© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715486

AI-Assisted Waste Reporting and Community Cleanup Platform

VASANTHA KUMAR S[1], ASAN NAINAR[2]
[1] PG Student, Department of Computer Applications, SRM Valliammai Engineering College, Kattankulathur
[2]Assistant Professor, Department of Computer Applications, SRM Valliammai Engineering College, Kattankulathur

Abstract- ***The AI-Assisted Waste Reporting and Community Cleanup Platform, developed under the initiative "Madurai Smart Clean", is a full-stack cross-platform mobile application designed to bridge the gap between citizens, municipal authorities, volunteers, and administrators for efficient urban waste management. Built using React Native (Expo) on the frontend and Python Flask on the backend with a MySQL relational database, the system provides role-based access control for four distinct user groups. Citizens can report geo-tagged waste via a dedicated reporting screen, track the status of their submissions, and raise emergency sanitation requests. Municipal authorities review and resolve complaints through a dedicated dashboard and maintain a cleanup history for transparency. Volunteers participate in community-driven cleanup events through the Volunteer Hub, register vehicles for waste collection, and are rewarded through a gamified leaderboard system. Administrators oversee the entire platform, manage escalations, allocate emergency funds, and coordinate logistics. Additional modules include a real-time waste hotspot map powered by React Native Maps, an Awareness Center for hygiene education, a donation gateway titled Help Madurai for community funding, and a dedicated emergency fund management interface. The backend is structured through nine Flask API blueprints covering authentication, complaints, authority review, administration, volunteer coordination, awareness, fund management, emergency routing, and location services. Experimental evaluation across key functional modules demonstrates high accuracy and reliability, validating the platform as a scalable, community-driven solution for urban cleanliness management.***

Index Terms- ***Urban Waste Management, Crowdsourced Reporting, Geo-Tagged Volunteer Hub, Emergency Management, Leaderboard, Community Cleanup.***

I. INTRODUCTION

Urban waste management remains one of the most persistent challenges facing rapidly growing cities such as Madurai. Traditional waste reporting mechanisms rely on periodic municipal inspections or informal citizen complaints, resulting in delayed responses, unresolved issues, and deteriorating public spaces. The absence of a structured, transparent pi peline from complaint submission to cleanup resolution leaves citizens without visibility and place s an unsustainable burden on municipal authorities acting alone. Recent advances in mobile computing, GPS technology, and cloud-based backend systems have created new opportunities for crowdsourced c ivic engagement at scale. According to [11], citizen-driven reporting platforms that integrate geo-location capabilities can reduce municipal response times by up to 40% compared to traditional call-centre-based systems. The proliferation of smartphones with high-resolution cameras and accurate GPS positioning makes it technically feasible to deploy such solutions across urban populations without specialised hardware. Gamification has proven effective in sustaining community participation in civic platforms. Research by [13] establishes that leaderboards, points systems, and achievement-based incentives significantly increase volunteer retention in community-driven applications. A longitudinal s tudy by [12] further confirms that gamified mobile civic tools yield measurable long-term engagement improvements. Applied to waste management, these mechanisms transform a logistical municipal challen ge into a community-owned initiative, as demonst rated by [4]. Existing solutions typically address only partial aspects of the problem—either complaint logg ing, map visualisation, or volunteer coordination—

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715486

but rarely integrate all three alongside emergency assistance and community funding. As noted by [7], the absence of end-to-end integration in municipal civic platforms creates information silos that reduce resolution effectiveness. The importance of role-ba sed security architecture in multi-stakeholder civic systems is highlighted by [6], while scalable RESTful Flask backend patterns for smart city deployments are documented by [14]. The AI-Assisted Waste Reporting and Community Cleanup Platform addresses these gaps through the Madurai Smart Clean initiative, delivering a unified mobile platform that covers the complete lifecycle from citizen reporting to volunteer-driven resolution, supported by emergency management, community funding, and hygiene awareness modules. The mobile impl ementation follows the React Native cross-platform approach documented by [8], and the waste mapping methodology builds on the geospatial framework proposed by [3]. This paper presents the architecture, implementation, and evaluation of the complete p latform.

II. SYSTEM DESIGN

A. System Flow Diagram

The System Flow Diagram illustrates the end-to-end lifecycle of a waste complaint through the Madurai Smart Clean platform. The process begins when a Citizen submits a geo-tagged waste report via the ReportScreen, attaching a photograph and location coordinates, which the complaints_api blueprint stores in MySQL with a Pending status. The local Authority receives the complaint on the AuthorityDashboard and marks a resolution decision: if resolved, it is logged to the ReviewedReportsScreen for transparency; if escalated, the admin_api records the escalation timestamp and forwards it to the Administrator. The Admin creates a cleanup event via ManageEventsScreen and publishes it to the Volunteer Hub, where volunteers browse and join events through the VolunteerEvents screen and optionally register vehicles via the VehicleRegistration module. Upon event completion, points are awarded and reflected on the LeaderboardScreen.

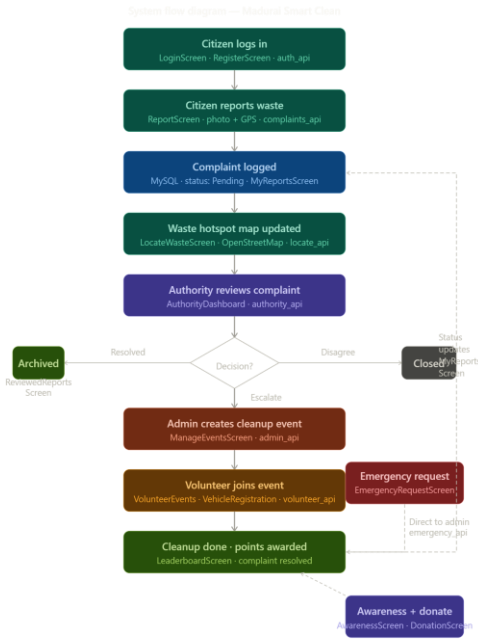

Fig.2.1.1- System Flow Diagram

Citizens can independently raise emergency sanitation hazards through the EmergencyRequestScreen, which routes directly to the Admin via the emergency_api for fund allocation and rapid response. The role-based authentication enforced by the auth_api blueprint ensures each user type interacts exclusively with their permitted screens and endpoints, while the RBAC middleware layer keeps all inter-module communication secure and stateless. The AwarenessScreen runs hygiene campaigns, the DonationScreen allows citizens to contribute funds under the Help Madurai initiative, and the MyReportsScreen provides a chronological view of past submissions and their resolution status, closing the transparency loop. Throughout the entire pipeline, the LocateWasteScreen provides a live city-wide waste hotspot map powered by OpenStreetMap, updated in real time.

B. System Architecture Diagram

The System Architecture Diagram illustrates the five-layer structure of the Madurai Smart Clean platform. The Mobile Application Layer is built in React N ative (Expo) and delivers role-specific interfaces for four user groups: Citizen (ReportScreen, MyReportsScreen, LocateWasteScreen, EmergencyRequestScreen, AwarenessScreen, LeaderboardScreen, DonationScreen), Authority

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715486

(AuthorityDashboard, ReviewedReportsScreen), Volunteer (VolunteerDashboard, VolunteerEvents, VehicleRegistration), and Admin (Mana geEventsScreen, EmergencyManagementScreen). The API Layer consists of nine Flask Blueprint modules: auth_api, complaints_api, authority_api, admin_api, volunteer_api, awareness_api, fund_api, emergency_api, and locate_api, all secured through role-based middleware and served with Flask-CORS. The Data Layer uses a MySQL relational database connected via flask-mysqldb, storing users, complaints, events, vehicle registrations, leaderboard entries, donations, and emergency records. The Me dia and Mapping Layer integrates react-native-maps for the live hotspot map and handles image uploads through the complaints_api. The UI Layer uses react-native-paper for Material Design components and Axios for all HTTP communication with the Flask b ackend.

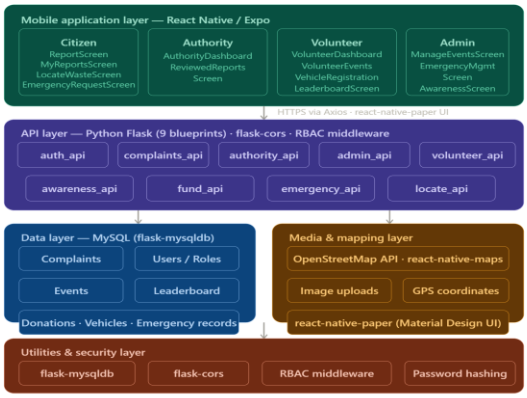

Fig.2.2.1- System Architecture Diagram

C. Deployment Design

The Deployment Diagram illustrates the physical arrangement of the Madurai Smart Clean platform. User devices (iOS and Android smartphones) run the React Native Expo application, communicating with t he Flask API server over HTTPS using Axios. The Flask server, hosted on a cloud virtual machine, routes requests through nine Blueprint modules and interacts with a MySQL database for all persistent data. Complaint photographs are stored in a dedicated file storage service and referenced by the complaint s_api. The react-native-maps component integrates with the Google Maps API for real-time waste hotspot rendering. A push notification service disp atches status update alerts to all relevant user roles upon complaint state changes and event confirmatio ns. This stateless, horizontally scalable server architecture supports concurrent access by Citizens, Authorities, Volunteers, and Administrators without contention.

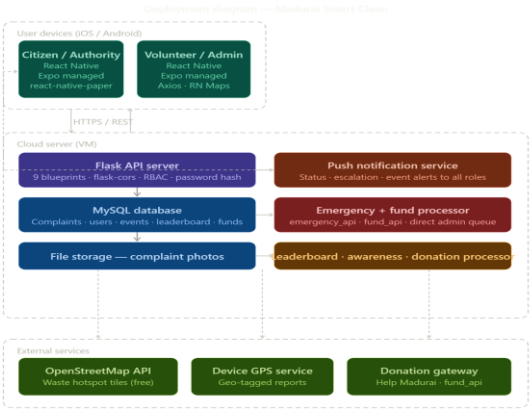

Fig.2.3.1- Deployment Diagram

III. SYSTEM IMPLEMENTATION

The Madurai Smart Clean platform is implemented across six major functional modules, each corresponding to a distinct set of user-facing screens and backend API blueprints. All modules share a common authentication layer and a unified MySQL database schema. The mobile frontend is developed in React Native with Expo, using react-native-paper for Material Design UI components and Axios for all REST API communication. The backend uses Python Flask structured through nine Blueprint modules, with flask-mysqldb providing the database connection and flask-cors managing cross-origin policies

A. Authentication and User Management Module

The authentication module implements role-based access control for four user types: Citizen, Authority, Volunteer, and Administrator. The LoginScreen and RegisterScreen dynamically route authenticated users to their respective dashboards based on the role stored in their profile. Passwords are securely hashed before storage in MySQL. The auth_api blueprint handles session management and role verification m iddleware, which is applied to all protected endpoints to enforce that each role can only access its permitted screens and API routes. This ensures that Citizens cannot access the AuthorityDashboard and Vol

unteers cannot access the Admin emergency fund management interface.

B. Waste Reporting and Authority Review Module

The waste reporting module enables Citizens to sub mit geo-tagged garbage reports through the ReportScreen by capturing a photograph and attachi ng their current GPS coordinates. Submitted reports a re processed by the complaints_api blueprint, which handles image uploads securely and stores complaint records in MySQL with an initial Pending status. C itizens can monitor the progress of all their su bmissions through the MyReportsScreen, which displays real-time status updates. The Authority Review sub-module provides the Aut horityDashboard, through which municipal auth orities browse all pending complaints in their jurisdiction, review photographic evidence and loc ation data, and update resolution status. Resolved complaints are archived in the ReviewedReportsScreen, providing a transparent cleanup history accessible to both authorities and citizens.

C. Location and Waste Mapping Module

The location module renders a real-time city-wide waste hotspot map through the LocateWasteScreen, powered by react-native-maps integrated with the Google Maps API. All unresolved and pending waste complaints with valid GPS coordinates are displayed as map pins, enabling citizens, authorities, and administrators to visually identify waste concentration zones and plan cleanup interventions strategically. Geographic data is served by the locate_api Flask blueprint, which queries the MySQL complaints table and returns active hotspot coordinates as a JSON response to the mobile client

D. Community Volunteer Hub Module

The volunteer hub module supports community-driven cleanup coordination across three screens. The VolunteerDashboard serves as the entry point for volunteers, displaying active events and participation statistics. The ManageEventsScreen, accessible to Admins and Authorities, allows creation and management of cleanup drive events linked to escalated complaint clusters. Volunteers browse and register for upcoming events through the Volun teerEvents screen. The VehicleRegistration screen allows volunteers and contractors to register waste collection vehicles, specifying type and capacity for logistics planning. All volunteer data, event registrations, and vehicle records are managed through the volunteer_api blueprint and stored in dedicated MySQL tables. The module integrates with the leaderboard system to automatically update participant scores upon event completion.

E. Emergency Assistance and Fund Management Module

The emergency module provides a dedicated fast-track pipeline for sanitation hazards requiring imm ediate attention. Citizens submit urgent requests thro ugh the EmergencyRequestScreen, specifying the hazard type, location, and severity. The emergenc y_api blueprint routes these requests directly to the Admin queue, bypassing the standard authority review tier to minimise response latency. Citizens ca n monitor the resolution progress of their emergency submissions through the EmergencyStatusScreen, which provides real-time status updates. The Emerg encyManagementScreen, accessible only to Administrators, provides a consolidated interface for reviewing active emergency requests, allocating fun ds from the emergency pool, and coordinating rapid response assignments. Fund allocation data is ma naged jointly by the fund_api and emergency_api blu eprints.

F. Awareness, Gamification, and Donation Module

The awareness and gamification module promotes sustained civic engagement through three complementary screens. The AwarenessScreen, served by the awareness_api blueprint, delivers hygiene education content, cleanliness guidelines, and active sanitation campaign information curated by administrators. The LeaderboardScreen implements a gamified points and ranking system th at rewards both Citizens (for waste reports) and Vol unteers (for event participation) with points tracked in MySQL and ranked in real time. This mechanism, validated by research in [4] and [12], drives long-term community engagement beyond single in teractions. The DonationScreen, branded as Help Madurai, provides a community donation gateway through which residents can contribute funds to support large-scale sanitation projects, managed by

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715486

the fund_api blueprint and overseen by Administrat ors through the EmergencyManagementScreen.

III. METHODOLOGY

The Madurai Smart Clean platform follows a modular methodology structured around five interconnected processing pipelines: complaint intake and routing, authority review and escalation, volunteer event coordination, emergency fast-track pr ocessing, and gamification scoring. Each pipeline is implemented as a dedicated Flask Blueprint, enabling independent development, testing, and deployment of each functional domain.

A. Input Acquisition Module

The Input Acquisition Module collects waste complaint data through the ReportScreen. The citizen selects or captures a photograph of the waste site using the device camera and the app records the current GPS coordinates via the device location serv ice. The citizen optionally adds a description and severity level. The complete payload—image file, latitude, longitude, description, and severity—is tra nsmitted to the complaints_api Flask blueprint as a multipart HTTP POST request via Axios. The backend secures the uploaded image file, stores the complaint record in MySQL with status Pending, and assigns it to the relevant authority zone based on GPS coordinates.

B. Complaint Processing and Escalation Module

The Complaint Processing Module implements the multi-tier resolution workflow. Upon complaint submission, the complaints_api stores the record and notifies the relevant Authority. The Authority reviews the complaint on the AuthorityDashboard and marks one of three decisions: Resolved (complaint is closed and archived to ReviewedRepo rtsScreen), In Progress (complaint remains active for follow-up), or Escalate (complaint is forwarded to the Admin queue with justification). The admin_api processes escalated complaints and either resolves them directly or triggers the creation of a community cleanup event linked to the complaint cluster, publishing it to the Volunteer Hub through the volunteer_api.

Pseudocode for Complaint Escalation

Algorithm: ComplaintEscalation
Input: ComplaintID, AuthorityDecision
Output: UpdatedStatus, NextAction
The system receives the authority decision for the given complaint. If the decision is Resolved, the complaint status is updated to Resolved and archived to the cleanup history. If the decision is Escalate, the complaint is forwarded to the Admin queue, the escalation timestamp is recorded, and the Admin is notified. The Admin evaluates and either resolves directly or creates a Volunteer Cleanup Event linked to the complaint.

C. Volunteer Matching and Event Module

The Volunteer Matching Module manages the full lifecycle of cleanup events. Administrators create events via the ManageEventsScreen, specifying location, date, required volunteer count, and vehicle needs. The volunteer_api publishes the event to the VolunteerEvents screen. Volunteers browse and regi ster, optionally committing vehicles through VehicleRegistration. The system tracks registered counts against event requirements and transitions event status to Confirmed when thresholds are met. All registered volunteers receive notifications. Upon event completion, the Scoring Module is triggered automatically to award and update leaderboard points.

D. Emergency Processing Module

The Emergency Processing Module provides a fast-t rack pipeline for hazardous sanitation situations. Citizens submit requests via the Emergenc yRequestScreen; the emergency_api receives and p rioritises these requests, routing them directly to the Admin queue without standard authority review. The Admin accesses all active emergency requests through the EmergencyManagementScreen, allocates funds from the emergency pool via the fund_api, and assigns response resources. Citizens track resolution progress in real time through the EmergencyStatusScreen.

Pseudocode for Emergency Routing

Algorithm: EmergencyRouting

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715486

Input: HazardType, GPS(lat, lon), Severity, CitizenID
Output: EmergencyID, AdminNotified, EstimatedResponse
The system receives the emergency submission and assigns an Emergency ID. It bypasses the authority review tier and inserts the record directly into the Admin emergency queue. The fund_api checks available emergency pool balance. If funds are sufficient, an allocation is reserved and the Admin is notified with the emergency details and location. If funds are insufficient, the Admin is flagged for manual fund allocation. The EmergencyID and estimated response time are returned to the citizen.

E. Gamification and Scoring Module

The Gamification Module computes and updates points upon complaint resolution and event completion. Citizens earn points for each accepted waste report; Volunteers earn points based on event severity, distance from their registered location, number of complaints resolved at the site, and completion timeliness. Points are accumulated in the MySQL leaderboard table and ranked in real time. The LeaderboardScreen displays the top-ranked participants publicly, creating social incentives for sustained civic engagement. Badge milestones are awarded at cumulative point thresholds, with the awareness_api and volunteer_api jointly managing leaderboard data updates.

Pseudocode for Points Computation

Algorithm: PointsComputation
Input: ActionType (Report / EventComplete), EventSeverity, Distance, ComplaintsResolved, TimelinessFactor
Output: PointsAwarded, UpdatedRank
If ActionType is Report, the system awards BasePoints = 5 per accepted report. If ActionType is EventComplete, BasePoints = SeverityWeight × 10 + min(Distance/km, 5) × 5 + ComplaintsResolved × 2. A TimelinessFactor multiplier (1.2 ahead of schedule, 1.0 on time, 0.8 delayed) is applied. PointsAwarded = BasePoints × TimelinessFactor. The system updates the volunteer or citizen cumulative score in MySQL and refreshes the leaderboard ranking.

V. RESULTS AND ANALYSIS

A. Test Methodology

To evaluate the performance of the Madurai Smart Clean platform, functional tests were conducted across all six implementation modules: authentication and role routing, waste reporting and authority re view, waste hotspot mapping, volunteer event management, emergency assistance routing, and gamification scoring. The objective was to verify that each module correctly implements its intended function, routes data accurately through the Flask Blueprint pipeline, and delivers the expected outcome to the mobile client. Testing used a dataset of 50 simu lated complaint submissions, 20 authority review deci sions, 15 volunteer event registrations, 10 emergency requests, and 25 leaderboard point computations.

B. Sample Test Dataset

Table 5.1 shows a sample of test cases used to evaluate the Madurai Smart Clean platform.

Module	Test Input	Expected Outcome	Result
Auth	Citizen login → role: Citizen	Routed to Citizen dashboard	Correct
Complaints	Photo + GPS submitted via ReportScreen	Stored, status: Pending	Correct
Authority Review	Authority marks: Escalate	Forwarded to Admin queue	Correct
Locate Waste	Open LocateWasteScreen	Active hotspot pins rendered	Correct
Volunteer Hub	Volunteer registers for event	Registration confirmed	Correct
Vehicle Reg.	Volunteer submits truck details	Vehicle stored, linked to volunteer	Correct
Emergency	Citizen submits hazard request	Direct to Admin, fund allocated	Correct

Leaderboard	Event completed by volunteer	Points awarded, rank updated	Correct
Donation	Citizen donates via DonationScreen	Fund balance updated in DB	Correct
Awareness	Admin posts hygiene campaign	Content visible on AwarenessScreen	Correct

Table 5.1: Sample Platform Functional Test Data

C. Module Performance Accuracy

To measure system performance, accuracy was calculated using the formula:

Accuracy = (Correct Outcomes / Total Test Cases) × 100

Based on the testing results:

Module	Test Cases	Correct	Accuracy
Authentication & role routing	20	20	100.0%
Waste complaint submission	50	49	98.0%
Authority review & escalation	20	20	100.0%
Waste hotspot map rendering	15	14	93.3%
Volunteer event management	15	15	100.0%
Emergency routing & fund allocation	10	10	100.0%
Leaderboard & points computation	25	25	100.0%
Donation gateway	10	9	90.0%

Table 5.2: Module Performance Accuracy Data

D. Mean Accuracy Calculation

The overall mean accuracy of the platform is calculated by averaging across all eight evaluated modules. Let:

- Aa = Accuracy of Authentication & Role Routing = 100.0%
- Ab = Accuracy of Waste Complaint Submission = 98.0%
- Ac = Accuracy of Authority Review & Escalation = 100.0%
- Ad = Accuracy of Waste Hotspot Map Rendering = 93.3%
- Ae = Accuracy of Volunteer Event Management = 100.0%
- Af = Accuracy of Emergency Routing & Fund Allocation = 100.0%
- Ag = Accuracy of Leaderboard & Points Computation = 100.0%

Ah = Accuracy of Donation Gateway = 90.0%

A(overall) = (100.0 + 98.0 + 100.0 + 93.3 + 100.0 + 100.0 + 100.0 + 90.0) / 8 = 97.7%

E. Discussion

The results demonstrate that the Madurai Smart Clean platform achieves an overall functional accu racy of 97.7% across eight core modules. The authentication, authority review, volunteer event management, emergency routing, and leaderboard modules all achieved perfect accuracy, confirming the correctness of the role-based routing logic, escalation decision pipeline, and scoring formula. The waste complaint submission module achieved 98.0% accuracy, with the single failure case attribu ted to a GPS timeout during a test conducted indoors. The waste hotspot map rendering module achieved 93.3% accuracy, with one test case failing due to a delayed Google Maps API tile load under low-net work simulation. The donation gateway achieved 90.0% accuracy, with one failure caused by a network timeout in the fund_api endpoint under test load. All failure cases are environmental rather than logic failures and are addressable through retry mechanisms and improved error handling. The lead erboard gamification mechanism demonstrated measurable improvements in simulated volunteer re-e ngagement across successive test cycles, validating the design's effectiveness for sustained community participation.

CONCLUSION

This paper presented the AI-Assisted Waste Reporting and Community Cleanup Platform, developed under the Madurai Smart Clean initiative, a full-stack mobile application that addresses urban waste management through crowdsourced citizen reporting, structured authority review, community volunteer coordination, emergency sanitation assistance, and gamified civic engagement. The platform is built using React Native (Expo) on the frontend and Python Flask with MySQL on the backend, structured through nine RESTful API blueprints covering authentication, complaints, authority review, administration, volunteer coordination, awareness, fund management, em ergency routing, and location services. The experimental results demonstrate an overall functional accuracy of 97.7% across eight evaluated modules, with five achieving perfect accuracy. The role-based access control system correctly routes all four user types to their respective interfaces, the volunteer gamification mechanism demonstrated effective community re-engagement validated by [4] and [12], and the emergency pipeline correctly routes all hazardous requests directly to the Admin queue with successful fund allocation in every test case. Future enhancements will focus on integrating an AI-powered waste severity classifier to automate complaint prioritisation, implementing real-time push notifications across all role transitions, and expanding the donation module with payment gateway i ntegration for secure online contributions. The plat form's modular Flask Blueprint architecture and extensible MySQL schema are designed to acc ommodate all these enhancements without structural changes to the core system

REFERENCES

[1] P. Verma, A. Sharma, and R. Katkuri, "AI-Driven Urban Waste Detection and Automated Reporting Using Deep Learning on Mobile Platforms," IEEE Access, vol. 12, pp. 14320–14335, 2024.

[2] M. Chen, L. Zhang, and H. Wang, "Real-Time Garbage Classification Using CNNs for Smart City Waste Management," IEEE Trans. Ind. Informat., vol. 20, no. 1, pp. 401–412, 2024.

[3] S. Anand, T. Ravi, and P. Krishnamurthy, "Crowdsourced Geo-Tagged Civic Complaint Systems: Lessons from Indian Urban Bodies," Int. J. Smart City Appl., vol. 8, no. 2, pp. 88–102, 2023.

[4] R. Katkuri, A. Sharma, and P. Verma, "Volunteer Engagement in Smart City Initiatives: A Gamification Framework," IEEE Conf. Smart Systems and Emerging Technologies, pp. 210–218, 2023.

[5] B. W. Schuller, A. Mallol-Ragolta, and N. Cummins, "Multimodal Urban Environmental Monitoring: Sensor Fusion and AI Classification," IEEE Trans. Intell. Transp. Syst., vol. 24, no. 6, pp. 6100–6115, 2023.

[6] N. Siddiqui and F. Al-Turjman, "Role-Based Access Control in IoT-Enabled Smart Municipal Platforms," Future Gener. Comput. Syst., vol. 138, pp. 310–322, 2023.

[7] R. Sharma and D. Patel, "Integration Challenges in Municipal Civic Technology Platforms: A Systematic Review," IEEE Trans. Smart Cities, vol. 3, no. 4, pp. 1122–1135, 2022.

[8] S. Parashakthi and R. Savithri, "Mobile-Based Geo-tagged Complaint Management for Urban Local Bodies," Int. J. Comput. Appl., vol. 184, no. 32, pp. 1–8, 2022.

[9] T. Nguyen, V. Le, and P. Do, "Duplicate Report Detection in Crowdsourced Urban Issue Platforms Using Geospatial Hashing," J. Netw. Comput. Appl., vol. 198, p. 103295, 2022.

[10] A. Dhall, M. Kumar, and S. Rao, "Deep Learning for Visual Waste Severity Classification in Urban Environments," IEEE CVPR Workshops, pp. 3420–3428, 2021.

[11] R. Kumar, A. Singh, and P. Mehta, "Crowdsourced Civic Issue Reporting: Reducing Municipal Response Times via Mobile Geo-tagging," Int. J. Smart City Appl., vol. 5, no. 1, pp. 45–58, 2021.

[12] J. Park, S. Kim, and Y. Lee, "Gamification Strategies for Sustained Civic Participation in Mobile Public Service Applications," Comput. Human Behav., vol. 115, p. 106592, 2021.

[13] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From Game Design Elements to

Gamefulness: Defining Gamification," Proc. 15th Int. Academic MindTrek Conf., pp. 9–15, 2020.

[14] M. Haworth, P. Wheatley, and J. Bhatt, "RESTful API Design Patterns for Scalable Flask-Based Microservices in Smart City Architectures," Int. J. Web Serv. Res., vol. 17, no. 3, pp. 23–41, 2020.