



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: <https://doi.org/10.22214/ijraset.2026.85261>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

Unified Grievance System: An AI-Assisted Web Platform for Complaint Management, Department Assignment, and Transparent Resolution Tracking

Mr. Vaibhav Shrivastava¹, Devyani Dongre², Nikita Yahake³, Tanvi Tilwani⁴, Parul Mishra⁵, Pratiksha Gaidhane⁶,
Keshar Bingewar⁷

^{1, 2, 3, 4, 5, 6, 7}Department of Computer Science and Engineering, G H Rasoni University, Amravati, Maharashtra, India

Abstract: *The Unified Grievance System is a web-based prototype intended to improve the reporting and handling of civic and residential-community complaints. The platform provides a common interface through which residents can submit complaints, attach supporting images, view progress, and receive status updates, while administrators and staff can review, assign, and manage cases. The current implementation uses a React-based client, Supabase authentication and storage services, a FastAPI backend, and a Gemini-based AI analysis path intended to assist with complaint interpretation and department assignment. The system is designed around issue categories such as potholes, garbage overflow, broken street lights, and water leakage, with department options including Electrical Department, Water Department, General Maintenance, Road Maintenance, Sanitation. Administrative views support complaint summaries, category and monthly-trend analytics, resident management, and configuration of selected system settings. The paper describes the system architecture, workflow, implementation choices, AI-assisted routing approach, and a proposed evaluation methodology. The prototype is still under development and AI analysis behaviour has not been validated on a labelled dataset. Therefore, no classification accuracy, response-time improvement, or user-satisfaction result is claimed. The next steps are to stabilize AI integration, complete a complaint activity timeline, test role-based access and notifications, and evaluate the system using realistic complaint records.*

Keywords: *Public grievance redressal, complaint management, civic technology, artificial intelligence, department assignment, web application, FastAPI, React, Supabase, Gemini*

I. INTRODUCTION

Public grievance redressal is an important part of responsive service delivery. Residents commonly encounter issues such as overflowing garbage, damaged roads, broken street lights, water leakage, and maintenance problems. When complaints are reported through separate channels, informal messages, or manual registers, it can be difficult to maintain a consistent record of what was reported, who is responsible, what action was taken, and whether the issue was resolved.

Digital civic-reporting platforms demonstrate how a common reporting interface can connect residents with the responsible authority. FixMyStreet, for example, allows people to report local problems and routes reports to the relevant local authority [1]. Research has also examined automatic classification of citizen requests, including a study of transportation-related requests using deep learning [2]. More recent work explores AI-supported grievance analysis, routing, prioritization, and transparency [3]–[5]. These systems provide useful design context, but their reported results cannot be transferred to the present prototype.

The Unified Grievance System is a desktop-oriented web application designed for residential societies and colonies. It aims to centralize complaint submission and administrative follow-up in one system. A resident can create a complaint and follow its status; administrators can inspect incoming complaints, assign departments, manage residents, and review aggregate patterns. An AI analysis component is intended to assist with understanding the complaint and recommending an appropriate department, while the final workflow remains subject to administrative oversight.

A. Research Contribution

- 1) A unified web workflow for complaint submission, review, department assignment, status updates, and resolution tracking.
- 2) An AI-assisted analysis path intended to interpret complaint text and support automatic department assignment.
- 3) Role-oriented resident and administrator experiences, including resident management and configurable administrative settings.
- 4) A proposed evaluation protocol that separates implemented functionality from unmeasured performance.

The rest of this paper is organized as follows. Section II reviews related work; Section III states the research gap and objectives; Section IV presents the architecture; Section V describes the methodology and implementation; Section VI gives the workflow algorithm; Section VII lists configuration and feature status; Section VIII compares the system with representative platforms; Section IX proposes experiments; Sections X–XII discuss current status, limitations, and future work; and Section XIII concludes the paper.

II. RELATED WORK

A. Digital Civic Reporting Platforms

FixMyStreet is a public problem-reporting platform that lets people report local issues and directs reports to the responsible local authority [1]. Its approach illustrates the value of simplifying the reporting path and reducing uncertainty about which authority should receive a complaint.

The Unified Grievance System follows a related principle at a smaller residential-community scope: complaints are submitted through one interface and linked to a department or responsible staff member.

B. Automatic Classification of Citizen Requests

Kim and Hong studied automatic classification of citizen requests related to transportation and examined deep-learning approaches for handling unstructured requests [2]. This work is relevant because complaint descriptions are often written in natural language rather than in a fixed technical vocabulary. A classification component can potentially reduce repetitive manual triage, but its suitability depends on the data, category definitions, language, and evaluation conditions.

C. AI-Assisted Grievance Analysis

Recent research proposes AI-enabled grievance systems that combine complaint analysis, urgency or priority estimation, routing, and tracking [3], [4]. Other work investigates multimodal complaint routing using both text and images [5]. These studies indicate a broader direction in which AI assists the operational workflow rather than merely storing reports. However, methods and results differ across datasets and implementations. The present paper therefore uses these publications as related work, not as evidence of performance by the Unified Grievance System.

D. Position of the Present Work

The present project focuses on a web-based complaint lifecycle for residential societies and colonies, combining resident-facing reporting, administrator-facing management, an AI analysis integration, status visibility, and configuration settings. Its contribution is the integration of these functions into a working project prototype. It does not claim to introduce a new machine-learning architecture, and it does not claim superiority over the cited systems.

III. RESEARCH GAP AND OBJECTIVES

Complaint portals can digitize submissions without necessarily providing a complete, traceable process from initial report to closure. Manual triage may require administrators to read every complaint, decide which department should handle it, and communicate updates separately. If status changes and staff remarks are not recorded in a consistent history, it becomes harder to understand how a complaint progressed.

A. Research Gap

- A complaint record needs more than a submission form: it should connect reporting, review, assignment, work updates, and resolution.
- Natural-language descriptions may be ambiguous or incomplete, creating a need for AI-assisted interpretation while retaining human review.
- Residents need a clear view of complaint status, while administrators need aggregate views of pending, resolved, and category-level complaints.
- System claims should be supported by measured tests; an integrated prototype alone does not establish classification accuracy or faster resolution.

B. Project Objectives

- Provide a centralized web interface for residents to submit and track complaints.
- Support complaint categories relevant to residential and civic maintenance, including potholes, garbage overflow, broken street lights, and water leakage.
- Connect complaint processing to department options such as Electrical Department, Water Department, General Maintenance, Road Maintenance, Sanitation, and Parks & Gardening.
- Integrate an AI analysis path to assist with complaint interpretation and automatic department assignment.
- Provide administrators with complaint summaries, category distribution, monthly trends, complaint management, and resident-management functions.
- Maintain a traceable complaint lifecycle, including status changes, remarks, and resolution details.
- Evaluate functional correctness, AI recommendations, access control, notification behaviour, and usability using a documented test plan.

IV. SYSTEM ARCHITECTURE AND DESIGN

The system is organized into a client layer, authentication and storage services, a backend API layer, an AI analysis component, and complaint-management data. The React client presents the resident and administrative interfaces. Supabase is used for authentication and file storage, and the project also uses Supabase-backed application data in the current implementation. A FastAPI backend exposes server-side functionality and connects complaint-processing requests to the AI analysis path. Gemini is the configured model family for the AI component; its output is intended to support categorization and department assignment rather than to replace administrative responsibility.

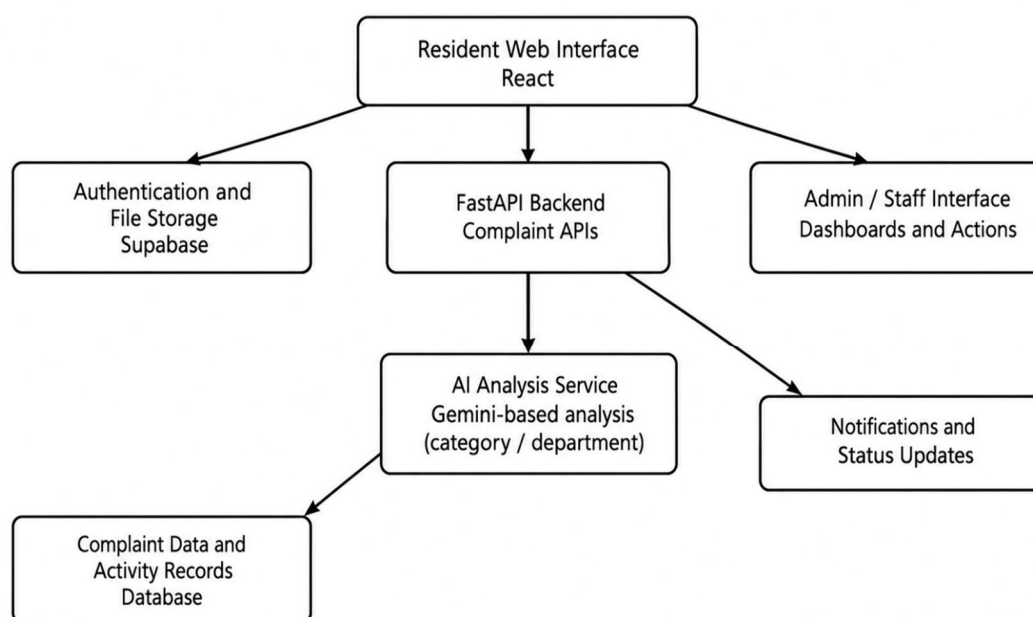


Fig. 1. High-level architecture of the Unified Grievance System.

Fig. 1. High-level architecture of the Unified Grievance System. The diagram represents the intended integration boundaries; exact data flow depends on the deployed configuration.

A. Complaint Processing Flow

A resident submits a complaint with a description, category information where applicable, and optional supporting evidence. The application validates the input and stores the complaint. The backend can invoke AI analysis to interpret the report and suggest a category or department. The complaint is then available for administrator review and assignment. Staff or administrators update progress, remarks, and status, and the resident can check the latest state.

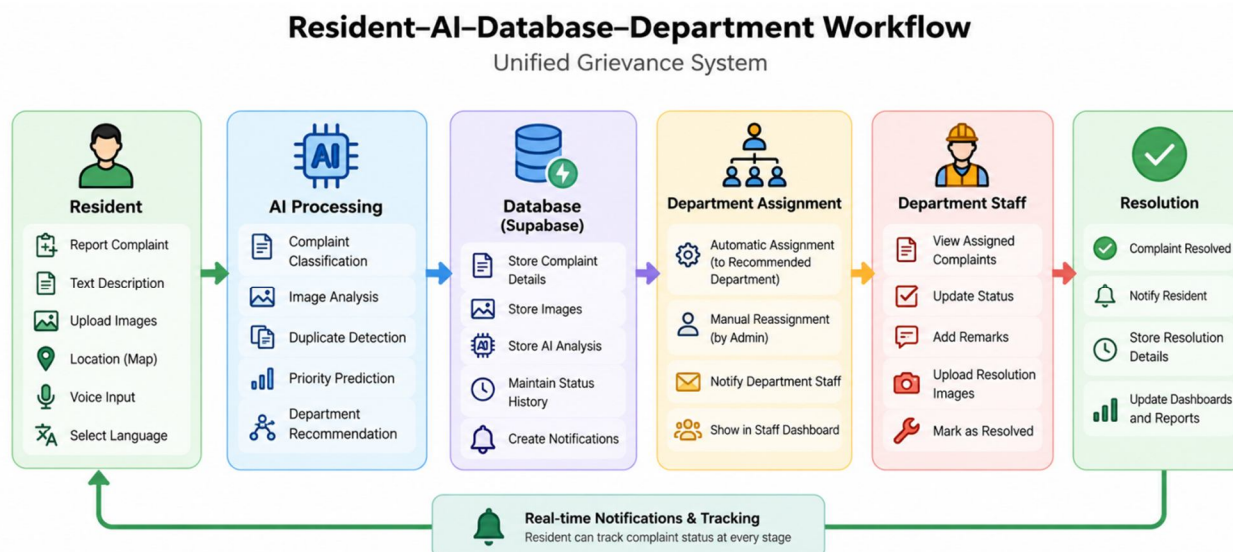


Fig. 2. Intended complaint lifecycle from submission to tracking and resolution.

B. User Roles

The resident-facing workflow is intended for account holders who submit and monitor their own complaints. The administrator workflow includes viewing and managing complaints, assigning departments, updating status, managing resident accounts, and reviewing analytics. Staff-facing lifecycle history is a planned extension: it should record relevant events such as complaint creation, AI analysis, assignment, work updates, remarks, and resolution. Access to each operation must be enforced by backend/database rules, not only by hiding controls in the interface.

C. Department Model

The current project context identifies six department options: Electrical Department, Water Department, General Maintenance, Road Maintenance, Sanitation, and Parks & Gardening. Categories such as broken street light, water leakage, pothole, and garbage overflow provide examples of issues that may map to those departments. The mapping should be configurable and reviewed against the actual responsibilities of the residential community.

V. METHODOLOGY AND IMPLEMENTATION

The system follows an iterative software-development approach. The interface and backend are developed as separate components, with authentication, complaint data, file handling, AI analysis, and administrator tools integrated progressively. This section describes the known implementation choices and does not treat unfinished features as completed or measured.

Layer	Technology / component	Purpose
Frontend	React	Resident and administrator web interfaces.
Authentication	Supabase Auth	Email/password authentication and account-related flows.
File storage	Supabase Storage	Storage and retrieval of complaint evidence and profile images, subject to bucket and access configuration.
Backend API	Python, FastAPI,	Server-side endpoints and integration logic for complaint processing.
AI analysis	Google Gemini model integration	Intended natural-language complaint analysis and department-assignment assistance.
Application data	Supabase tables / configured database	Profiles, complaints, settings, and related application records in the current project setup.
Development	Git and GitHub; VS Code	Source control and application development.

A. Authentication and Access

The current authentication flow includes email/password login and registration, email-verification handling, and a resend-verification path. Google OAuth has also been included in the latest login work. Phone-number login should not be presented as available unless phone authentication is configured and tested. Password recovery should likewise be described as pending until the forgot-password flow is implemented and verified.

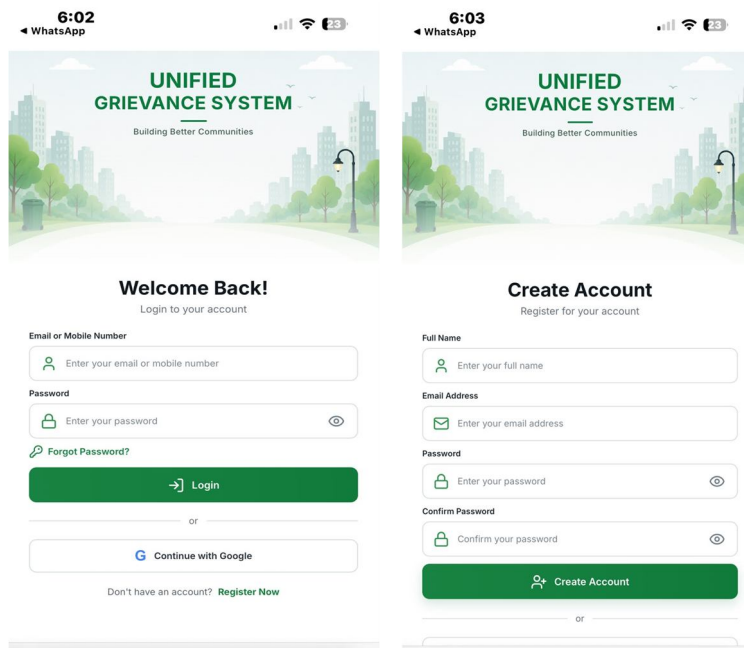


Fig. 3. Authentication interface of the Unified Grievance System, showing the login and registration pages. The interface supports email/password authentication, email verification, and Google sign-in, providing secure access for registered users and a simple account creation process for new residents.

B. Complaint Management

The complaint-management interface is intended to show complaint details and enable authorized administrators to assign a department, change status, and add comments. Complaint records should have stable identifiers and should retain links to uploaded evidence. When evidence is stored, the database should retain the correct storage path or object key rather than relying on an outdated public URL. Access rules must allow authorized viewers to retrieve the appropriate file while preventing unrelated users from accessing private material.

C. AI-Assisted Analysis and Assignment

The backend integration uses a Gemini model attempt to analyze complaint content. The intended output is a structured recommendation, such as a category and responsible department, with a confidence or explanation field only if the model and application reliably produce one. The backend should validate the model response against the allowed department list, handle malformed or empty output, record failures, and fall back to manual assignment when analysis is unavailable. A displayed confidence percentage must not be treated as calibrated probability unless calibration has been evaluated.

During development, the backend has logged that automatic department assignment is enabled and has listed the available departments. Logs also showed a Gemini model attempt and a warning concerning direct automatic function calling in a generate-content call. These observations demonstrate that the integration path is being invoked, but they do not prove that the complete analysis result is valid, persisted, or shown correctly in the interface. That end-to-end behaviour requires testing.

D. Administrative Dashboard and Settings

The planned administrator dashboard includes total complaints, pending complaints, resolved complaints, complaints by category, monthly trends, a complaints map view, and navigation for categories, residents, departments, reports, and settings. Administrative settings currently include controls for AI auto-assignment, admin reassignment, complaint deletion, notification types, and theme

selection. A setting should affect backend behaviour only when the relevant code checks and enforces it; a toggle in the user interface alone is not sufficient.

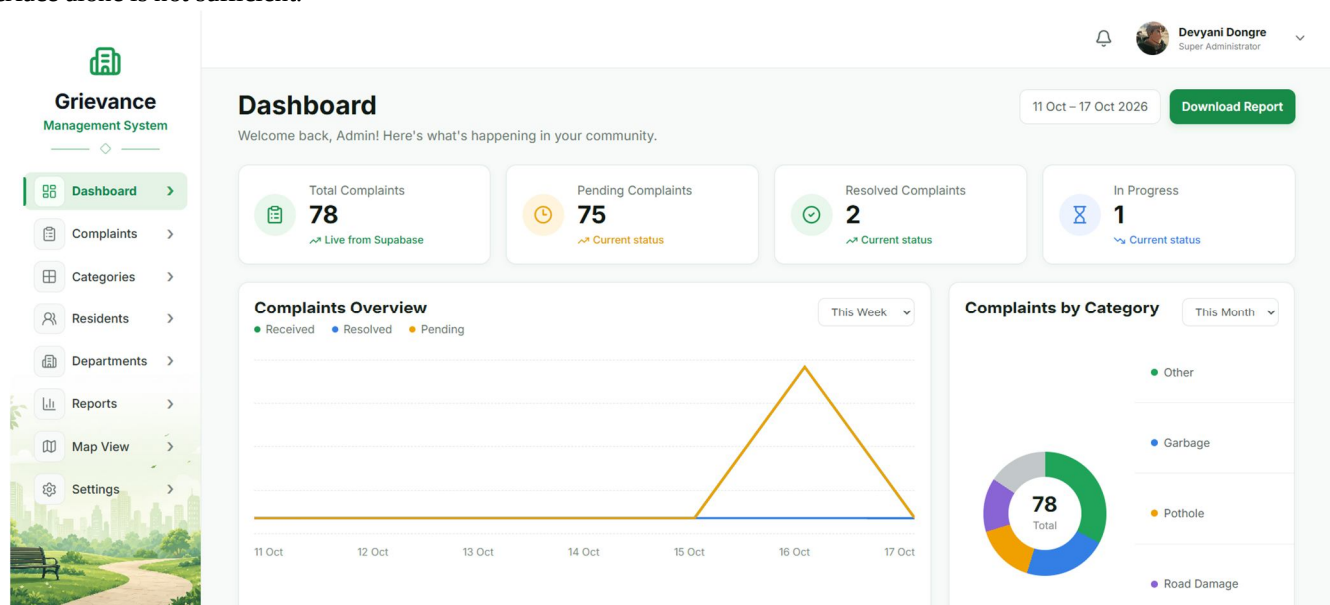


Fig. 4. Administrative dashboard displaying total, pending, resolved, and in-progress complaints, along with complaint trends and category-wise statistics for monitoring grievance management activities.

E. Notifications and Activity History

The project includes database-trigger/function work for complaint-created notifications and related status events. Notification behaviour should be tested for duplicate delivery, missing recipients, permissions, and transaction failures. A staff complaint history or activity timeline is a planned extension and should record timestamped lifecycle events rather than only displaying the current status. Suggested events include submitted, AI analysis completed or failed, department assigned, reassigned, staff remark added, status changed, and resolved.

VI. WORKFLOW ALGORITHM

Algorithm 1 outlines the intended processing logic for a complaint. It is a workflow specification, not a claim that every branch is already implemented and verified.

Algorithm 1. Complaint intake and routing

Input: complaint C containing resident, description, optional category, optional image

- 1: Validate required fields and confirm the requester is authenticated.
- 2: Store complaint C and its evidence reference.
- 3: If AI auto-assignment is enabled, send the complaint text and allowed department list to the analysis service.
- 4: Validate the returned category and department against the allowed values.
- 5: If the AI output is valid, store the recommendation and analysis status; otherwise record failure and keep the complaint available for manual review.
- 6: If automatic assignment is enabled and policy permits, assign the recommended department; otherwise await administrator action.
- 7: Record an activity event for creation, analysis, assignment, or failure.
- 8: Notify the relevant resident or staff member when the corresponding notification rule is enabled.
- 9: On authorized status/remark updates, persist the update and append a timestamped activity event.
- 10: When resolution is confirmed, record the resolution state and expose the updated status to the resident.

A. Department Validation

AI output must be treated as untrusted application input. The backend should parse a structured response, reject unknown department names, handle incomplete fields, and retain the original complaint text. If confidence is shown, its meaning and source should be explicit. The system should preserve a manual correction path so that an administrator can change an incorrect recommendation.

B. Lifecycle Event Model

A minimal activity record can contain complaint_id, event_type, old_value, new_value, remarks, actor_id, created_at, and optional metadata. Event records should be append-oriented so that later changes do not erase the history of earlier actions. The exact schema must be aligned with the project's existing database before implementation.

VII. SYSTEM CONFIGURATION AND FEATURE STATUS

Table II separates the known implementation context from work that still needs verification. A feature is not classified as fully **complete** merely because a UI control or database table exists.

Feature	Current status	Qualification
React web interface	Implemented / in active development	Resident and administrator screens exist; end-to-end checks remain necessary.
Supabase email/password authentication	Implemented / being refined	Verification and resend flows have been worked on; test account states individually.
Google OAuth	Integrated in latest login work	Requires provider configuration and successful sign-in testing.
Complaint submission and tracking	Core project feature	Validate persistence, ownership rules, and status visibility.
Complaint image storage	Needs end-to-end verification	Cross-account visibility depends on correct object paths and storage policies.
FastAPI backend	Running in development	Uvicorn startup has been observed; endpoint correctness requires separate tests.
Gemini AI analysis	Integration under debugging	Model invocation has been observed; correct structured output and persistence are not yet established.
Automatic department assignment	Configured path present; verify end-to-end	Must test model output, allowed department mapping, and fallback behaviour.
Admin dashboard and settings	Implemented in parts	Each metric and toggle must be checked against database/backend behaviour.
Notifications	Database functions/triggers in progress	Test recipient rules, duplicate events, and delivery.
Staff activity timeline	Planned next feature	Should cover creation through assignment, work updates, remarks, and resolution.
Measured evaluation and user study	Not yet completed	No performance numbers are reported in this paper.

Table II. Feature status based on the current project context; it should be updated after a fresh end-to-end verification pass.

VIII. COMPARISON WITH EXISTING SYSTEMS

Table III provides a functional comparison based on the cited publications and public descriptions. “Not stated” means that the reviewed source did not establish the feature; it does not mean the system lacks it. The comparison is not a benchmark because the Unified Grievance System has not completed a controlled performance evaluation.

System / work	Main scope	AI / routing	Tracking / transparency	Evidence basis
FixMyStreet [1]	Public local-issue reporting	Routes reports to responsible local authority	Public reporting and follow-up mechanisms	Platform description and related research
Kim and Hong [2]	Transportation-related citizen requests	Studies deep-learning classification	Classification-focused study	Published research article
AI-powered grievance framework [3]	Petition analysis and grievance routing	AI analysis, prioritization and routing proposed	Transparent tracking proposed	Published/preprint description; methods/results are source-specific
Multimodal smart complaint routing [5]	Public issue reports with text and images	Combines text and image signals for routing	Workflow-oriented	Conference publication abstract
Unified Grievance System (this work)	Residential-community complaint lifecycle	Gemini-based analysis path intended for category/department support	Resident status tracking; activity timeline planned	Current prototype and project implementation context

The main distinction is the intended scope and integration. The Unified Grievance System combines resident reporting, administrator tools, department management, AI-assisted analysis, and configurable settings for a residential-community workflow. Whether the AI path reduces misrouting or improves resolution time remains an empirical question.

IX. EXPERIMENTAL METHODOLOGY

No labelled evaluation dataset or controlled user study has been completed for this paper. The following protocol defines how the system should be evaluated before performance claims are made.

A. Test Setup and Reproducibility

Record the frontend and backend versions, Python and package versions, model identifier, database schema version, browser, operating system, network conditions, and relevant configuration settings. Use a test environment with synthetic or appropriately de-identified complaints. Do not include residents’ private details in published results.

B. Functional Testing

Create test cases for registration, email verification, login, logout, complaint creation, evidence upload, complaint ownership, status changes, department assignment, resident blocking/unblocking, settings, and notification generation. Record expected and observed outcomes, severity of failures, and reproducible steps.

C. AI Classification and Department Assignment

Build a labelled test set of complaint descriptions covering all supported departments and common ambiguous cases. Labels should be reviewed by people familiar with the community’s responsibilities. Compare AI recommendations with the reference labels using accuracy, per-class precision, recall, F1-score, and a confusion matrix. Report the number of examples, category distribution, model version, prompt/configuration, and treatment of invalid responses. Do not report a single overall accuracy without class-level context.

D. Assignment and Fallback Reliability

Test valid, incomplete, malformed, contradictory, and unavailable model responses. Measure the proportion of complaints that receive a valid recommendation, the proportion requiring manual review, and the number of incorrect or unsupported department values. Verify that disabling AI auto-assignment prevents automatic assignment while still allowing a complaint to be processed.

E. Timing and End-to-End Latency

Log timestamps for complaint submission, backend receipt, AI request start, AI response, persistence, assignment, notification creation, and interface update. Report median and 95th-percentile latency over repeated trials, along with failures and test conditions. Avoid claiming real-time operation unless the latency requirement is defined and measured.

F. Activity Timeline and Auditability

Use scripted cases that progress from creation to assignment, reassignment, staff remarks, status changes, and resolution. Verify that each event is timestamped, linked to the correct complaint, associated with the responsible actor, and displayed in chronological order. Check that unauthorized users cannot edit or delete historical events.

G. Usability Evaluation

After institutional and project permissions are obtained, conduct task-based testing with representative residents and administrators. Tasks can include submitting a complaint, finding its current status, assigning a department, and locating a previous remark. Measure task completion, time on task, error rate, and structured feedback. Clearly report participant count and study limitations.

X. PROTOTYPE STATUS AND DEVELOPMENT OBSERVATIONS

The project has a React web interface and a FastAPI backend. Development logs have shown the backend starting successfully under Uvicorn and have shown the AI assignment path enabled with a list of available departments. The backend also attempted to call a Gemini model. These are development observations, not quantitative evaluation results.

The AI analysis path has required debugging, and a model-call warning concerning automatic function calling was observed. This means the integration should be reviewed for supported SDK usage, response parsing, error handling, and consistency between backend output and frontend display. It would be incorrect to state that AI analysis is fully reliable until representative complaints are tested end to end.

Image access has also required attention: an image uploaded by one account was not consistently visible from another account, and an older public URL was reported as unavailable. Such behaviour can arise from stale URLs, incorrect storage paths, bucket policies, or row-level access rules. The correct diagnosis must be established from the active storage configuration and database records rather than assumed.

The current project context also includes login verification and resend-verification handling, administrative settings, resident management, and notification SQL work. The staff complaint history/activity timeline is identified as the next development task. No accuracy, processing-speed, resolution-time, or user-satisfaction figures are reported because no controlled measurement has been completed.

XI. LIMITATIONS

- 1) AI analysis and automatic department assignment have not been evaluated on a labelled complaint dataset.
- 2) Model output can be incomplete, ambiguous, inconsistent, or outside the permitted department list; backend validation and manual fallback are necessary.
- 3) The quality of assignment depends on the clarity of complaint descriptions, supported categories, and the model/prompt configuration.
- 4) Authentication and storage configuration require end-to-end security testing, especially when complaint evidence must be visible to authorized users across accounts.
- 5) Notification triggers and settings require testing to ensure that events are not lost or duplicated and that the correct recipient is selected.
- 6) The complete staff activity timeline is planned and should not be described as an implemented audit trail until it is built and tested.
- 7) Dashboard analytics are only as reliable as the underlying data, date handling, status definitions, and aggregation queries.

- 8) No formal user study, performance benchmark, or comparative experiment has been completed.
- 9) The prototype is intended for residential-community complaint management and is not a substitute for emergency services or official municipal reporting channels.

XII. FUTURE SCOPE

- 1) Complete the staff complaint history/activity timeline from creation through AI analysis, assignment, staff action, remarks, status changes, and resolution.
- 2) Stabilize the Gemini integration, validate structured outputs, add robust retry/error handling, and provide a safe manual-assignment fallback.
- 3) Build a labelled local complaint dataset and compare the AI recommendation against human-reviewed department labels.
- 4) Add duplicate-complaint suggestions using text similarity and, if suitable data becomes available, location and time proximity.
- 5) Add transparent priority recommendations with documented rules and administrator override; avoid presenting uncalibrated model scores as probabilities.
- 6) Complete notification workflows for new complaints, AI assignments, status updates, duplicate detection, and new residents.
- 7) Strengthen storage access rules and test image visibility across resident and administrator roles.
- 8) Add analytics such as median time to first response, resolution time by category, unresolved-case age, and department workload, after definitions are standardized.
- 9) Explore multilingual complaint entry and analysis for English, Hindi, and Marathi after reliable evaluation data is available.
- 10) Conduct usability and security testing before deploying the platform for real community use.

XIII. CONCLUSION

The Unified Grievance System is a web-based platform developed to simplify and centralize complaint reporting and management within residential communities. The system provides a common interface through which residents can report civic issues, submit supporting images, record complaint locations, and monitor the progress of their complaints. By bringing residents, administrators, and staff together on a single platform, the system aims to make grievance management more organized, transparent, and accessible. The platform integrates a React-based frontend, Supabase authentication and storage services, a FastAPI backend, and a Gemini-based Artificial Intelligence component. These technologies work together to support essential functions such as complaint registration, user authentication, complaint management, department assignment, status tracking, and administrative monitoring. The system also incorporates features intended to assist administrators in identifying duplicate complaints, analyzing reported issues, and recommending appropriate departments for complaint resolution.

One of the key contributions of the project is the integration of AI-assisted analysis into the complaint management workflow. Instead of relying entirely on manual interpretation, the system is designed to use AI to analyze complaint descriptions and assist in determining the appropriate department. This can help organize incoming complaints and reduce repetitive administrative work. However, the reliability of these AI-assisted functions must be evaluated through systematic testing before their effectiveness can be established.

The system also emphasizes transparency and accountability by allowing complaints to be tracked throughout their lifecycle. Residents can follow complaint progress, while administrators can manage complaints, coordinate departments, monitor resolution status, and oversee user activities. Role-based access control helps separate resident and administrative functions, while authentication and database-level security measures contribute to protecting user information and maintaining controlled access to system resources.

The development of this project demonstrates how modern web technologies and AI-assisted features can be combined to address practical problems in residential community management. By digitizing complaint registration and tracking, the platform provides an organized alternative to fragmented reporting methods, such as verbal complaints or informal communication channels. It also establishes a foundation for maintaining structured complaint records that can support future analysis and administrative decision-making. Although the current prototype implements the core complaint management workflow, certain aspects require further development and validation. These include comprehensive end-to-end testing, evaluation of AI-generated analysis, verification of access permissions across different user accounts, implementation of a complete complaint activity timeline, and improvements to notification reliability. Further enhancements could include external SMS integration, advanced complaint analytics, geographical visualization of recurring issues, multilingual support, and improved AI-assisted prioritization based on complaint severity.

In conclusion, the Unified Grievance System provides a practical foundation for a more structured, transparent, and responsive grievance management process in residential communities. Its modular architecture allows additional features and integrations to be introduced as requirements evolve. With further testing, security verification, and functional improvements, the system has the potential to support more efficient complaint handling, better coordination between residents and administrators, and improved visibility into the resolution of community issues.

XIV. ACKNOWLEDGMENT

We would like to express our sincere gratitude to our project guide, **Mr. Vaibhav Shrivastava**, for his constant guidance, support, and encouragement throughout the development of this project. His valuable suggestions, constructive feedback, and technical insights helped us understand the problem, improve our approach, and enhance the overall quality of the Unified Grievance System. We also extend our heartfelt thanks to the Department of Computer Science and Engineering, G H Rasoni University, Amravati, for providing us with the academic support, infrastructure, and resources required to successfully complete this project. We are grateful to the Honorable Vice Chancellor, Dean – School of Engineering and Technology, project coordinators, and all faculty members for their continuous support and encouragement.

Devyani Dongre contributed to the overall system development, including frontend and backend implementation, AI integration, database management, and system testing. Nikita Yahake contributed to requirements analysis, literature review, UI/UX design, and project documentation. Tanvi Tilwani contributed to the AI-based features, including complaint classification, duplicate detection, and priority prediction, along with model testing and evaluation. Parul Mishra contributed to the development of the admin and staff modules, complaint tracking features, and system integration. Pratiksha Gaidhane contributed to database design, API development, and testing of core functionalities. Keshar Bingewar contributed to research, data collection, analysis of existing systems, preparation of datasets for AI models, and preparation of the report and presentation.

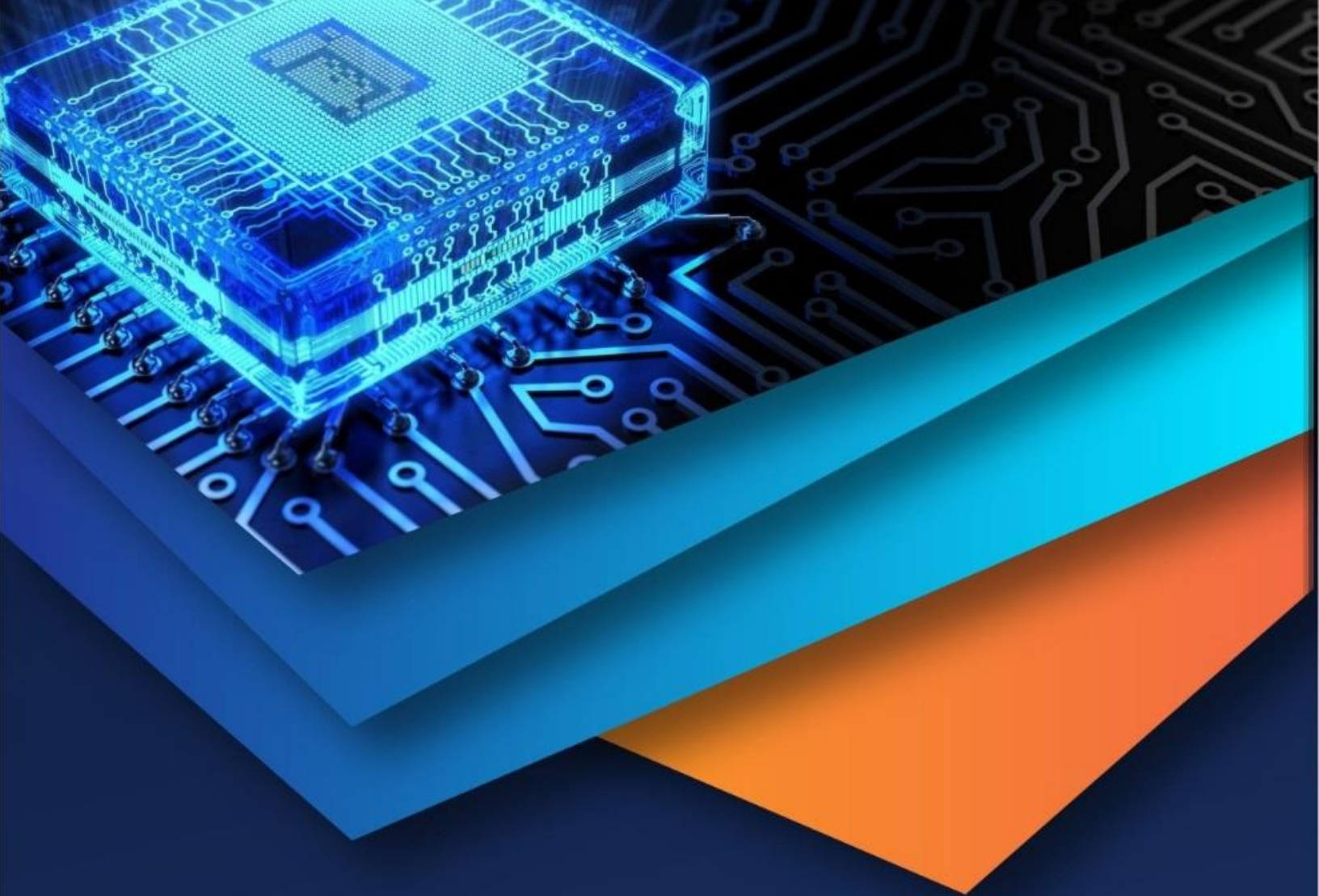
We would also like to acknowledge the open-source technologies, libraries, frameworks, and online resources that supported the development of this project. We are thankful to all individuals who provided valuable suggestions, feedback, and assistance directly or indirectly during different stages of the project.

REFERENCES

- [1] mySociety, "FixMyStreet," online platform and project description. Available: <https://www.fixmystreet.com/>
- [2] N. Kim and S. Hong, "Automatic classification of citizen requests for transportation using deep learning: Case study from Boston city," *Information Processing & Management*, vol. 58, no. 1, article 102410, 2021, doi: 10.1016/j.ipm.2020.102410.
- [3] V. Devi Nandhini et al., "AI-Powered Grievance Management System for Petition Analysis, Prioritization, Routing and Transparent Tracking," conference paper, 2026, doi: 10.5281/zenodo.19408619. Verify publication metadata and peer-review status before formal submission.
- [4] S. C. Rajkumar et al., "A zero-shot LLM framework for multimodal grievance classification, urgency scoring, and abuse detection in civic feedback systems," *Scientific Reports*, vol. 16, article 2299, 2026, doi: 10.1038/s41598-025-32079-7.
- [5] "Smart Complaint Routing Systems for Public Issues Using Multimodal Artificial Intelligence," IEEE conference publication, 2026, doi: 10.1109/I3CTCON68242.2026.11508062.
- [6] React, "React Documentation." Available: <https://react.dev/>
- [7] Supabase, "Supabase Documentation." Available: <https://supabase.com/docs>
- [8] FastAPI, "FastAPI Documentation." Available: <https://fastapi.tiangolo.com/>
- [9] Google, "Gemini API Documentation." Available: <https://ai.google.dev/gemini-api/docs>
- [10] P. Cholke, A. S. Mishra, V. N. Mehar, M. R. Chikhale, H. S. Marathe, and D. Y. Nhalade, "AI-Driven Complaint Management System: An Automated Framework for Prioritizing, Tracking, and Resolving Public Service Complaints Efficiently," in *Proc. 2026 Int. Conf. on Emerging Smart Computing and Informatics (ESCI)*, 2026, doi: 10.1109/ESCI68015.2026.11493166. [DOI](#)
- [11] "Smart Complaint Routing Systems for Public Issues Using Multimodal Artificial Intelligence," in *Proc. 2026 IEEE Int. Conf. for Convergence in Computing Technology (I3CTCON)*, 2026, doi: 10.1109/I3CTCON68242.2026.11508062. [DOI](#)
- [12] R. Kumar B. N., T. S., S. R., S. P. R., and S. R., "CivicFix: Smart Complaint Routing for Urban Solutions," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 14, no. 12, Dec. 2025, doi: 10.17148/IJARCCCE.2025.141222. [IJARCCCE](#)
- [13] A. N., G. Yogeshwaran, U. D. R., M. Kumar C., and S. R., "Smart Civic AI: Intelligent Complaint Management System," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 15, no. 5, 2026, doi: 10.17148/IJARCCCE.2026.155196. [IJARCCCE](#)
- [14] "An Intelligent Government Complaint Prediction Approach," *Big Data Research*, vol. 30, Art. no. 100336, Nov. 2022, doi: 10.1016/j.bdr.2022.100336. [ScienceDirect](#)
- [15] "Proactive Complaint Management in Public Sector Informatics Using AI: A Semantic Pattern Recognition Framework," *Applied Sciences*, vol. 15, no. 12, Art. no. 6673, 2025. [MDPI](#)
- [16] B. S. Menon and D. R. S., "Civi-Snap: A Deep Learning Powered Geotagged Civic Issue Detection and Monitoring Framework," *International Journal of Creative and Open Research in Engineering and Management*, vol. 2, no. 4, 2026. [ResearchGate](#)
- [17] M. M. Mayuranathan, M. Dharshini, A. Aruneswaran, and N. G. Bose, "AI-Driven Predictive Analytics for Citizen Complaint Trends in Smart Cities," *International Journal of Novel Research and Development*, 2026. [IJNRD](#)



- [18] M. Abusalam, M. Sartaz, and S. Ghosh, "AI-COMPLAINT: An Intelligent Web-Based Complaint Management System Utilizing Natural Language Processing for Automated Complaint Classification and Smart Department Routing," *Journal of Science Engineering Technology and Management Science*, vol. 3, no. 6, pp. 147–154, 2026, doi: 10.64771/jsetms.2026.v03.i06.pp147-154. [JSETMS](#)
- [19] G. Khlaf, "An Automated Complaint Analysis and Classification System using Machine Learning and Natural Language Processing," Zenodo, May 2026. [Zenodo](#)
- [20] R. A. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
- [21] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, LNCS, vol. 8693, Springer, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.
- [22] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in *Proc. 12th USENIX Symp. on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.
- [23] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [24] M. C. Stonebraker and L. A. Rowe, "The Design of POSTGRES," in *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 1986, pp. 340–355.
- [25] FixMyStreet, "FixMyStreet: Report, view and discuss local street problems," [Online]. Available: <https://www.fixmystreet.com>
- [26] CivicPlus, "SeeClickFix 311 CRM: Citizen Request Management," [Online]. Available: <https://www.civicplus.com/seeclickfix-311-crm/>



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)