.
Express.js	REST API routes, middleware and request handling.
MongoDB	Storage of user accounts and complaint records.
Multer / upload middleware	Handling complaint image uploads, with validation controls.
JWT or session-based authentication	Maintaining authenticated access to protected functions.
HTML, CSS and JavaScript	Web page structure, styling and client-side interactions.

I. Testing and Evaluation Approach

Functional testing should verify registration, login, protected access, complaint submission, image upload, complaint listing, status updates and complaint tracking. Negative tests should include invalid credentials, missing required fields, unsupported file types, oversized files and attempts to access administrator functions using a citizen account. Usability evaluation may collect feedback on navigation clarity, form completion and ease of tracking. No numerical performance or user-satisfaction result is claimed in this paper; such results should be added only after actual testing.

IV. RESULTS

The system is presented as a proposed design. No implementation results, test measurements or user feedback are reported, and no quantitative results are claimed. The expected outcomes and the planned evaluation are discussed in the next section.

V. DISCUSSION

A. Expected Outcomes

The proposed system is intended to provide a consistent workflow from complaint submission to administrative status updates. Its main expected outcomes are a centralized complaint repository, easier access to complaint history and improved visibility of complaint progress for citizens. The administrator dashboard is intended to support organized review of incoming complaints.

B. Evaluation Plan

The implementation should be evaluated using a test-case matrix. At minimum, testing should cover successful and unsuccessful registration and login, role restrictions, complaint creation with valid and invalid inputs, image upload validation, citizen complaint-

history access and administrator status updates. Response time, error rate and user feedback may also be measured in a later evaluation. Since no verified test measurements are available, no quantitative results are reported.

C. Advantages

- Centralized complaint registration and record management.
- Improved transparency through complaint reference and status tracking.
- Convenient online access for citizens.
- Photographic evidence can support complaint review.
- Role-based administration separates citizen and administrator functions.
- Structured data can support future reporting and analysis.
- Modular design allows future integration with maps, notifications and municipal departments.

VI. CONCLUSION

The Smart City Complaint Management System proposes a web-based approach to organizing civic complaint registration, evidence submission, administrative review and status tracking. By bringing these activities into a single application, the system aims to make complaint handling more accessible to citizens and more structured for administrators. The MERN-based architecture separates the user interface, backend services and database, allowing the application to be maintained and extended. Authentication, role-based access and secure image-upload validation are important considerations for protecting user and complaint information.

A. Limitations

- The work is a proposed design; no implementation results, test measurements or user feedback are reported.
- The effectiveness of the system has not been assessed and remains to be established through functional testing and usability evaluation.
- Integration with official municipal systems would require suitable authorization and cooperation from the relevant public authorities.

B. Future Enhancement

Future enhancements may include map-based complaint location selection, GPS coordinates, email or SMS notifications, department-wise complaint assignment, escalation rules for overdue complaints, duplicate-complaint detection, analytics dashboards and a mobile application.

ACKNOWLEDGMENT

The authors thank Asst. Prof. Deepika Verma, Department of Information Science and Engineering, Aditya College of Engineering and Technology, for guidance throughout this project.

REFERENCES

[1] Government of India, “Centralised Public Grievance Redress and Monitoring System (CPGRAMS),” Official Public Grievance Portal. [Online]. Available: https://pgportal.gov.in/. Accessed: Nov. 11, 2026.

[2] MongoDB, “Integrate MongoDB with React: MERN Stack Application,” MongoDB Documentation. [Online]. Available: https://www.mongodb.com/docs/drivers/node-frameworks/react/. Accessed: Nov. 11, 2026.

[3] OWASP Foundation, “File Upload Cheat Sheet,” OWASP Cheat Sheet Series. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html. Accessed: Nov. 11, 2026.

[4] MongoDB, “MongoDB Node.js Driver Documentation,” MongoDB Documentation. [Online]. Available: https://www.mongodb.com/docs/drivers/node/current/. Accessed: Nov. 11, 2026.

[5] Ministry of Information and Broadcasting, Government of India, “Grievance Redressal Mechanism,” CPGRAMS information page. [Online]. Available: https://mib.gov.in/sites/default/files/2025-06/cpgram_1.pdf. Accessed: Nov. 11, 2026.