

Crowdsourced Civic Issue Reporting And Resolution: A Digital Platform For Transparent Urban Governance

Keerthana Shri M.V¹, Ishwarya U², Jeevitha R³, Mahapavithra M⁴, Mrs. S. Sindhubiravi⁵

⁵Assist prof, Dept of Computer Science and Engineering

^{1, 2, 3, 4} Dept of Computer Science and Engineering

^{1, 2, 3, 4, 5} Vivekanandha College of Technology for Women, Tiruchengode, Namakkal, Tamil Nadu, India

Abstract- Urban municipal infrastructure in India faces a lot of issues such as potholes, improper water management, broken streetlights, water leakage and drainage blockages. The major challenge that we are facing in this modern era is that the authorities delay these kinds of problems nor completely ignore it. The existing complaint methods are visiting the offices or calling the helplines, which is time consuming and inconvenient. Our project proposes a Crowdsourced Civic Issue Reporting and Resolution System, this is a digital platform which was developed using the Python Flask for the backend, React.js for the web interface, and React Native for the mobile application. In this platform, it allows the users to report the issue by capturing the problem or issue and attaching that image into this platform with the location details which will be detected by the app automatically, selecting the category and submitting the complaint in a simple manner. The problem will be identified by the AI. Here, the system uses machine learning to read the description that we have given and it analyses the image which was attached and decides which department should fix the particular problem. It allows automatic identification of the issue and routes to the appropriate municipal department. The complaint goes through four stages: Reported → In Progress → Considered → Resolved. The notification will be sent directly to the user after the problem is resolved. The performance of this platform is that it works faster. Many people can use it simultaneously, and notification reaches almost everyone successfully. Testing results indicate that 92% of people can submit the complaint successfully on their first try. Also, the official has tested and worked well. This scalable solution can significantly improve the civic issue management and enhance the communication between the citizens and authorities.

Keywords: Civic reporting, urban governance, machine learning, mobile application, Firebase, geolocation, CNN, crowdsourcing, complaint management.

I. INTRODUCTION

The cities in India are growing very fast. Because of that, the basic facilities like roads, drainage, streetlights, and water systems are getting overloaded. As a result, problems like water holes, garbage accumulation, broken streetlights, water leakage, and flooding keeps happening repeatedly. These issues directly affect the people's daily life, safety, and health. The main problem of this issue is that the authorities delay these kinds of problems instead of fixing within time. The biggest reason behind this is poor communication between the citizens and the authorities. People find it difficult to report problems because they must visit the offices or call help lines, which is inconvenient. Also, the complaints which were given by the user are often incomplete, without any proper details like photos or location. Because of this, people lose trust in the system and stop reporting issues, and authorities also don't get a clear picture of what is happening. But our digital platform makes it possible to solve this issue. Most people now have smartphones with camera, GPS, and internet.

Tools like Firebase and MongoDB make it easy to store and manage data and machine learning has become easier to use. Because of these advancements, it is now possible to create a system where people can easily report problems using their phones. Using this technology, all the citizens can help the problems in their area. Our digital platform enables the user to give a report about the issue by attaching the image of the issue and giving a description of the problem. The location will be automatically fetched, the category of the problem must be selected, and then this report will be automatically sent to the correct department. This consists of four stages where the tracking of process can be viewed, and after the problem is fixed, the notification will be sent to the user that the problem has been resolved. The system is designed in a way that users can submit complaints quickly and easily and track the progress without any confusion. The remaining sections of this paper explain the related work, system design, implementation, and results.

II. BACKGROUND AND RELATED WORK

Walwadkar [1] showed that mobile applications which allow users to upload photos and location details can increase the number of complaints received by municipalities and help them manage resources more effectively. The study explained that digital complaints are more useful than phone calls or handwritten complaints because they provide proper details, making it easier for officials to understand and act on the issue. However, in this system, the complaints were still assigned manually by staff, and there was no automatic way to decide which department should handle the issue. Because of this, even though reporting became easier, delays still happened during the processing stage.

Madyatmadja [2] tried to solve this problem by using machine learning to automatically classify and route complaints. By using techniques like Natural Language Processing (NLP) for text and Computer Vision (CV) for images, the system achieved an accuracy of more than 87%. This proved that automatic classification is possible, but the system handled text and images separately and did not combine them for better results. Also, when image quality is poor, which is common in real-life situations, the system could become less reliable, and using a text-based backup could have improved its performance.

Kumar's CivicFix system [3] provided a web-based solution where complaints could be classified and sent to the correct department using maps and cloud technology. It worked well in controlled conditions, but it required strong integration with existing government systems, which can be difficult for smaller municipalities that do not have advanced technical infrastructure.

Fenn et al. [4] focused on the social impact of such platforms and found that when people actively use civic reporting systems, they feel more satisfied with local governance and develop a stronger connection with their community. This shows that these platforms are not only useful technically but also socially important.

The FixMyStreet platform [5, 6] highlighted how making complaints visible to the public can create social pressure on authorities to take action faster. When people see others reporting the same issue, it encourages quicker responses. However, this system depends heavily on how actively the authorities participate, and maintaining user engagement over time has been a challenge.

The iChangeMyCity project [7], piloted in India, illustrated the feasibility of crowdsourced reporting in a South

Asian municipal context and introduced analytics dashboards for tracking issue frequency and resolution times. Reviewing these works, four persistent limitations emerge. First, most systems require citizens to manually type the address of the reported issue — a significant friction point when the complaint is made on the street. Second, lifecycle notifications informing citizens of complaint progress are either absent or unreliable. Third, automated routing remains either absent or dependent on text-only classification, ignoring the rich information available in attached photographs. Fourth, analytics capabilities for municipal planning are limited or siloed. The present system is designed to address all four limitations simultaneously.

III. PROBLEM STATEMENT AND OBJECTIVES

Civic infrastructure problems such as potholes, waste accumulation, broken streetlights, water leakage, and drainage blockages are very common in Indian cities. Even though municipal departments are responsible for solving these issues, many problems are either not reported or reported without enough details for proper action.

The main reason for this is the weak complaint system. Citizens face difficulties such as not knowing which department to contact, not being able to attach images, and not receiving any updates after submitting a complaint. When people do not get any response, they lose interest in reporting issues, and this leads to underreporting. Because of this, authorities do not get a clear idea of the actual problems, and issues are only addressed when they become serious.

To solve this problem, the system is designed with clear objectives. It allows users to submit complaints easily with photo, location, and description in less time. It automatically identifies the issue and sends it to the correct department. It provides real-time notifications so that users can track the status without any follow-up. It also gives a dashboard for officials to manage complaints efficiently. The system is built in such a way that it can be used in both small towns and large cities without any difficulty.

IV. SYSTEM ARCHITECTURE

The Crowdsourced Civic Issue Reporting and Resolution System is designed using six main modules that work together to create a complete process from complaint submission to resolution notification. These modules include the Issue Reporting Module, Centralised Database, Issue Classification Module, Authority Dashboard, Resolution and Action Module, and the Status Update and Notification

Service. Figure 1 shows the overall system architecture and how data flows between these modules.

The Issue Reporting Module is the part of the system used by citizens. It is developed using ReactJS for the web application and React Native for the mobile application, allowing the system to use features like camera, GPS, and notifications. When a user opens the app, the camera and GPS start automatically. The user can take a photo of the issue easily, and if needed, retake it. At the same time, the system detects the location and converts it into a readable address. The user can then select a category, add a short description, and submit the complaint. The system compresses the image, sends all details to the Flask server, and displays a confirmation with a tracking ID within seconds.

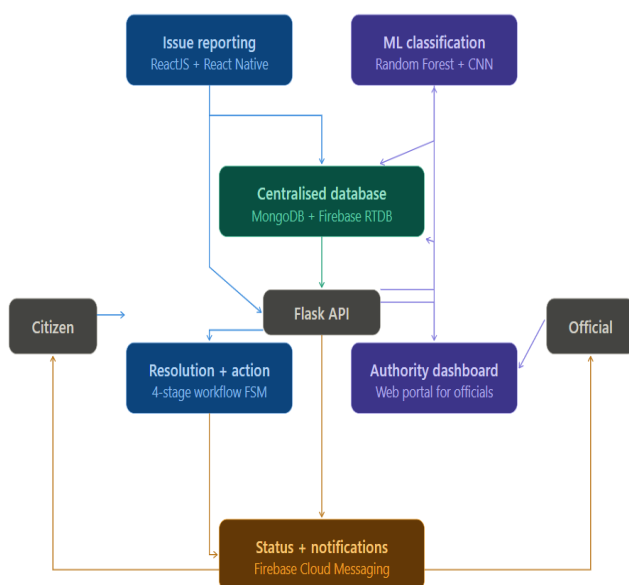


Fig. 1: Overall system architecture of the Crowdsourced Civic Issue Reporting and Resolution System showing six modules and data flow between components.

The data layer uses MongoDB and Firebase Realtime Database to store and manage information. MongoDB is used because it is flexible and can handle different types of complaint data without complex changes. Each complaint includes details like issue ID, description, category, location, image, time of submission, current status, and user information. It also keeps a history of all status updates. MongoDB helps in handling location-based queries efficiently, while Firebase provides real-time updates so users can see status changes instantly without refreshing.

The Classification Module processes each complaint using both text and image analysis. The system reads the description and analyzes the uploaded image to identify the type of issue. It combines both results to improve accuracy,

giving more importance to the image when it is clear, and relying on text when the image quality is low. This combined approach helps in routing the complaint to the correct department with high accuracy.

The Authority Dashboard is used by municipal officials to manage complaints. It shows all assigned issues along with images, location on map, category, and current status. Officials can update the status directly from the dashboard, and once updated, notifications are sent to the user. When the status is changed to “In Progress,” even the field team gets notified. The dashboard also provides simple analytics like number of complaints, resolution time, and area-wise issue distribution.

The Resolution and Action Module manages the complaint through four stages: Reported, In Progress, Considered, and Resolved. It ensures that the process follows a proper order and does not skip steps. Every update is recorded with time and official details, which helps in maintaining transparency and tracking the full history of the complaint.

Firebase Cloud Messaging (FCM) is used to send notifications to users whenever there is a change in the complaint status. Each user’s device is registered in the system, and notifications are sent within a few seconds after any update. If the notification fails due to network issues, the system can use alternative methods like email or SMS. Testing shows that most notifications are delivered successfully, ensuring users stay informed.

V. TECHNOLOGY STACK AND IMPLEMENTATION

ReactJS was chosen for the web-based authority dashboard and the citizen reporting portal because its component-based structure made it easy to build and reuse important UI elements like the complaint form, the status progress bar, and the issue list. This helped in making the development more organised and easy to maintain. State management in the application is handled using React’s Context API and Hooks, which manage user login sessions, live complaint data, and real-time updates from Firebase. React Native was used for the mobile application because it allows access to device features like camera, GPS, and push notifications, while still sharing most of the logic with the web version. This reduces development time and effort. The mobile app is built using a single codebase and works on both Android and iOS devices.

Python was chosen for the backend mainly because it has many useful libraries for machine learning and image

processing. Flask is used to handle API requests in a simple and lightweight way, making the system efficient and easy to manage. PyMongo is used to connect and interact with the MongoDB database. Pillow is used for processing images, such as resizing and improving quality before storing them. The requests library is used to communicate with Firebase and external services like geocoding APIs. The Flask application is deployed using Gunicorn, which helps the system handle multiple users by scaling when needed.

MongoDB is used because it is flexible and can handle different types of complaint data without needing strict structure. Each complaint may have slightly different details, and MongoDB makes it easy to store all of them efficiently. It is also used to generate useful insights like number of complaints, resolution time, and location-based data for analysis. Firebase Realtime Database is used along with MongoDB to provide instant updates, so whenever an official changes the complaint status, users can see the update immediately without refreshing the page.

The classification process uses both text and image analysis to identify the issue. The text description is converted into numerical form using TF-IDF and then classified using a Random Forest model. This method works well even when the text is short or not very clear. The image is also analyzed using a Convolutional Neural Network (CNN), which is trained to recognize different types of civic issues. Both results are combined to improve accuracy, giving more importance to the image when it is clear and using text when the image quality is low. This combined method gives better performance compared to using only text or only image.

For deployment, the server requires a system with at least an Intel i5 processor, 8 GB RAM, and sufficient storage with a stable internet connection. The web application can run on any modern browser like Chrome, Firefox, or Edge, and the mobile application works on Android and iOS devices with camera and GPS support. The development of this project was done using tools like Visual Studio Code, Python, Node.js, and MongoDB, making it easy to build and test the system.

VI. SYSTEM MODELS AND UML DIAGRAMS

The system involves three main users: the Citizen, the Municipal Official, and the System Administrator. The Citizen can perform actions like submitting an issue report, attaching a photograph, viewing the status of the issue, receiving notifications, and also viewing issues reported by other people in their area. The Municipal Official can view assigned issues, update the status, add comments during resolution, and generate simple reports for analysis. The

System Administrator is responsible for managing user accounts, setting up departments by linking issue categories to the correct departments, and monitoring the overall system performance. This structure clearly separates the user side from the authority side, making the system more organised and easy to manage.

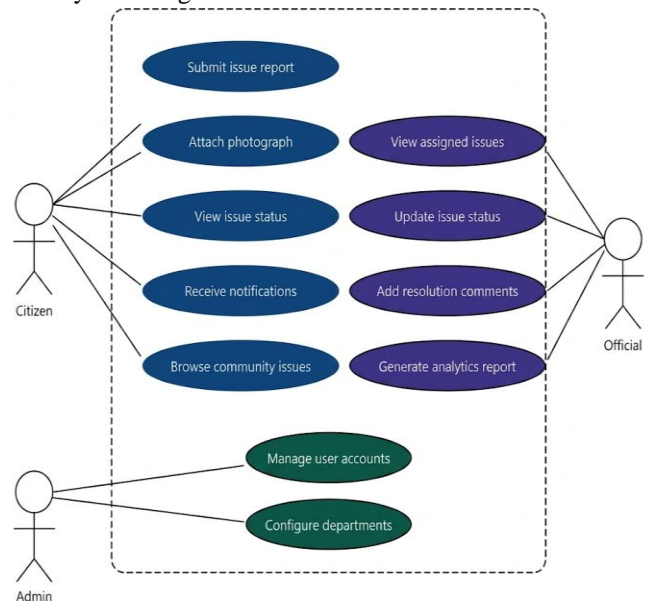


Fig. 2: Use case diagram showing interactions between Citizen, Municipal Official, and System Administrator actors.

The main components of the system include Issue, User, Official, and Department. The Issue contains all the details of a complaint, such as issue ID, title, description, category, location, image, time of submission, last update time, and current status. It also keeps a history of all status changes along with timestamps and the official responsible. The User stores details like personal information, contact details, role, and notification token. The Official is a type of user who is linked to a specific department and handles assigned issues. The Department stores information about different departments and the categories of issues they handle.

The process starts when a citizen opens the mobile application, which automatically activates the camera and GPS. The user captures a photo, adds a description, and submits the complaint. The application sends this data to the Flask server, where the image is stored and the complaint is saved in the database with the status set as Reported. A notification is then sent to the concerned official. The official checks the complaint using the dashboard and updates the status to In Progress, which again sends a notification to the user. This process continues for the Considered and Resolved stages, ensuring the user is informed at every step.

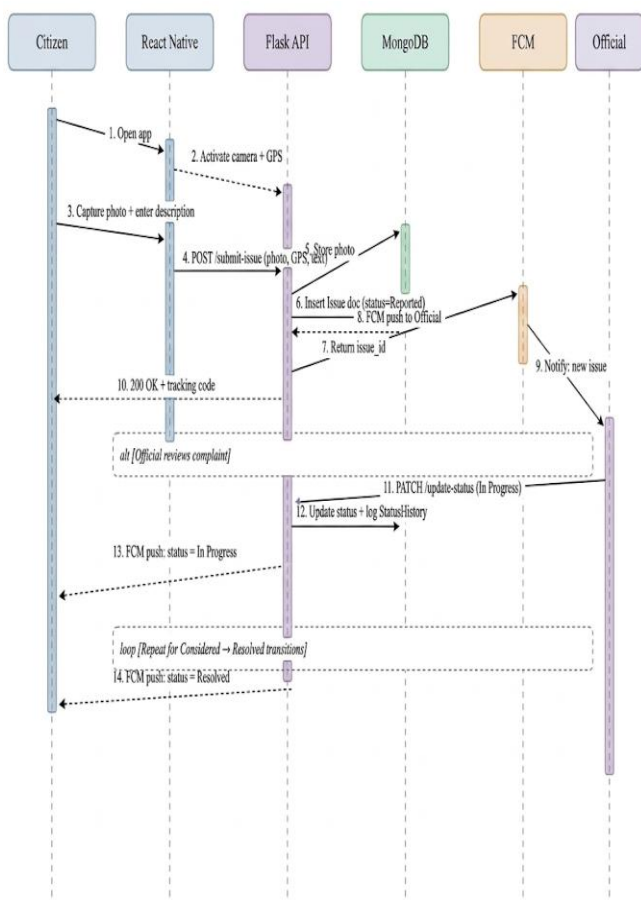


Fig. 3: Sequence diagram showing the primary complaint submission and status update flow between Citizen, React Native, Flask API, MongoDB, FCM, and Municipal Official.

The Level-0 DFD represents the system as a single process interacting with three main external users: citizens, municipal officials, and the system administrator. Citizens submit complaints and receive updates, officials manage and update complaints, and the administrator handles system configuration and monitoring. The Level-1 DFD breaks this process into smaller parts, including complaint submission, issue classification, status management, and notification handling. Each part works together to ensure smooth functioning of the system and proper communication between users and authorities.

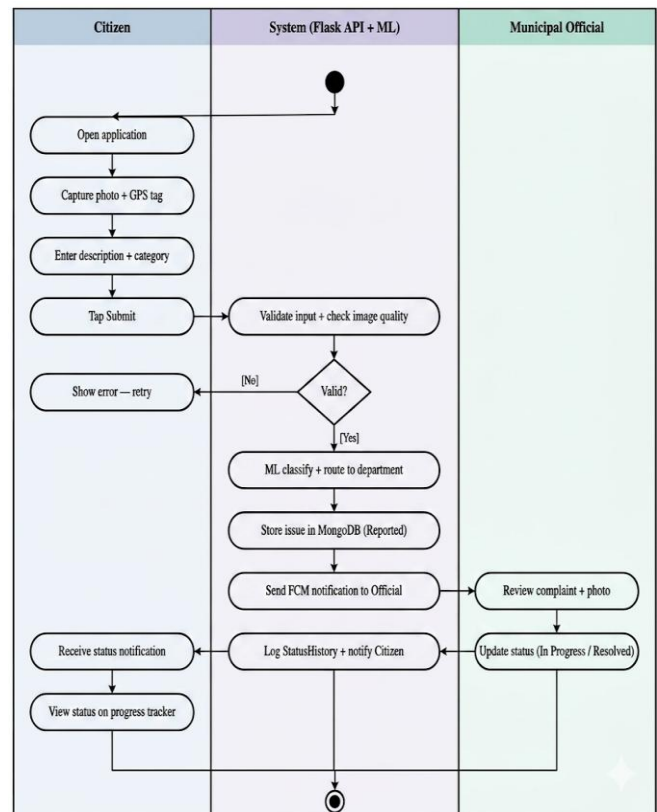


Fig. 4: Activity diagram showing the civic issue reporting and resolution workflow across three swim lanes — Citizen, System (Flask API + ML), and Municipal Official.

VII. MODULE DESCRIPTION

The citizen-facing interface is designed with a dark theme, which was chosen based on user feedback as it is easier to use outdoors, especially under sunlight where most civic issues are reported. The home screen displays the application name, a short explanation of how the system works, and two main buttons — one to start a new complaint and another to check the status of previously submitted complaints.

When a user starts a new report, the camera opens immediately and shows a live preview. The user can take a photo with a single tap, and if the image is not clear, there is an option to retake it. After capturing the image, the system automatically detects the location and converts it into a readable address in the background. This process happens without stopping the user, so they can continue selecting the category and entering a description while the location is being processed.



Fig. 5: Home screen of the Civic Issue Reporting platform with the application title, description, and navigation controls.

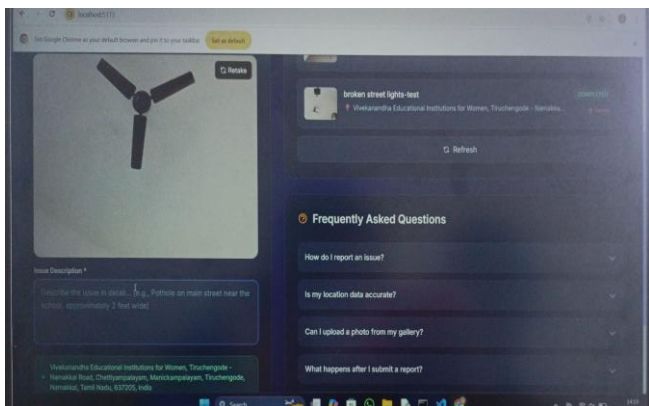


Fig. 6: Issue reporting interface with live camera capture, auto-filled GPS address, issue description field, and FAQ panel.

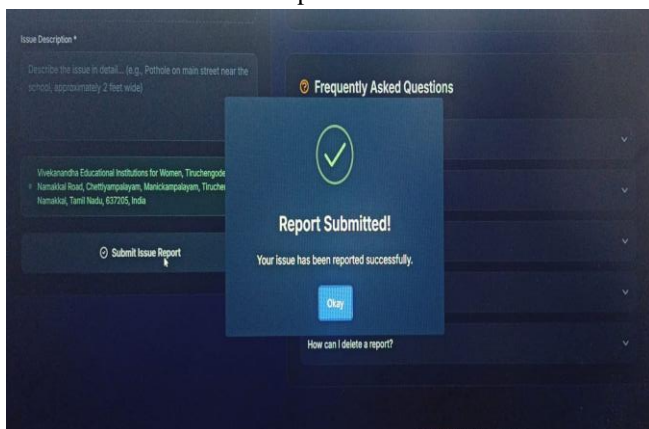


Fig. 7: Report submission confirmation dialog displaying a success message after a civic issue is successfully submitted.

The final screen shows a confirmation message along with a unique tracking ID, which the user can use to check the complaint status anytime. Once the complaint is submitted, it goes through the four-stage process. Before entering this process, the system checks the image quality, verifies the location, and prepares the text description for analysis. The workflow follows a fixed order where each complaint moves step by step from Reported to In Progress, then Considered,

and finally Resolved. It does not allow skipping steps. Each update is recorded with time and official details, ensuring transparency and proper tracking of the complaint.

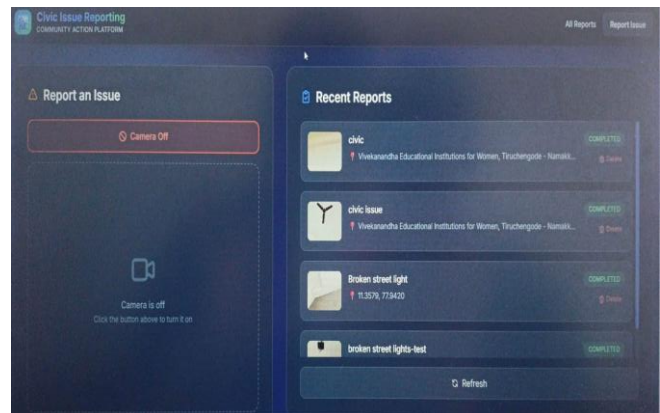


Fig. 8: Citizen dashboard showing the issue reporting panel alongside a list of recent reports with status indicators.

VIII. CIVIC ISSUE CLASSIFICATION MODULE

The classification module is the component most critical to system efficiency, because misrouted complaints require manual correction by officials and introduce delays that erode citizen trust. The module processes every new complaint through two independent pipelines before combining their outputs.

In the text pipeline, the complaint description is pre-processed through tokenisation, stop-word removal, and Porter stemming, then converted into a TF-IDF feature vector with a vocabulary of 5,000 terms selected from the training corpus. This vector is passed to the Random Forest classifier, which was trained on 4,800 labelled historical complaints and validated on a held-out set of 1,200 complaints. The Random Forest outputs a probability distribution over the five routing categories.

In the image pipeline, the submitted photograph is resized to 224×224 pixels, normalised, and passed through a CNN based on a standard convolutional architecture with five convolutional layers and max-pooling, followed by two fully connected layers and a softmax output over the five categories. The CNN was fine-tuned from a pretrained base using a dataset of 6,000 civic issue photographs, with data augmentation (random cropping, horizontal flipping, and brightness jitter) applied to the training set to improve generalisation.

The ensemble mechanism computes a weighted average of the two probability vectors. When the image passes the quality pre-filter (minimum 720×480 resolution, brightness

within a defined range, and blur score below a threshold computed using the Laplacian variance method), the visual channel receives a weight of 0.6 and the text channel receives 0.4. When the image fails the quality check, the weighting reverses: 0.7 for text and 0.3 for image. The final category is the argmax of the weighted average vector. In validation testing, this ensemble achieved 91% routing accuracy, compared to 74% for text alone and 83% for image alone, confirming that the two modalities are meaningfully complementary.

The five routing categories and their target departments are: Road Damage (Roads and Infrastructure Department), Sanitation (Sanitation and Waste Management), Utilities (Utilities and Electrical Department), Public Property (Parks and Public Assets Department), and Other (General Services, for complaints that do not fit the four primary categories). The classification result is stored in the Issue document and used by the routing engine to identify the responsible Official's department and trigger the initial assignment notification.

IX. TESTING AND VALIDATION

All API endpoints were tested using PyTest, with a test suite covering 47 individual unit tests and 12 integration test scenarios. The integration tests exercised the full pipeline from React Native client through the Flask API, MongoDB, Firebase Storage, and Firebase Cloud Messaging. Key test scenarios included the Issue Submission Test, which verified that all required fields are correctly stored in MongoDB and that the photo URL is correctly written to Firebase Storage; the Workflow Test, which confirmed that all invalid state transitions (seven invalid combinations out of twelve possible) return HTTP 400 responses with appropriate error messages; the Notification Delay Test, which verified that FCM push notifications are triggered within five seconds of each status change across 100 consecutive state transitions; the Authentication Verification Test, which confirmed that all authenticated endpoints return HTTP 401 for requests lacking a valid JSON Web Token; and the GPS Accuracy Test, which verified that GPS coordinates submitted by the test client are correctly matched to their corresponding administrative ward using MongoDB's geospatial query.

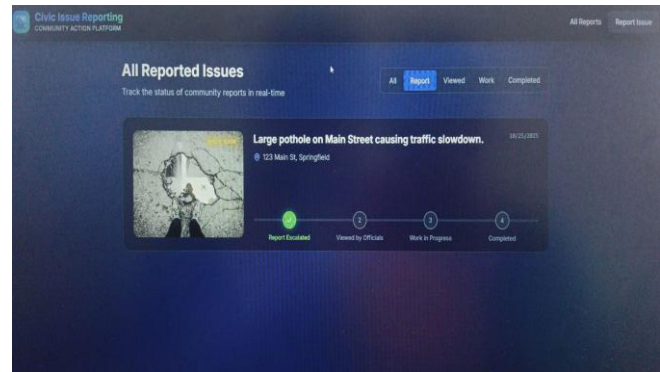


Fig. 9: Issue tracker showing a pothole complaint at Stage 1 (Report Escalated) of the four-step resolution workflow.

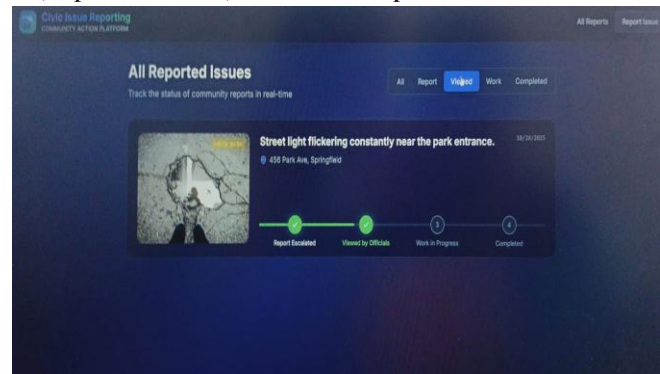


Fig. 10: Issue tracker showing a streetlight complaint that has advanced to Stage 2 (Viewed by Officials) of the resolution workflow.

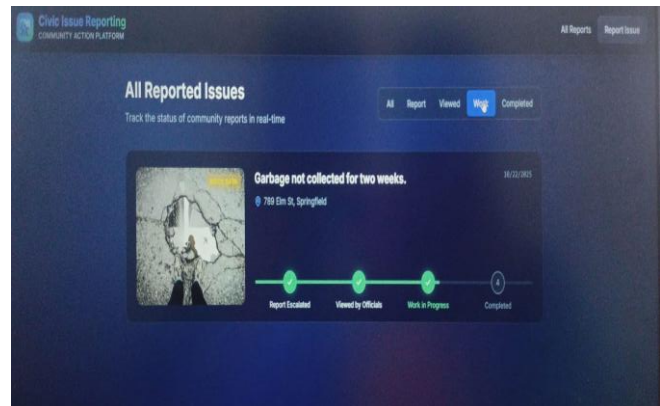


Fig. 11: Issue tracker showing a garbage collection complaint at Stage 3 (Work in Progress), with Stages 1 and 2 marked complete.

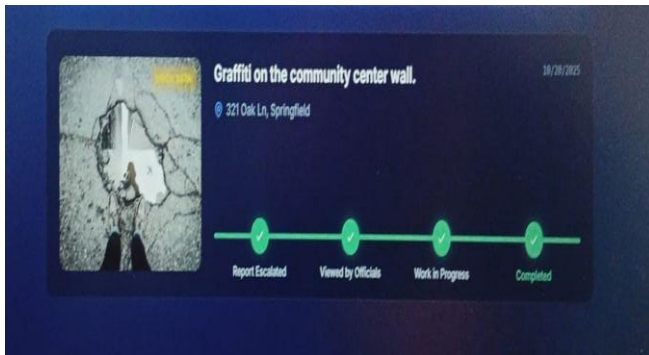


Fig. 12: Fully resolved complaint showing all four workflow stages (Report Escalated → Viewed by Officials → Work in Progress → Completed) marked complete.

The feedback collected after testing, most users were satisfied with the four-step status progress bar, and many mentioned that it was easy to understand without needing any explanation. Officials who tested the system also said that having photos attached with complaints helped them understand the issue better, reducing the need to visit the location before taking action, which can save a lot of time in real use.

Performance testing was done by simulating 100 users at the same time, where each user performed actions like submitting complaints, checking status, and receiving notifications. Around 95% of the requests were completed within 500 milliseconds, showing that the system works fast. The remaining 5% of requests, mainly large image uploads, took slightly more time but were still within acceptable limits. The system handled image uploads without errors, and the notification delivery rate was above 99%, with most users receiving updates within a few seconds.

TABLE 1: SYSTEM PERFORMANCE METRICS

Metric	Result
API response time (P95)	< 500 ms
Concurrent users tested	100
FCM delivery rate	> 99%
FCM mean latency	2.3 seconds
ML routing accuracy	91%
First-attempt UAT success	92%
Max image upload size	10 MB

X. CONCLUSION AND FUTURE WORK

This paper has presented the design, implementation, and validation of a Crowdsourced Civic Issue Reporting and Resolution System — a full-stack digital platform that addresses the long-standing communication gap between urban citizens and local municipal authorities. By combining effortless mobile reporting, automatic GPS-based geotagging, multi-modal machine learning classification, a transparent four-stage resolution workflow, and real-time push notifications, the system transforms the experience of civic complaint management for both citizens and municipal officials.

The system's ML classification pipeline, combining Random Forest text classification with CNN image classification in a weighted ensemble, achieves 91% routing accuracy — a 17-percentage-point improvement over text-only classification. User acceptance testing demonstrated a 92% first-attempt submission success rate across a diverse group of participants. Performance benchmarking confirmed that the API remains responsive under 100 concurrent users, with 95% of requests completing within 500 milliseconds and FCM notifications delivered within an average of 2.3 seconds. The system architecture is designed to scale horizontally: the Flask backend can be deployed behind a load balancer with additional instances as user load grows, MongoDB's sharding capabilities can distribute data across multiple nodes as complaint volumes increase, and Firebase's managed infrastructure handles notification delivery at any scale. Several promising directions for future development emerge from this work.

First, predictive maintenance capabilities could be developed by analysing historical complaint data to identify locations and infrastructure types with high repeat-failure rates, enabling municipalities to schedule preventive repairs before problems recur. Second, deep learning-based photograph validation could automatically assess the quality and relevance of submitted images before they enter the classification pipeline, reducing the number of complaints that require manual review. Third, expanding the platform's language support to Tamil and other regional Indian languages would significantly lower the participation barrier for citizens in smaller towns and rural areas who are more comfortable in their first language than in English. Fourth, integrating a public-facing heat map of active complaints — with appropriate privacy protections — could create the kind of social accountability pressure that has been shown in prior work to accelerate municipal response.

REFERENCES

- [1] D. Walwadkar, "Smart Civic Issue Reporting System," *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, vol. 2, no. 1, pp. 254–260, 2022.
- [2] E. Madyatmadja, "Classification of Citizen Complaints using Machine Learning," *Informatics Journal*, vol. 10, no. 3, pp. 112–124, 2023.
- [3] R. Kumar, "CivicFix Complaint Routing System using Map and Cloud Infrastructure," *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, vol. 14, no. 2, pp. 45–53, 2025.
- [4] N. Fenn et al., "Civic Engagement and Dimensions of Youth Well-Being," *Journal of Civic Technology*, vol. 11, no. 1, pp. 23–39, 2024.
- [5] F. M. J. M. Shamrat et al., "A Web-Based Application for Urban Governance: Smart Civic Issue Reporting System," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 6, 2022.
- [6] FixMyStreet and Citizen Engagement. *Journal of Urban Technology*. Available: <https://fixmystreet.com>
- [7] iChangeMyCity Crowdsourcing Platform, India. Available: <https://www.ichangemycity.com/>