Int. J. Sci. R. Tech., 2026 3(11)
A Multidisciplinary peer-reviewed Journal
www.ijsrtjournal.com [ISSN: 2394-7063]

Original Article

AI-Based Smart Government Grievance System

Aastha Baderiya*, Monika Rajput, Chetan Choudhary, Pawan Shah, Dr. Angad Dixit

Department of Computer Science and Data Science, Oriental College of Technology, Bhopal, Madhya Pradesh, India.

ABSTRACT

Citizens use public grievance systems to report problems involving services such as roads, water supply, sanitation, electricity and local administration. A complaint may be written in everyday language, may contain incomplete information, or may be sent to the wrong department. These issues can make complaint handling harder to organise, particularly when many requests arrive at the same time. India already has the Centralised Public Grievance Redress and Monitoring System (CPGRAMS), which supports online grievance submission, status tracking and feedback. The project described in this paper is an academic prototype concept that explores how a web-based complaint workflow could be supported by optional artificial-intelligence (AI) features. The proposed system uses Python and Flask for the web application, HTML, CSS and JavaScript for the interface, and SQL for structured storage. After a suitable labelled dataset becomes available, natural language processing could be used to suggest a complaint category, priority level and likely department. Similarity matching and short summaries are also considered as support features. These outputs should remain recommendations that an authorised person can review, rather than automatic final decisions. At the time of writing, the team reports that a website/prototype has been developed, but a final dataset and quantitative model evaluation have not been supplied. Therefore, this paper describes the system design and evaluation plan without claiming model accuracy or proven performance gains.

Keywords: e-governance, public grievance, complaint management, natural language processing, text classification, Flask, SQL, human-in-the-loop AI.

INTRODUCTION

Public services affect everyday life, so citizens need a clear way to report service problems and follow the action taken. A grievance might concern a damaged road, irregular water supply, uncollected waste, a faulty streetlight or a delay in a public service. When complaint descriptions are informal or lack key details, staff may need to read each submission, determine its subject and decide which office should handle it. This manual work can become difficult to manage when the number of complaints increases.

Online grievance platforms have already made important parts of this process more accessible. The official CPGRAMS portal allows citizens to submit grievances to public authorities and track them through a registration identifier; it also provides feedback and an appeal process in applicable cases (Department of Administrative Reforms and Public Grievances, n.d.). The National Government Services Portal describes related online services for lodging and checking grievance status. The proposed student project does not replace these services and does not claim official integration. It investigates a smaller, configurable workflow in which AI could help organise complaint text for administrative review.

The main design principle is that automation should assist, not replace, responsible officials. A model may misread a local place name, mixed Hindi-English sentence, sarcasm or urgent context. For that reason, category and priority predictions should be visible as suggestions, with the final assignment and status change controlled by an authorised administrator.

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

1.1 Problem Statement

Citizens may be unsure how to describe a complaint, which department is responsible, or how to follow its progress. Administrators may spend time sorting unstructured text, locating similar reports and preparing summaries. The project explores whether a structured website, supported by carefully evaluated text-processing tools, can make complaint records easier to organise and monitor.

1.2 Objectives

Provide a simple web interface for complaint submission and status tracking.

Store complaint details, department information and status history in a relational SQL database.

Design an NLP pipeline that can recommend complaint categories when labelled training data are available.

Explore priority recommendations and category-to-department routing rules, with human confirmation.

Investigate similarity-based duplicate alerts and short summaries for administrative convenience.

Plan Hindi, English and mixed-language support, then evaluate each language setting separately.

Present complaint counts and status information through an administrator dashboard.

Evaluate the system using a documented test set and suitable metrics before making performance claims.

1.3 Scope

The current scope is an academic web prototype intended for demonstration and further development. It is not an official government service, it is not connected to CPGRAMS unless a separately authorised integration is established, and it should not be used to handle real emergencies. Any real-world deployment would require legal and institutional approval, privacy safeguards, accessibility testing, security review and operational support.

2. Related Work and Background

2.1 Existing public grievance services

CPGRAMS demonstrates that online grievance submission, registration-based tracking and citizen feedback are established parts of public grievance management. Official government pages also describe role-based access for ministries and states and an appeal route for eligible cases. These features provide a useful baseline for understanding the expected lifecycle of a complaint. The project therefore focuses on a prototype workflow and the possible addition of AI-assisted organisation, rather than presenting basic online tracking as a new contribution.

The Open Government Data (OGD) Platform publishes aggregate grievance resources, including year-wise figures for grievances received and redressed on CPGRAMS from 2020 to 2024 and state/UT-wise resources. Such data can help describe overall workload trends. However, aggregate counts do not contain the individual complaint text and labels needed to train a complaint-classification model. A case-level dataset with suitable permissions and reliable labels would be required for that task.

2.2 AI and text classification

Text classification maps a written document to one of a predefined set of categories. In this project, categories might represent service areas such as roads, sanitation, water supply or electricity, but the final taxonomy must be agreed with the intended users. A common baseline converts text into numerical features using Term Frequency–Inverse Document Frequency (TF-IDF). TF-IDF gives greater weight to terms that are informative for a document relative to the wider collection; the scikit-learn documentation describes its use in document classification and information retrieval. A classifier such as Logistic Regression or a linear Support Vector Machine can then be trained on examples whose categories are known.

More advanced multilingual or transformer-based models may eventually handle context and

language variation better, but they also require careful evaluation and can be harder to explain. A simple baseline is valuable because it is easier to reproduce and provides a fair comparison for more complex models. Model selection should be driven by a held-out evaluation set, not by assumptions that a newer model must be better.

2.3 Design implications from prior systems

Use a clearly defined category list and document who is responsible for maintaining it.

Keep the official complaint record separate from model-generated suggestions.

Measure errors by category and language, rather than reporting accuracy alone.

Treat duplicate detection as an alert for review, because similar wording does not always mean the same incident.

Provide an understandable status history and a way for authorised staff to correct mistakes.

3. Proposed System

The proposed system combines a conventional complaint-management workflow with optional AI assistance. A citizen enters the complaint through a browser. The application validates required fields and stores the record in SQL. If the AI component is implemented and evaluated, it can suggest a category, priority and department based on the complaint text and configured rules. An administrator reviews the suggestion, updates the record and the citizen can check progress using the complaint reference.



Figure 1. Proposed complaint workflow. AI outputs are suggestions; final assignment remains under authorised human review.

*The AI recommendation step is conditional on the relevant component being implemented, connected to a suitable dataset and tested.

3.1 Main modules

Module	Purpose	Important design point
Citizen interface	Submit complaint, receive reference, track status and provide feedback.	Forms should explain required fields and avoid collecting unnecessary personal data.
Complaint validation	Check required fields and basic input quality before saving.	Validation should give clear error messages and preserve the original complaint meaning.
Complaint database	Store complaint details, department, timestamps and status history.	Access must be role-based; changes should be traceable.

AI/NLP assistance	Suggest category, priority, similar reports or a short summary.	Only enable features after evaluation; show uncertainty and allow correction.
Administrator dashboard	Review records, assign departments and	Human review is required for routing,
Notifications and feedback	update status.	priority and closure.

3.2 User roles

Citizen: submits a complaint, receives a tracking reference, checks progress and shares feedback. Administrator or authorised officer: reviews complaint records, confirms the department and priority, updates status, and corrects inaccurate AI suggestions. Technical administrator: manages accounts, category mappings, backups and application configuration. The precise role model should be adjusted to the final implementation.

3.3 Example use case

Suppose a citizen reports that waste has not been collected from a neighbourhood for several days. The system stores the description and may suggest "sanitation" as the category. A rule-based mapping can show the department configured for that service area. If the classifier is uncertain, the record should be marked for manual review rather than assigned automatically.

The authorised officer confirms the assignment and updates the complaint status as work progresses.

4. Technology and Methodology

4.1 Technology stack

Technology	Role in the proposed project
Python	Backend logic and possible text-processing / machine-learning components.
Flask	Web routes, form handling and application workflow.
HTML, CSS and JavaScript	Citizen forms, tracking views and dashboard interface.
SQL database	Structured complaint records, users, department mappings and status history.
scikit-learn (planned for baseline AI)	TF-IDF feature extraction and baseline text-classification models, if implemented.

4.2 Dataset plan and constraints

The team has identified official aggregate CPGRAMS datasets on the OGD Platform as a source for background statistics. The year-wise resource lists fields such as year, carried-forward cases, receipts and disposed cases. This is useful for describing workload, but it is not sufficient for supervised complaint-text classification. Before training a model, the team must identify a dataset that actually contains complaint descriptions and reliable category labels,

confirm its licence or permission, and document its fields and limitations.

Potential case-level fields, if lawfully available, include complaint text, category, responsible department, service area, submission date and status. Personal identifiers should be excluded from the training data wherever possible. The team should not assume that restricted government records are openly downloadable. If suitable case-level data cannot be obtained, the paper should describe the AI component as a design proposal or use a clearly labelled, ethically prepared demonstration dataset rather than implying access to real citizen records.

4.3 Data preprocessing

1. Inspect the schema, missing values, class balance, label quality and data-use conditions.

2. Remove exact duplicate records and document how inconsistent labels are resolved.

3. Normalise whitespace and formatting while retaining meaningful terms, negation and local place names.

4. Define how Hindi, English and code-mixed text will be handled; do not call the system bilingual until this is tested.

5. Separate training and test records before model tuning, keeping near-duplicate complaints from leaking across splits where feasible.

6. Record all preprocessing choices so that results can be reproduced.

4.4 Proposed AI pipeline

A practical first experiment would use TF-IDF features with a simple multi-class model such as Logistic Regression or Linear SVM. The same split and evaluation metrics should be used when comparing models. If enough labelled data become available, later experiments could consider multilingual embeddings or transformer models. A priority recommendation may be based initially on transparent rules agreed by the project team; it should not be inferred solely from historical closure time, because that could reproduce past delays or bias.

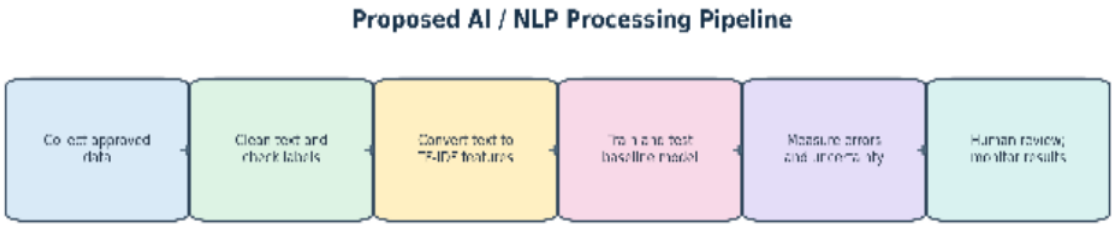


Figure 2. Proposed AI/NLP pipeline. The stages are a plan for evaluation, not evidence that a trained model has already been validated.

4.5 Database design

A relational schema may include Users, Complaints, Categories, Departments, ComplaintStatusHistory and Feedback. The Complaints table should use a unique complaint identifier and store submission and update timestamps. Status history should preserve prior states rather than overwrite the full record without a trace. Department mappings should be configurable because responsibilities may differ by service and jurisdiction. The final schema should match the actual prototype rather than the conceptual design shown here.

4.6 Privacy, security and responsible use

Complaint descriptions can contain names, phone numbers, addresses or sensitive circumstances. The prototype should collect only necessary information, restrict access by

role, protect credentials, validate inputs and use secure transport in any deployed environment.

A retention policy and backup procedure should be defined before real data are used. AI suggestions should be logged separately from official decisions so that a model error does not silently change the record. Citizens should have a clear route to request correction or seek help from a person.

5. System Architecture

The system can be organised into five logical layers. The presentation layer provides forms, tracking pages and the dashboard. The Flask application layer validates requests and manages the complaint lifecycle. The AI assistance layer is responsible for text cleaning and any tested recommendations. The SQL layer stores complaint data and status history. The review and monitoring layer supports authorised human decisions, audit trails and evaluation.

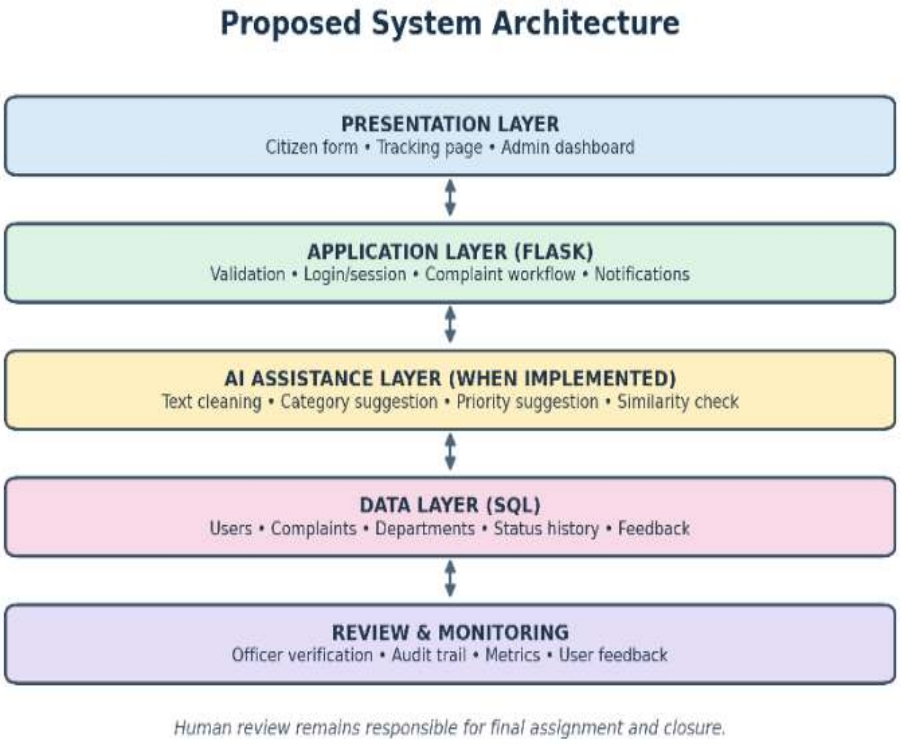


Figure 3. Logical architecture of the proposed system. The AI layer is optional until the required data and tests are available.

A typical request begins in the browser and reaches the Flask backend. The backend checks the input, saves a record and generates a tracking reference. If the AI module is enabled, it returns suggestions to the review interface. The authorised officer confirms or changes them.

The application then stores the decision and makes the status available to the citizen. For a public deployment, the design would also need secure authentication, monitoring, tested notifications, backups and an incident-response process.

6. Implementation Status

According to the project information provided by the team, a website/prototype has been developed using Python, Flask, HTML, CSS, JavaScript and SQL. The exact pages, backend

operations and AI modules have not been independently inspected for this paper. The final submission should therefore distinguish features demonstrated in the prototype from features that remain planned. The paper does not claim that AI classification, duplicate detection, bilingual NLP, SMS/email notifications or model evaluation are operational unless the team can show working evidence.

Item	Status that can currently be stated	Evidence to add before final submission
Website / prototype	Reported by the team as developed.	Screenshots, running demo or code walkthrough.
Complaint submission and tracking	Intended core workflow; verify actual working behaviour.	Test cases showing successful submission and status lookup.
SQL storage	Technology specified by the team; verify schema and persistence.	Schema screenshot and example test reord with personal data removed.
AI classification / priority	Proposed; no validated results supplied.	Dataset description, training code, test protocol and metrics.
Duplicate detection / summaries	Proposed feature unless working demo is available.	Examples, acceptance criteria and error review.
Notifications	Requires external configuration and tests.	Delivery logs or screenshots from a safe test environment.

7. Evaluation Plan

A credible evaluation needs a documented dataset and a test set that was not used during training or model selection. The paper should report the number of records, category distribution, languages represented, preprocessing steps, train/test split and model configuration. Accuracy can be included, but it should not be the only metric when some categories are much more common than others.

Measure	What it tells the reader
Accuracy	Share of test complaints assigned to the correct
Precision	Among records predicted as a category, how many actually belong to it.
Recall	Among records belonging to a category, how many the model finds.
Macro-F1	Average F1 across categories, giving each category equal weight.
Confusion matrix	Shows which categories the model most often confuses.
Review rate / uncertainty	How often the system should defer a prediction to a human reviewer.
Duplicate-alert precision	How often flagged pairs are genuinely duplicates, based on reviewed examples.

For a fair baseline, the team can compare a simple rule-based or keyword method with TF-IDF plus a classifier. All approaches should use the same held-out test data. Language-specific performance should be reported separately when there are enough examples. A small user test can also examine whether citizens understand the form and whether administrators can find and update records without confusion. No numerical evaluation results are included in this draft because no final dataset or test output has been supplied.

8. Results and Discussion

At the current stage, the supportable outcome is a proposed system design and a website/ prototype reported by the team. There is not enough evidence to report a classification accuracy, reduction in processing time or improvement in citizen satisfaction. Such claims should only be added after a repeatable evaluation has been completed.

The proposed design offers a clear sequence from submission to review and tracking. Its potential value lies in organising information and making possible AI suggestions visible to an authorised person. The design also acknowledges that existing government services already provide important grievance functions. As a result, the project should be assessed on the quality of its own prototype, the clarity of its workflow, the reliability of its data handling and the measured performance of any implemented AI features—not on feature lists alone.

An important operational risk is that a complaint may describe a time-sensitive or sensitive situation. An automated score must not block access to official emergency services or prevent an officer from escalating a case. Similarly, a duplicate detector should flag possible matches for review instead of automatically deleting or merging records. These safeguards are necessary because two citizens may report the same continuing problem from different locations or at different times.

9. Limitations

The final dataset and category taxonomy have not yet been documented in this paper.

No measured classification results or end-to-end test outcomes were supplied.

Hindi, English and mixed-language complaints may require different processing methods and separate evaluation.

Department mappings need to be checked against the correct service area and current administrative responsibilities.

Priority suggestions may be biased if training labels or historical outcomes are inconsistent.

Similarity detection can incorrectly flag separate complaints about the same recurring issue.

SMS/email delivery depends on external configuration and has not been established by the technology list alone.

The prototype is an academic project and is not an official government portal or validated emergency service.

10. Future Scope

Find and document a lawful case-level dataset with complaint text and reliable labels.

Build a reproducible TF-IDF baseline and compare it with suitable multilingual models.

Test Hindi, English and code-mixed complaints separately, including low-confidence examples.

Add a clear category-to-department mapping with version history and human approval.

Evaluate duplicate detection on reviewed complaint pairs and tune thresholds to reduce harmful false matches.

Add secure notifications, complaint history, exportable reports and dashboard analytics after testing.
Conduct usability testing with participants and document what they found difficult.

Explore official integration only through authorised, documented mechanisms and with appropriate approval.

Publish a model card or short model note describing data, intended use, limitations and evaluation.

CONCLUSION

This paper presents the design of an AI-Based Smart Government Grievance System intended to support complaint submission, record management, department review and status tracking.

It proposes optional AI assistance for complaint categorisation, priority recommendation, similarity checking and summarisation. The design uses a conventional web stack and keeps human review central to decisions that affect complaint handling.

The work is at a prototype and design stage based on the information available from the team.

Official aggregate grievance statistics can provide context, but they do not by themselves supply the text labels required for supervised complaint classification. A suitable dataset, documented implementation evidence and a repeatable evaluation are therefore the next essential steps. Until those steps are completed, the system should be presented as a prototype with planned AI features rather than as a validated operational service.

REFERENCES

1. Department of Administrative Reforms and Public Grievances, Government of India. “Centralised Public Grievance Redress and Monitoring System (CPGRAMS).” Official portal. https://pgportal.gov.in/Home/
2. National Government Services Portal, Government of India. “Check status of complaint lodged on Portal for Public Grievances (CPGRAMS).” https://services.india.gov.in/service/detail/check-status-of-complaint-lodged-on-portal-for-public-grievances-cpgrams
3. National Government Services Portal, Government of India. “Lodge your grievance against Central and State Government organisations.” https://services.india.gov.in/service/detail/lodge-your-grievance-against-central-and-state-government-organisations
4. Open Government Data (OGD) Platform India. “Year-wise Total Number of Grievances Received and Redressed on CPGRAMS Portal from 2020 to 2024.” Published 20 February 2025. https://www.data.gov.in/resource/year-wise-total-number-grievances-received-redressed-centralised-public-grievance
5. Open Government Data (OGD) Platform India. “CPGRAMS datasets and related public grievance resources.” https://www.data.gov.in/keywords/CPGRAMS
6. Press Information Bureau, Ministry of Personnel, Public Grievances and Pensions, Government of India. “Parliament Question: Enhancing Complaint Redressal through CPGRAM.” 11 December 2024. https://www.pib.gov.in/PressReleaseIframePage.aspx?PRID=2083263
7. Ministry of Electronics and Information Technology, Government of India. “Grievance Redressal.” https://www.meity.gov.in/connect/grievance-redressal
8. Pedregosa, F., et al. “Scikit-learn: Machine Learning in Python.” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. https://jmlr.org/papers/v12/pedregosa11a.html
9. scikit-learn developers. “TfidfTransformer — Feature extraction and text processing documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html
10. Salton, G., and C. Buckley. “Term-weighting approaches in automatic text retrieval.” Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988. https://doi.org/10.1016/0306-4573(88)90021-0
11. Manning, C. D., P. Raghavan, and H. Schütze. Introduction to Information Retrieval. Cambridge University Press, 2008. https://nlp.stanford.edu/IR-book/
12. Breiman, L. “Random Forests.” Machine Learning, vol. 45, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324

13. Flask documentation. “Flask Documentation.” https://flask.palletsprojects.com/
14. OWASP Foundation. “Application Security Verification Standard.” https://owasp.org/www-project-application-security-verification-standard/
15. National Institute of Standards and Technology. “Artificial Intelligence Risk Management Framework (AI RMF 1.0).” NIST AI 100-1, 2023. https://doi.org/10.6028/NIST.AI.100-1
16. Wibowo, G. D. “Artificial Intelligence-Based Classification of Public Complaints to Enhance Public Service Efficiency.” ESENSI: Jurnal Manajemen Bisnis. https://doi.org/10.56943/esensi.v27i1.32
17. “Smart Complaint Routing Systems for Public Issues Using Multimodal Artificial Intelligence.” IEEE International Conference for Convergence in Computing Technology, 2026. https://doi.org/10.1109/I3CTCON68242.2026.11508062

HOW TO CITE: Aastha Baderiya*, Monika Rajput, Chetan Choudhary, Pawan Shah, Dr. Angad Dixit, AI-Based Smart Government Grievance System, Int. J. Sci. R. Tech., 2026, 3 (11), 68-77. https://doi.org/10.5281/zenodo.23913263