WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Civic Eye: AI and IoT-Based Civic Complaint System

Prof. Roshni Wasnik[1], Vasihnavi Padole[2], Rucha Kosare[3], Sagar Gajbhiye[4], Pritam Chikate[5], Yash Baisware[6]

[1]Dept. of Computer Science & Engg. Jhulelal Institute of Technology Nagpur, India r.wasnik@jitnagpur.edu.in
[2]Dept. of Computer Science & Engg.Jhulelal Institute of Technology Nagpur, India vaishnavipadole786@gmail.com
[3]Dept. of Computer Science & Engg. Jhulelal Institute of Technology Nagpur, India ruchakosare09@gmail.com
[4]Dept. of Computer Science & Engg. Jhulelal Institute of Technology Nagpur, India gajbhiyesagar1909@gmail.com
[5]Dept. of Computer Science & Engg. Jhulelal Institute of Technology Nagpur, India Pranavchikate56@gmail.com
[6]Dept. of Computer Science & Engg. Jhulelal Institute of Technology Nagpur, India yashbaisware29@gmail.com

ABSTRACT—

Rapid urbanisation has placed sustained pressure on municipal bodies, while conventional complaint redressal channels remain slow, opaque, and difficult to audit. Citizens seldom receive feedback on reported civic issues, and authorities lack a consolidated, prioritised view of problems across a city. This paper presents Civic Eye, a web-based smart civic complaint management platform that integrates artificial intelligence and Internet of Things (IoT) monitoring to streamline the complete complaint lifecycle. Citizens report issues such as potholes, garbage overflow, water leakage, drainage blockage, flooding, air quality concerns, and streetlight faults along with a description and a map-based geo-location. An AI classification engine categorises each complaint, assigns a priority level, estimates the resolution time, and routes it to the responsible municipal department. Administrators assign officers, officers update progress and submit resolution proof, and citizens verify the outcome before a complaint is closed. The system is implemented using a React and Vite front end, a Node.js and Express back end, and a Supabase PostgreSQL database secured through role-based access control and row-level security, and it is complemented by an IoT module that raises threshold-based alerts for smart bins, water-level sensors, air quality sensors, and streetlights. Functional testing of the prototype shows that representative complaints in all eight supported categories are correctly classified, prioritised, and mapped to the responsible department, while a seven-stage verified workflow ensures transparent accountability. The results affirm the practical viability of deploying intelligent, citizen-centric platforms for transparent and efficient urban civic governance.

Keywords—Smart City, Civic Complaint Management, Artificial Intelligence, Internet of Things, Citizen Engagement, E-Governance, Role-Based Access Control, Geo-tagging, Supabase

I. INTRODUCTION

Rapid urban growth, rising population density, and ageing civic infrastructure have collectively increased the volume and diversity of everyday problems that municipal authorities must address, including damaged roads, overflowing garbage, water leakages, blocked drains, waterlogging, polluted air, and faulty streetlights. In developing economies such as India, these challenges are amplified by limited administrative capacity, fragmented departmental responsibilities, and a heavy reliance on manual, paper-based, or telephone-based complaint handling. The resulting delays carry tangible consequences for public safety, public health, and citizens' trust in local governance.

A civic complaint management system is a platform through which citizens report issues in their neighbourhood and authorities track, resolve, and close them. An effective platform must capture the issue type, a clear description, and an accurate location, and must move each complaint through a well-defined lifecycle from submission to verified closure. Each complaint should also be associated with a priority level, a responsible department, and an expected resolution time so that scarce resources are directed to the most urgent problems first. Transparent status tracking allows citizens to follow progress and enables administrators to measure departmental performance objectively.

Conventional grievance channels rely predominantly on manual registration and manual routing, which, while familiar, offer no automated triage, no real-time visibility, and no independent confirmation that a reported issue has actually been resolved. The integration of artificial intelligence and IoT technologies represents a significant shift, enabling automatic complaint classification and prioritisation while allowing city infrastructure itself to report problems, such as an overfilled bin or a rising water level, before any citizen files a complaint. Among the available web technologies, cloud-hosted relational databases with row-level security have emerged as particularly effective for multi-role public service applications.

This research focuses on developing and evaluating Civic Eye, a role-based smart civic complaint management platform for citizens, municipal officers, and administrators. The proposed framework encompasses geo-tagged complaint submission, AI-assisted classification and routing, a verified resolution workflow, IoT-based monitoring, and analytical dashboards. The remainder of this paper is structured as follows: Section II describes the overall

methodology; Section III elaborates the AI classification engine; Section IV presents the problem statement and hypothesis; Section V details the system architecture; Section VI discusses experimental results; Sections VII and VIII address advantages and future scope; and Section IX concludes the paper.

II. METHODOLOGY

The Civic Eye platform is developed as a structured, iterative pipeline covering requirement analysis, data acquisition, validation, intelligent processing, implementation, and testing. Each phase is carefully designed to ensure the reliability, security, and maintainability of the resulting civic governance system.

A. Requirement Analysis and Data Collection

The system is designed around three user roles: citizens who report and verify issues, officers who resolve them, and administrators who supervise the whole process. Complaint data is captured directly from citizens through a submission form that records the issue category, a free-text description, an optional photograph, a postal address, and latitude and longitude coordinates selected on an interactive Leaflet map with OpenStreetMap-based address search covering locations across India. In addition, IoT devices of four types, namely smart garbage bins, water-level sensors, air quality sensors, and smart streetlights, report readings together with a unit and timestamp. All records are stored in a Supabase PostgreSQL database comprising eight tables covering profiles, departments, complaints, status history, IoT devices, IoT readings, IoT alerts, and notifications.

B. Data Preprocessing and Validation

Raw citizen input is frequently incomplete, inconsistent, or ambiguous, so a systematic validation pipeline is applied before any record is stored. Mandatory fields, namely category, description, and location, are checked, and descriptions shorter than five characters are rejected. Free-text category labels are normalised to one of eight canonical issue types through keyword matching, with unrecognised values mapped to a general category. Latitude and longitude values are parsed and validated, and a default city-centre coordinate pair is substituted when valid coordinates are unavailable, ensuring that every complaint can be plotted on the map. Each request is authenticated through a signed JSON Web Token and the user's profile is checked to be active, while rate limiting protects the authentication endpoints and secure HTTP headers are applied to all responses.

C. Feature Extraction and Priority Rules

Textual features are extracted from the complaint description through case-normalised keyword analysis, with terms such as pothole, garbage, leak, drain, flood, smoke, and light acting as discriminating indicators for the corresponding issue categories. Rule precedence is defined so that overlapping vocabulary is resolved deterministically. Based on the identified category, a priority level is assigned, with potholes, water leakage, and drainage issues rated High, flooding and air quality issues rated Critical, and the remaining categories rated Medium, while a citizen may explicitly override the priority. A category-specific estimate of the resolution time, ranging from 8 to 24 hours, is attached to each complaint, and the category is finally mapped to one of six municipal departments.

D. System Implementation and Testing

The front end is implemented using React 19, Vite, Tailwind CSS, React Router, Recharts, and React-Leaflet, and the back end is implemented as a Node.js and Express REST API organised into authentication, complaint, dashboard, IoT, notification, and officer modules. Supabase provides authentication and the PostgreSQL database, in which row-level security policies, database triggers, and indexes enforce role-specific data access, automatic complaint numbering, and timestamp maintenance. System behaviour is assessed through scripted test suites that exercise authentication, multi-user access, complaint submission, the complaint lifecycle, and location handling across India, together with direct verification of the classification engine and the IoT threshold logic.

III. AI-BASED COMPLAINT CLASSIFICATION

A. AI Classification Engine

The AI classification engine is the component that converts an unstructured citizen complaint into a structured, actionable record. On submission, the complaint description, location, and image reference are passed to an analysis service that returns the predicted issue classification, a confidence value, a priority level, and an estimated resolution time in hours. These outputs are stored alongside the complaint and are subsequently displayed to citizens, officers, and administrators, so that every stakeholder can see how the system has interpreted the report.

A defining characteristic of the engine is its deterministic, ordered rule evaluation. The lower-cased description is tested sequentially against category vocabularies, and the first matching rule determines the class, which avoids ambiguous multi-label assignments and guarantees reproducible outputs for identical inputs. Complaints that match no rule are assigned to a general category rather than being rejected, ensuring that no citizen report is lost. In the present prototype this service operates as a rule-based demonstration module, and the confidence value is a fixed placeholder of 0.94 rather than a learned probability.

The engine design offers several attributes that make it well suited for civic complaint processing. It is lightweight and returns results instantly without any training data, which is essential for a newly deployed system with no historical complaint corpus. Because the analysis is isolated behind a single service interface, the rule-based module can be replaced by a trained natural language processing model or an image-based convolutional classifier

without altering the controllers, database schema, or user interface. The classification output also transparently drives downstream automation, namely department routing, priority escalation, and notification generation.

The resulting category-wise decision rules are summarised as follows. Potholes are rated High priority with an estimated 18 hours, garbage overflow Medium with 8 hours, water leakage High with 12 hours, drainage blockage High with 16 hours, flooding Critical with 24 hours, air quality complaints Critical with 20 hours, streetlight faults Medium with 10 hours, and other issues Medium with 14 hours. These values encode the relative urgency of public safety and health risk, so that flooding and pollution reports are escalated to the top of the administrative queue, consistent with the expectation that such events pose the greatest immediate danger to citizens.

IV. PROBLEM STATEMENT AND HYPOTHESIS

A. Problem Statement

The rapid acceleration of urbanisation across Indian cities has led to a measurable and sustained increase in the number of civic issues reported by citizens. Existing grievance mechanisms are largely confined to manual registration, telephone helplines, or isolated web forms that offer no automated categorisation, no priority-based triage, and little visibility into what happens after a complaint is lodged. This reactive and opaque paradigm severely constrains the effectiveness of municipal response, since critical issues such as flooding or open drains are handled in the same queue as minor requests, and citizens are never asked whether the problem was truly resolved.

There is a pressing need for an intelligent platform capable of capturing complaints with accurate location data, classifying and prioritising them automatically, routing them to the correct department, and holding authorities accountable through verifiable resolution. The principal challenge lies in constructing a secure, role-aware system that coordinates citizens, officers, and administrators across a multi-stage workflow while also exploiting IoT sensor data to detect infrastructure problems proactively. This work addresses the challenge through the Civic Eye platform, combining an AI classification engine, a verified complaint lifecycle, and IoT-based alerting.

B. Hypothesis

The central hypothesis of this research posits that a unified platform combining automated AI-assisted triage, geo-tagged reporting, a citizen-verified resolution workflow, and IoT-driven alerts can make civic complaint handling faster, more transparent, and more accountable than conventional manual channels. The objective of this hypothesis is to establish that automated classification, priority assignment, and department routing at the time of submission remove the need for manual sorting, while citizen verification before closure ensures that complaints are closed only when problems are genuinely resolved. Furthermore, a role-based architecture with database-level security is better suited to handle the multi-stakeholder data structures of public service delivery. Through systematic functional evaluation of the implemented prototype, this study provides initial evidence for these claims.

V. SYSTEM ARCHITECTURE AND WORKING

A. System Architecture

The Civic Eye system is organised as a layered modular architecture, with each layer responsible for a distinct stage of the complaint lifecycle. The Presentation Layer is a React single-page application that serves as the system entry point and provides role-specific dashboards for citizens, officers, and administrators, together with complaint forms, an interactive Leaflet map, charts, and alert views. The Authentication Layer uses Supabase Auth and signed JSON Web Tokens to identify users and attach one of three roles to every request.

The Application Layer is an Express REST API that exposes six route groups for authentication, complaints, dashboard statistics, IoT readings, notifications, and officer management, with middleware enforcing authentication, role authorisation, security headers, and centralised error handling. The AI Service Layer hosts the classification engine described in Section III, which returns the category, priority, and estimated resolution time for each submission. The Data Layer is a Supabase PostgreSQL database with row-level security policies, triggers for automatic complaint numbering and timestamps, and a status history table that records every transition. The IoT Module receives sensor readings, compares them with device thresholds, and generates alerts, while the Notification Module informs administrators of new complaints.

B. Working of the System

The operational workflow initiates when a citizen signs in and submits a complaint with its category, description, and map-selected location. The server validates the input, passes it through the AI classification engine, resolves the responsible department, assigns a unique complaint number of the form CIV-XXXXXX, and stores the record with status Open together with an initial history entry. A notification is raised for the administrator, who reviews the complaint and assigns it to a department and an officer, moving it to the Assigned state.

The assigned officer then moves the complaint to In Progress, records resolution notes, marks the resolution as submitted, and uploads resolution proof, after which the complaint enters Verification Pending. The citizen reviews the outcome and either verifies it, which closes the complaint and records the closing time, or rejects it with feedback, which reopens the complaint for further action. Throughout, every status change is written to the history table and shown as a timeline, and the IoT module concurrently updates device states as Normal, Warning, or Critical. The architecture is designed for extensibility, supporting future integration of live sensor streams, trained machine learning models, and mobile applications.

VI. EXPERIMENTAL RESULTS

The Civic Eye prototype was evaluated through functional testing of its back-end services, database layer, and workflow logic. The implemented system comprises eight database tables created through eleven migration scripts, six API route groups, three user roles, six municipal departments, eight supported issue categories, seven complaint status stages, and four IoT device types. Scripted test suites covering authentication, multi-user access, complaint submission, the complete complaint lifecycle, and location handling across Indian cities were developed to exercise these components end to end.

The AI classification engine was verified using eight representative complaint descriptions, one for each supported issue category. In every case the engine returned the intended category together with the priority level and resolution-time estimate defined by the rule table. The IoT threshold logic was verified on seeded devices: a smart bin at 95% against a 90% threshold is flagged as Warning, a water level of 78 cm against a 60 cm threshold is flagged as Critical because it exceeds 120% of the threshold, and an air quality reading of 42 against a threshold of 50 remains Normal. These results confirm that the rules are applied consistently and that alert states are derived automatically from sensor values.

Table I presents the category-wise classification outcome of the AI engine, including the assigned priority and the estimated resolution time. The engine escalates flooding and air quality complaints to Critical priority, rates potholes, water leakage, and drainage issues as High, and assigns the remaining categories Medium priority, with estimated resolution times ranging from 8 hours for garbage overflow to 24 hours for flooding.

TABLE I. Complaint Category Classification Results

Sr.	Issue Category	Priority	Est. Time (hrs)
1	Pothole	**High**	18
2	Garbage Overflow	**Medium**	8
3	Water Leakage	**High**	12
4	Drainage	**High**	16
5	Flood Water Level	**Critical**	24
6	Air Quality	**Critical**	20
7	Streetlight Fault	**Medium**	10
8	Other	**Medium**	14

Each category is further mapped to a responsible department: potholes and other issues to the Road Department, garbage overflow to Sanitation, water leakage to Water, drainage and flooding to Drainage, streetlight faults to Electrical, and air quality complaints to the Environment Department. The seven-stage workflow records every transition in the status history table, and a rejected resolution returns the complaint to the In Progress state, so that closure always depends on citizen confirmation. The present rule-based engine does not yet report a statistical accuracy, which is left for evaluation on a labelled complaint dataset.

VII. ADVANTAGES

The proposed Civic Eye platform delivers a comprehensive set of practical and technical advantages over conventional complaint handling approaches.

• Automated Triage: Every complaint is classified, prioritised, and routed to the responsible department at the moment of submission, removing manual sorting and reducing the delay before work begins.

• Transparent Tracking: A complete status history and timeline allows citizens to follow each complaint from submission to closure, improving trust in municipal services.

• Citizen-Verified Closure: A complaint is closed only after the citizen confirms the resolution, and a rejection reopens it, ensuring genuine accountability from the departments.

• Geo-tagged Reporting: Map-based location selection with address search across India provides precise coordinates, enabling accurate field dispatch and spatial visualisation of problem areas.

• Role-Based Security: Authentication tokens, role authorisation, and row-level security policies ensure that citizens, officers, and administrators access only the data relevant to them.

• Proactive IoT Monitoring: Threshold-based alerts from smart bins, water-level sensors, air quality sensors, and streetlights allow problems to be detected before they are reported by citizens.

• Scalability: The modular layered architecture and cloud-hosted database facilitate seamless addition of new departments, issue categories, device types, or advanced classification models as the deployment expands.

• Visual Analytics: Integrated dashboards, charts, and incident maps enable intuitive interpretation of complaint trends and departmental performance for administrators and officers.

VIII. FUTURE SCOPE AND APPLICATIONS

A. Future Scope

The present Civic Eye system offers multiple compelling directions for future enhancement. Replacing the rule-based classification service with a trained natural language processing model and an image-based convolutional network would allow the system to learn from real complaint data, report a measured accuracy, and classify complaints directly from uploaded photographs. Persisting IoT readings from physical sensor hardware, rather than a simulated device registry, and storing notifications in the database would transition the platform from a demonstration setup to a continuously operating monitoring service.

Predictive analytics based on historical complaint records could forecast complaint hotspots and resolution times, complementing the rule-based estimates currently in use. A native mobile application with push notifications, multilingual support for regional languages, and voice-based complaint submission would substantially extend the system to a general, non-specialist audience. Long-term integration with smart city infrastructure, municipal ERP systems, and open government data platforms could support automated governance mechanisms, enabling dynamic dispatch of field teams in response to detected problems.

B. Applications

The Civic Eye platform addresses a wide array of real-world application domains. Municipal corporations and urban local bodies can deploy it as a unified grievance redressal system, replacing manual registers and helplines with a trackable digital workflow. Individual departments, including road, sanitation, water, drainage, electrical, and environment, can prioritise field work according to the AI-assigned priority and monitor their own performance through dashboards. Citizens benefit from a convenient channel to report issues, receive updates, and verify that their problems have actually been solved.

Smart city command centres can integrate the platform with sensor networks to monitor waste levels, flood risk, and air quality in real time and to dispatch response teams proactively. Disaster management authorities can use the flood and water-level alerts as an early warning input during monsoon seasons. Academic and policy research communities can employ the platform as an analytical tool for investigating complaint patterns, response times, and the impact of governance interventions on civic service quality.

IX. CONCLUSION

This paper has presented Civic Eye, a comprehensive AI and IoT-enabled platform for smart civic complaint management. The system captures geo-tagged complaints from citizens, classifies and prioritises them through an AI engine, routes them to one of six municipal departments, and manages them through a seven-stage workflow involving citizens, officers, and administrators. A rigorous validation pipeline, encompassing mandatory field checks, category normalisation, and coordinate handling, together with token-based authentication and row-level database security, ensures data quality and secure multi-role access.

Functional evaluation of the implemented prototype demonstrates that representative complaints in all eight supported categories are consistently classified, prioritised, and assigned resolution estimates, and that IoT readings are correctly converted into Normal, Warning, and Critical states. The citizen-verified closure mechanism and the complete status history provide transparency and accountability that conventional manual channels cannot offer, while the modular architecture allows individual components to be improved independently.

The findings support the core hypothesis that automated triage combined with verified resolution and IoT monitoring can make civic complaint handling more efficient and accountable. Future research will focus on replacing the rule-based classifier with trained text and image models evaluated on real complaint data, connecting physical sensor hardware, and deploying the system as a mobile-accessible platform to maximise its impact on urban communities.

X. ACKNOWLEDGMENT

The authors express their sincere appreciation to Prof. Imran Khan for his expert guidance, constructive mentorship, and continuous encouragement throughout the course of this research. Gratitude is extended to the Department of Computer Science and Engineering, Jhulelal Institute of Technology,

Nagpur, for providing the computational infrastructure and academic environment essential to the completion of this work. The authors also acknowledge the open-source communities behind React, Node.js, Leaflet, and OpenStreetMap, together with the Supabase platform, whose tools formed the technical foundation of this project. The cooperative efforts and dedication of all team members are recognised as fundamental to the quality and integrity of this research.

XI. REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22–32, 2014.

[2] M. Batty et al., "Smart cities of the future," The European Physical Journal Special Topics, vol. 214, no. 1, pp. 481–518, 2012.

[3] R. Kitchin, "The real-time city? Big data and smart urbanism," GeoJournal, vol. 79, no. 1, pp. 1–14, 2014.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[5] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[6] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The Internet of Things is the backbone," IEEE Consumer Electronics Magazine, vol. 5, no. 3, pp. 60–70, 2016.

[7] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," Advanced Engineering Informatics, vol. 25, no. 3, pp. 507–515, 2011.

[8] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Univ. California, Irvine, CA, USA, 2000.

[9] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," IETF RFC 7519, May 2015.

[10] Supabase Inc., "Row Level Security," Supabase Documentation. [Online]. Available: https://supabase.com/docs/guides/database/postgres/row-level-security