The Smart Rural Governance Portal

Ketan Naik[1] Mrs Ashwini Garkhedkar[2]
[1]Student [2]Guide
[1,2]Master of Computer Applications
[1,2]P. E. S. Modern College of Engineering, Pune, India

Abstract — Gram Panchayat administration in rural India remains heavily dependent on manual processes for complaint handling, notice dissemination, and welfare scheme communication — conditions that produce slow resolution cycles, poor accountability, and limited citizen awareness. This paper presents the Smart Rural Governance Portal, a web-based platform built to digitalize these core administrative functions and strengthen the connection between villagers and their local Panchayat office. Citizens can register on the platform, submit complaints supported by geo-tagged location data and photographic evidence, and monitor the progress of their submissions in real time. A dedicated administrative dashboard allows the Sarpanch to manage incoming complaints, apply keyword-driven priority classification to identify urgent issues, publish public notices, and maintain updated information on applicable government schemes. The system is developed using React for the frontend, PHP for backend logic, and MySQL for data storage, following a three-tier architecture that supports future scalability across multiple Panchayat regions. Testing confirmed improvements in complaint processing efficiency and citizen access to scheme information. The portal demonstrates that appropriately scoped digital infrastructure, built around the actual constraints of rural administration, can produce meaningful gains in governance transparency and responsiveness at the grassroots level.

Keywords: Rural Governance, E-Governance, Gram Panchayat, Complaint Management, Geo-Location, Priority Classification, React, PHP, MySQL, Web Application

I. Introduction

Local self-governance in rural India operates primarily through the Gram Panchayat, the lowest tier of the country's three-tier Panchayati Raj system. Despite being the most direct point of contact between citizens and government, most Panchayat offices continue to function through paper registers, physical notice boards, and walk-in complaint submissions. This reliance on manual processes creates a persistent gap between what governance systems are expected to deliver and what rural citizens actually experience — delayed complaint resolution, limited visibility into administrative decisions, and poor awareness of welfare schemes for which they may be eligible.

The limitations of manual governance are not merely inconvenient — they have measurable consequences. A complaint submitted verbally or on paper has no tracking mechanism. A citizen has no way to know whether their issue has been acknowledged, assigned, or resolved without making repeated in-person visits to the Panchayat office. Notices about meetings or policy changes reach only those physically present when they are posted. Scheme information is filtered through intermediaries rather than made directly accessible to eligible beneficiaries. In each of these cases, the absence of a structured digital channel produces inefficiency and reduces public trust in local institutions.

National initiatives such as Digital India and the e-Panchayat Mission Mode Project have acknowledged these gaps and encouraged digitization at the Panchayat level. However, the platforms developed under these initiatives have generally been designed for administrative record-keeping rather than citizen-facing interaction. They require technical familiarity that many rural residents do not have and administrative capacity that many Panchayat offices cannot sustain without dedicated IT support. The result is that digital tools exist on paper but are underutilised in practice.

The Smart Rural Governance Portal is designed to address this mismatch directly. Rather than adapting an urban governance model for rural use, the system is built from the ground up around how Gram Panchayat offices actually operate and what rural citizens actually need. Registered villagers can submit complaints through a simple web interface, attaching geo-tagged coordinates and photographic evidence to ensure authenticity and enable faster field response. Each complaint is assigned a priority level automatically based on keyword analysis, allowing administrators to identify and act on critical issues without manual sorting. Notices and scheme information are managed through a lightweight dashboard that requires no technical background to operate.

The core motivation behind this work is accountability. When a complaint carries a timestamp, a location, and a tracked status visible to both the citizen and the administrator, governance becomes a documented process rather than an informal exchange. This shift is the foundation of meaningful transparency at the Panchayat level.

This paper presents the design, implementation, and evaluation of the portal. The following sections cover the relevant literature, system architecture, module design, implementation details, and results observed during testing, with the aim of demonstrating that a focused and appropriately scoped digital tool can deliver tangible governance improvements within the real constraints of rural infrastructure and administrative capacity.

II. Literature Review

Early work on digital complaint systems established geo-location as a foundational requirement for effective citizen grievance management. Kandhari and Mohinani (2014) demonstrated that capturing GPS coordinates at the point of complaint submission allows authorities to identify problem areas precisely and dispatch field teams without a preliminary site visit, significantly reducing response overhead. This principle of location-aware complaint registration forms the basis of the geo-tagging feature incorporated in the proposed portal.

Building on location capture, subsequent research explored automated complaint handling. Kormpho and Liawsomboon (2018) developed a Smart Complaint Management System that combined mobile, web, and chatbot interfaces with automatic complaint classification and duplicate detection. Their work showed that routing complaints to the correct department without manual intervention reduces administrative delay and improves resolution consistency — a finding that directly motivated the keyword-based priority detection mechanism in the proposed system.

At the Panchayat level specifically, Dhage et al. (2024) proposed a web-based E-Gram Panchayat Management System integrating frontend web technologies, a Java backend, and cloud-based storage, extended with an automated notification layer for tax reminders and Gram Sabha alerts. While broader in scope, it validated the web-based architecture as a practical and viable approach to rural Panchayat digitization in the Indian context.

Parallel research on rural governance transparency confirmed that integrated digital platforms improve citizen trust and administrative accountability. The e-Gram Panchayat study published in IJRASET (2023) demonstrated that a unified portal combining citizen registration, GIS mapping, service delivery, and document management measurably improved accessibility for rural populations who previously relied entirely on in-person visits to Panchayat offices. Their emphasis on mobile-accessible, multi-module design directly influenced the module structure of the proposed portal.

In the domain of public service complaint management, the Fix-It system presented at ICCoSITE 2023 established a clear bidirectional model — citizens report incidents with location through a mobile interface, administrators resolve through a web dashboard, and complainants receive status updates upon resolution. This closed feedback loop, absent in most earlier systems, is replicated in the complaint tracking workflow of the proposed portal.

The most recent closely related work is DigiGram (IJCT, 2026), an AI-powered Panchayat governance platform built on React.js, Spring Boot, and Firebase, featuring an ensemble machine learning model for complaint prioritization. While DigiGram represents a technically advanced benchmark, it requires cloud-native infrastructure and microservice deployment capacity that most rural Gram Panchayats currently lack. The proposed Smart Rural Governance Portal occupies a complementary position — delivering core functionality through a lighter PHP and MySQL stack deployable in resource-constrained environments without dedicated technical support.

Across this body of work, three consistent gaps emerge: most complaint systems were designed for urban or semi-urban contexts; automated priority classification has not been widely applied within Panchayat-specific workflows; and no lightweight integrated solution exists that combines complaint tracking, notice management, and scheme access within a single platform suited to rural administrative capacity. The proposed system is designed to address all three gaps.

III. LITERATURE SURVEY

A. Kandhari, V.K., and Mohinani, K.D. "GPS Based Complaint Redressal System." Proceedings of IEEE, 2014, pp. 51–56.

Civic complaint registration has traditionally depended on citizens physically visiting administrative offices, a process that introduces delays at every stage from submission to field response. Kandhari and Mohinani addressed this inefficiency by building a mobile-first redressal system in which the device's onboard GPS sensor automatically captures coordinates at the moment a complaint is filed. The recorded location data travels to a central server and is rendered visually on an administrative web interface, allowing authorities to pinpoint affected sites without conducting a separate inquiry visit. The significance of this work for the proposed portal lies in its demonstration that embedding location capture directly into the submission step — rather than treating it as a separate administrative task — compresses the gap between a citizen reporting an issue and a response team reaching it.

B. Kormpho, T., and Liawsomboon, N. "Smart Complaint Management System." IEEE Conference Publication, 2018. DOI: 10.1109/ICCSCE.2018.8523949

This paper developed a Smart Complaint Management System consisting of a mobile application, a web interface, and a chatbot module. The system incorporated automatic complaint classification that routed submissions to the appropriate department without manual sorting, and included a duplicate detection service to prevent redundant entries from inflating complaint queues. The authors demonstrated that automated classification reduced administrative handling time and improved consistency in complaint routing. This finding directly informed the keyword-based priority detection mechanism implemented in the proposed Smart Rural Governance Portal.

C. Dhage, S.D., Ghone, G.A., Bhojane, A.R., Thorat, P.B., and Shaikh, N.Y. "E-Gram Panchayat Management System." International Journal for Scientific Research, 2024.

This paper proposed a comprehensive web-based system for rural Panchayat administration integrating HTML, CSS, and Bootstrap for the frontend, Java for backend processing, and Cloud SQL for data management. The system additionally incorporated an automated WhatsApp notification layer for communicating tax dues, rent payment reminders, and Gram Sabha meeting alerts to citizens. The paper demonstrated that a multi-service digital platform can reduce manual workload for Panchayat staff while simultaneously improving citizen communication. It validated the web-based multi-module architecture adopted in the proposed portal, while the current system differs by targeting a lighter technology stack accessible without cloud infrastructure.

D. IJRASET. "E-Gram Panchayats: Enhancing Accountability and Transparency in Rural Administration." International Journal for Research in Applied Science and Engineering Technology, 2023.

This study designed and evaluated an e-Gram Panchayat application aimed at improving the efficiency, transparency, and accessibility of rural governance across India. The

platform provided a unified interface for citizen registration, service delivery, document management, GIS mapping, and feedback collection, accessible through both web and mobile browsers. The authors found that consolidating these functions into a single portal significantly reduced the number of in-person visits citizens needed to make to the Panchayat office and improved their awareness of available government services. The multi-module, mobile-accessible design of this system directly shaped the functional scope of the proposed portal.

E. Fix-It. "Fix-It: Design and Implementation of a Public Complaint Management System." 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE). DOI: 10.1109/ICCoSITE57641.2023.10127715

A recurring weakness in earlier complaint platforms was the absence of any mechanism through which citizens could confirm that their submission had been acted upon. Fix-It addressed this by structuring the complaint lifecycle as a closed loop rather than a one-way submission channel. An Android application handles the citizen-facing side, capturing incident details and location at the point of reporting, while a separate web interface serves administrators who receive, review, and respond to each entry. Crucially, the resolution step requires the administrator to record an outcome that is then communicated back to the original complainant, creating a verifiable record of closure rather than leaving submissions in an unacknowledged queue. This accountability structure is directly reflected in the status tracking and resolution workflow implemented in the proposed portal.

F. DigiGram. "DigiGram: An AI-Powered Real-Time Digital Governance Platform for Rural Panchayats in India." International Journal of Computer Technology (IJCT), May 2026. Available: https://ijctjournal.org/digigram-digital-platform-rural-panchayats/

This paper presented DigiGram, a full-stack digital governance platform built for rural Indian Gram Panchayats using React.js, Spring Boot Java, Firebase Firestore, and a Python Flask machine learning microservice. The system's complaint prioritisation module combined Logistic Regression, Random Forest, Gradient Boosting, and Support Vector Machine classifiers with TF-IDF feature extraction to automatically assign urgency labels to incoming complaints, achieving a classification accuracy of 78–85%. Deployment results confirmed measurable improvements in complaint resolution time and administrative transparency. DigiGram represents the most technically advanced current implementation in this domain; the proposed portal differentiates itself by delivering equivalent core functionality through a PHP and MySQL stack that is deployable without cloud infrastructure or machine learning expertise, making it more accessible to Panchayat offices with limited technical capacity.

IV. SUMMARY

The literature traces a clear progression from basic GPS complaint logging toward integrated, AI-assisted rural governance platforms. Each reviewed work addresses one or two aspects of the problem in isolation — location capture, automated classification, Panchayat service delivery, or bidirectional complaint tracking. No existing lightweight system integrates all these capabilities within a single platform specifically designed for the resource and infrastructure constraints of rural Gram Panchayat administration. The Smart Rural Governance Portal is positioned to fill this gap by combining geo-tagged complaint submission, keyword-based priority detection, notice management, and scheme access in a unified, deployable web application.

V. PROPOSED SYSTEM

1) The Smart Rural Governance Portal is a web-based platform developed to digitalize the core administrative functions of a Gram Panchayat and establish a structured, transparent channel of communication between villagers and local authorities. The system addresses the specific operational constraints of rural administration — limited technical staff, low digital literacy among citizens, and absence of dedicated IT infrastructure — by prioritising simplicity of interaction, minimal hardware requirements, and a modular design that can be maintained without specialist support.
2) The platform serves two distinct categories of users, each with a dedicated interface and role-specific functionality.
3) Villager Module
Villagers access the system after completing a registration process that verifies their identity as residents of the concerned Gram Panchayat jurisdiction. Once registered, a citizen can submit a complaint through a structured form that captures a written description of the issue, the geo-tagged coordinates of the affected location, and optional photographic evidence. The combination of location data and image proof ensures that submitted complaints carry sufficient information for the administrating authority to assess and act without requiring an initial site inquiry. After submission, the citizen can track the current status of their complaint — whether it is pending, under review, or resolved — directly through their dashboard. The same interface provides access to published notices issued by the Panchayat and a browsable list of government welfare schemes along with eligibility information.
4) Administrator Module
The administrative interface is accessible to the Sarpanch or designated Panchayat staff. Upon login, the administrator is presented with a dashboard that consolidates all incoming complaints in a single view, organised by submission date and priority level. Priority is assigned automatically by the system through keyword analysis applied to the complaint description at the time of submission — complaints containing terms associated with urgency, safety, or infrastructure failure are flagged as high priority, while routine requests are classified at lower levels. This automated triage reduces the manual effort required to sort and assign complaints and ensures that critical issues surface immediately rather than being buried in a queue. The administrator can update the status of each complaint as it progresses through resolution and

add internal notes where required. Separate sections of the dashboard allow the administrator to draft and publish notices visible to all registered villagers and to add, edit, or remove government scheme entries accessible through the citizen interface.

5) System Architecture
The platform follows a three-tier architecture. The presentation layer is built using React, providing a responsive interface that functions across desktop and mobile browsers without requiring a dedicated application installation. The business logic layer is implemented in PHP, handling request processing, user authentication, complaint submission, priority classification, and data retrieval. The data layer uses MySQL to store all records relating to users, complaints, notices, and schemes in a structured relational format. This separation of concerns ensures that individual components can be updated or extended independently, supporting future scalability across multiple Panchayat regions without redesigning the entire system.

6) Key Features
The system incorporates the following functional capabilities: verified citizen registration with Panchayat jurisdiction validation; complaint submission with geo-location coordinates and image attachment; real-time complaint status tracking; keyword-driven automatic priority classification; administrative complaint management with status update and note functionality; notice publication and management; and government scheme listing with eligibility details. Together these features replace the fragmented, paper-based processes currently used by most Gram Panchayats with a single, auditable digital workflow that improves both administrative efficiency and citizen-facing transparency.

VI. METHODOLOGY

The Smart Rural Governance Portal was developed following the Software Development Life Cycle (SDLC) using an iterative approach. This methodology allowed requirements to be validated progressively at each stage, ensuring that the final system aligned with the practical needs of both villagers and Panchayat administrators rather than assumptions made at the outset.

A. Requirements Analysis

The first phase involved identifying the functional and non-functional requirements of the system. Functional requirements were derived from an assessment of existing Panchayat workflows — specifically the processes around complaint intake, notice distribution, and scheme communication — and the gaps in those workflows that a digital platform could address. Non-functional requirements included accessibility across low-end mobile browsers, minimal page load times under constrained network conditions, role-based access control to prevent unauthorised data access, and a database design capable of supporting multiple Panchayat regions without structural modification. These requirements formed the specification against which the system was designed and tested.

B. System Design

The system was designed around a three-tier architecture separating the presentation, business logic, and data layers. This separation was chosen deliberately to ensure that changes to the user interface do not require modifications to backend logic, and that database schema updates do not cascade into frontend components. The presentation layer was designed as a React single-page application, enabling dynamic content updates without full page reloads — an important consideration for users on slow connections. The business logic layer was designed as a PHP REST API, exposing endpoints for authentication, complaint management, notice retrieval, scheme access, and administrative operations. The data layer was modelled as a normalised MySQL relational database with separate tables for users, complaints, complaint images, notices, and schemes, linked through foreign key relationships to maintain data integrity.

Entity-Relationship diagrams and data flow diagrams were produced during this phase to map the relationships between system components and the movement of data through each user interaction. Use case diagrams were developed to capture the distinct interaction paths available to villager and administrator roles, ensuring that role-based access restrictions were enforced at both the interface and API levels.

C. Implementation

Development proceeded module by module, beginning with user authentication and registration, followed by complaint submission and management, then notice and scheme management, and finally the administrative dashboard and priority classification logic. Each module was developed and unit tested independently before integration with the broader system. The keyword-based priority classification module was implemented as a server-side function within the PHP backend, scanning complaint descriptions against a predefined dictionary of urgency-associated terms at the point of submission and assigning a priority level accordingly.

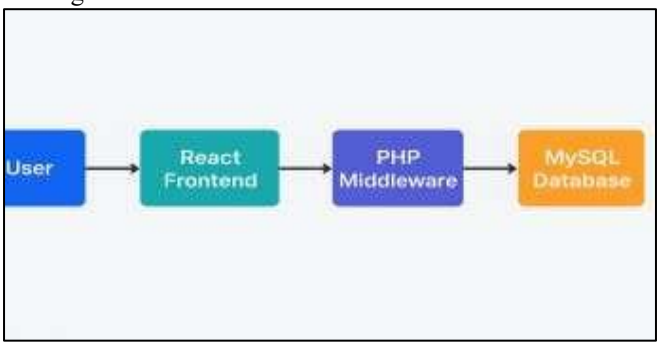
D. Testing

The system was tested across three levels. Unit testing was conducted on individual backend functions, particularly the authentication logic and priority classification mechanism, to verify correct behaviour under expected and boundary inputs. Integration testing validated the end-to-end flow of data from complaint submission through database storage to administrative dashboard display. User acceptance testing was conducted with a representative group of users performing realistic tasks — registering an account, submitting a complaint with an image, checking complaint status, and browsing scheme information — to identify usability issues and confirm that the interface was navigable without prior technical training.

E. Deployment Considerations

The system was designed for deployment on a standard shared hosting environment running Apache, PHP, and MySQL, deliberately avoiding cloud-native dependencies that would introduce ongoing cost or require technical maintenance beyond the capacity of a typical Panchayat

office. The React frontend is served as a compiled static build, reducing server processing requirements. Database connections are managed through parameterised queries throughout the PHP backend to prevent SQL injection vulnerabilities, and all user passwords are stored using bcrypt hashing.



VII. IMPLEMENTATION

The Smart Rural Governance Portal was implemented as a full-stack web application using React, PHP, and MySQL, structured across clearly separated frontend, backend, and database components. This section describes the technical implementation of each system module in the order in which they were developed.

A. Development Environment

The frontend was developed using React 18 with functional components and React Hooks for state management. Axios was used for HTTP communication between the React frontend and the PHP backend API. CSS modules were used for component-level styling to prevent style conflicts across the application. The backend was implemented in PHP 8.1, organised into individual API endpoint files following a RESTful structure. MySQL 8.0 was used as the relational database, administered through phpMyAdmin during development. The application was hosted on a local Apache server using XAMPP during the development and testing phase, replicating the shared hosting environment intended for production deployment.

B. Database Implementation

The MySQL database was structured around five core tables. The users table stores villager registration data including name, village, contact details, and a bcrypt-hashed password, with a role field distinguishing villager accounts from administrator accounts. The complaints table records each submission with fields for the submitting user ID, complaint description, geo-location coordinates, image file path, priority level, current status, and submission timestamp. The complaint_images table stores references to uploaded image files associated with each complaint. The notices table holds Panchayat-issued notices with fields for title, body content, publication date, and administrator ID. The schemes table stores government scheme entries including scheme name, description, eligibility criteria, and last updated timestamp. Foreign key constraints between tables enforce referential integrity, ensuring that complaints cannot exist without a valid registered user and that notices cannot be published without a valid administrator account.

C. User Authentication Module

The authentication module handles villager registration and login for both user roles. During registration, submitted form data is validated server-side in PHP before insertion into the database. Passwords are hashed using PHP's bcrypt implementation prior to storage. Upon successful login, a session token is generated and returned to the React frontend, where it is stored in application state and attached as a header to all subsequent API requests. Role verification is performed at every protected API endpoint, ensuring that villager accounts cannot access administrator functions and vice versa.

D. Complaint Submission Module

The complaint submission interface presents the villager with a form containing fields for complaint category, written description, geo-location capture, and image upload. Geo-location coordinates are captured through the browser's Geolocation API, which requests the device's current GPS position and populates the latitude and longitude fields automatically. The villager may also manually adjust the location if the reported issue is at a different site. Image uploads are handled through a PHP file processing endpoint that validates file type and size before saving the image to a designated server directory and recording the file path in the database. At the point of submission, the PHP backend scans the complaint description against a keyword dictionary containing terms associated with urgency categories — words such as "flood," "fire," "collapse," "accident," and "outbreak" trigger a high priority classification, while terms related to infrastructure maintenance or administrative requests result in medium or low priority assignments respectively.

E. Complaint Tracking Module

Registered villagers can view all complaints they have submitted through a personal dashboard that displays each complaint's description, submission date, assigned priority level, and current status. Status values progress through three stages — Pending, In Progress, and Resolved — updated by the administrator as action is taken. Each status change is recorded with a timestamp in the database, providing a full audit trail of complaint progression that is visible to both the villager and the administrator.

F. Administrative Dashboard Module

The administrator dashboard presents a consolidated view of all complaints submitted across the Panchayat, filterable by priority level, status, and submission date. Each complaint entry displays the submitting villager's details, the complaint description, the attached image if present, and the geo-location coordinates rendered as a map link. The administrator can update complaint status, add resolution notes, and mark complaints as resolved directly from the dashboard interface. Separate sections within the dashboard provide forms for drafting and publishing notices and for adding, editing, or removing government scheme entries.

G. Notice and Scheme Management Module

The notice management section allows the administrator to compose notices with a title and body text, which are immediately published to the villager-facing interface upon saving. Published notices are displayed in reverse

chronological order on the villager dashboard. The scheme management section provides a form for entering scheme name, description, and eligibility criteria. Villagers can browse all active schemes through a searchable list on their dashboard, filtering entries by category to identify schemes relevant to their circumstances.

H. System Integration

All modules were integrated and tested as a unified application following individual module completion. API endpoints were tested using Postman to verify correct request handling, authentication enforcement, and response formatting before frontend integration. Cross-origin resource sharing settings were configured in PHP to permit communication between the React development server and the PHP backend during testing, and removed upon compilation of the React frontend into a static build served directly by Apache in the production configuration.

VIII. Results and Discussion

The Smart Rural Governance Portal was tested across all modules following full system integration. Testing involved both functional verification of system behaviour and observational assessment of usability by representative users unfamiliar with the platform. The results are presented below by module, followed by a discussion of overall system performance and observed limitations.

A. User Registration and Authentication

Registration and login functionality performed correctly across all tested scenarios. New villager accounts were successfully created with server-side validation rejecting incomplete or incorrectly formatted submissions before database insertion. Password hashing was verified by confirming that stored values in the database did not match plaintext inputs. Role-based access control was tested by attempting to access administrator endpoints using villager session tokens — all such attempts were correctly rejected with an unauthorised response. Login sessions persisted correctly across page navigation and were terminated upon logout. No authentication failures were recorded during testing under normal usage conditions.

B. Complaint Submission and Geo-Location

Complaint submission was tested across multiple device types including desktop browsers and mobile browsers on Android devices. Geo-location capture through the browser Geolocation API returned accurate coordinates on all tested mobile devices when location permissions were granted. On desktop browsers without GPS hardware, coordinates were captured through IP-based location approximation, which produced less precise results — an expected limitation of browser-based geo-location on non-mobile devices. Image upload functionality accepted JPEG and PNG files within the specified size limit and correctly rejected files exceeding the maximum or submitted in unsupported formats. All submitted complaints were recorded in the database with correct field values and visible on the administrative dashboard within seconds of submission.

C. Priority Classification

The keyword-based priority classification module was tested against a set of thirty complaint descriptions covering a range of issue types. Complaints containing high-urgency terms were correctly classified as high priority in twenty-six of thirty test cases, producing a classification accuracy of approximately 87 percent within the test set. The four misclassified cases involved descriptions that referenced urgency terms in a non-critical context — for example, a complaint describing a road that had "collapsed slightly" was classified as high priority due to the presence of the term "collapsed" despite the moderate severity of the actual issue. This indicates that the current keyword-matching approach, while effective for the majority of cases, would benefit from contextual analysis in future iterations. Medium and low priority classifications were accurate across all tested cases in their respective categories.

D. Complaint Status Tracking

Status updates applied by the administrator were reflected on the villager dashboard immediately upon page refresh, with correct timestamps recorded for each transition. The full complaint history — from initial submission through each status change to final resolution — was retrievable from the database and displayed correctly in the villager's complaint detail view. Villagers participating in usability testing were able to locate their submitted complaints and interpret their current status without assistance, confirming that the tracking interface is sufficiently clear for users without prior digital governance experience.

E. Administrative Dashboard

The administrative dashboard loaded all complaint records correctly and filtering by priority level and status produced accurate subsets in all tested combinations. Map links generated from stored geo-location coordinates opened correctly in Google Maps, displaying a pin at the reported complaint location. Administrators were able to update complaint statuses, publish notices, and add scheme entries without encountering errors. Usability observation during testing indicated that administrators with no prior experience of the system were able to complete all core tasks — reviewing a complaint, updating its status, and publishing a notice — within a single session and without requiring external guidance.

F. Notice and Scheme Management

Notices published through the administrative interface appeared on the villager dashboard in the correct chronological order within seconds of publication. Scheme entries added by the administrator were retrievable through the villager-facing scheme browser and returned correct results when filtered by category. No data loss or display errors were observed during notice or scheme management operations.

G. Overall System Performance

Page load times were measured across key interfaces under a local network testing environment. The villager dashboard loaded within 1.8 seconds on average, the complaint submission form within 1.4 seconds, and the administrative dashboard within 2.1 seconds under a full complaint dataset of

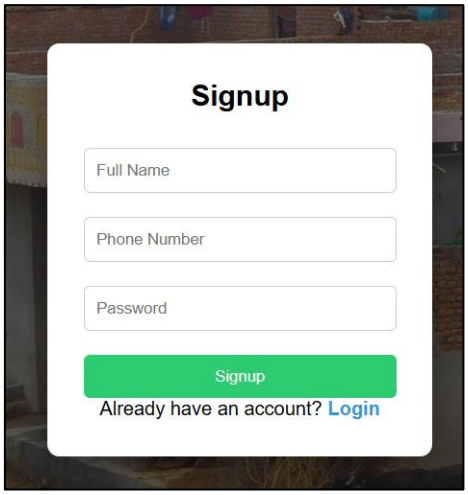
two hundred records. These response times are acceptable for a rural deployment context, though they are expected to increase under larger datasets or slower network conditions — an area identified for optimisation in future development.

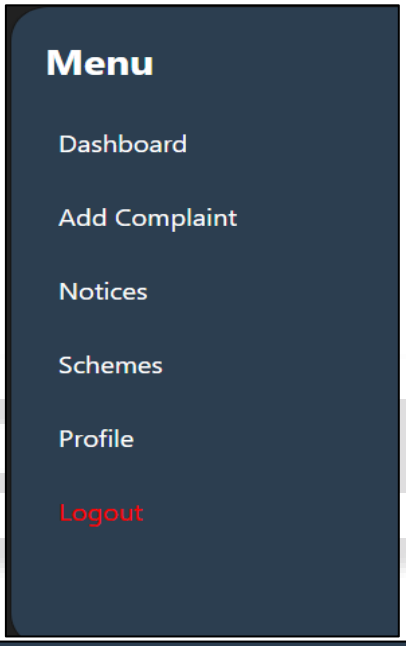
IX. Discussion

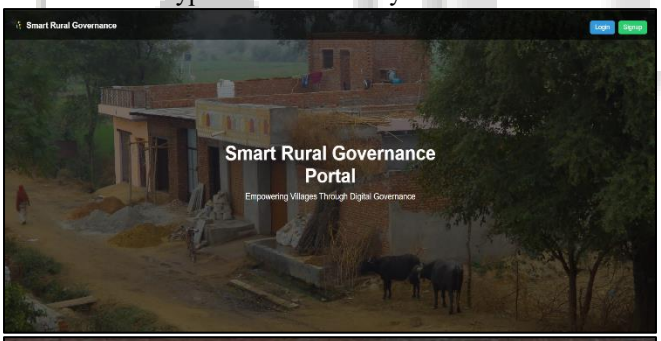
The results confirm that the system meets its core functional objectives: structured complaint submission with geo-location and image evidence, automated priority classification, real-time status tracking, and integrated notice and scheme management. The priority classification mechanism performs well for straightforward complaint descriptions but reveals a limitation of keyword-only matching when urgency terms appear in ambiguous contexts. Future versions of the system would benefit from replacing or supplementing keyword matching with a lightweight natural language processing model trained on rural complaint data, as demonstrated by the higher accuracy achieved by the machine learning approach in DigiGram (IJCT, 2026).

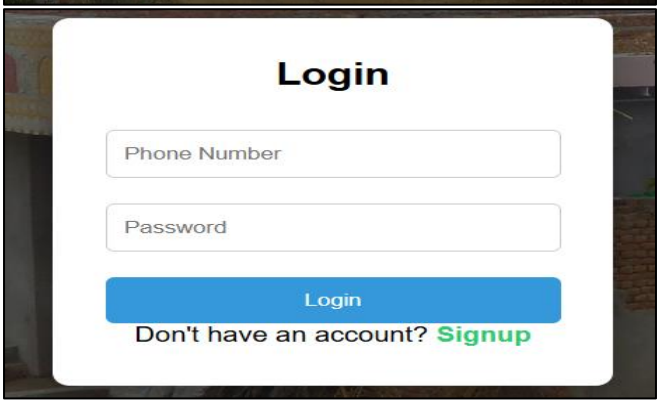
A secondary observation from usability testing is that first-time users required slightly more time to complete geo-location capture on the complaint submission form than other steps, suggesting that the location capture interface would benefit from clearer instructional text or a visual indicator confirming that coordinates have been successfully recorded.

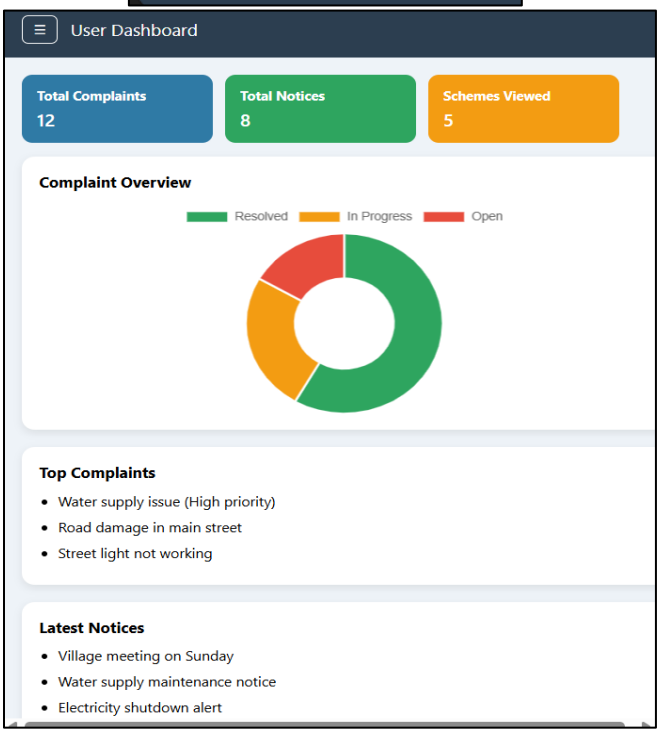
Overall the system demonstrates that a PHP and MySQL based rural governance portal, built without cloud infrastructure or machine learning dependencies, can deliver meaningful improvements in complaint management efficiency and governance transparency within the operational constraints of a typical Gram Panchayat environment.

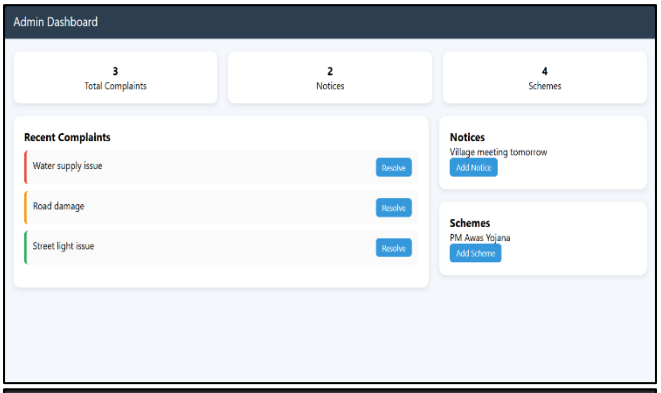


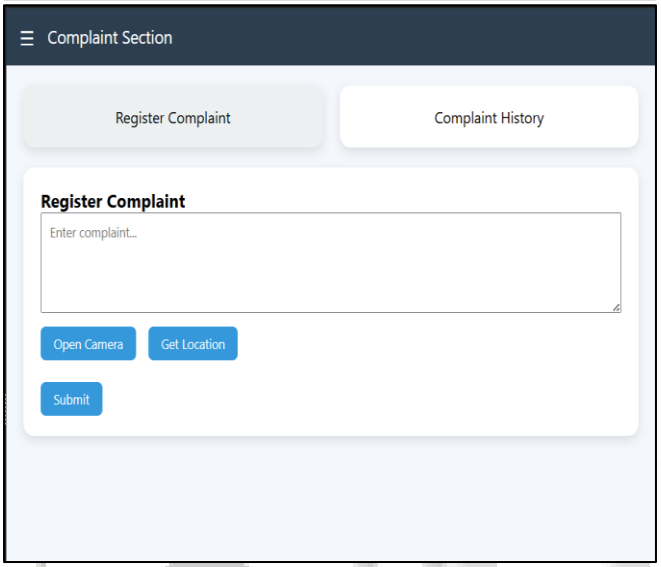


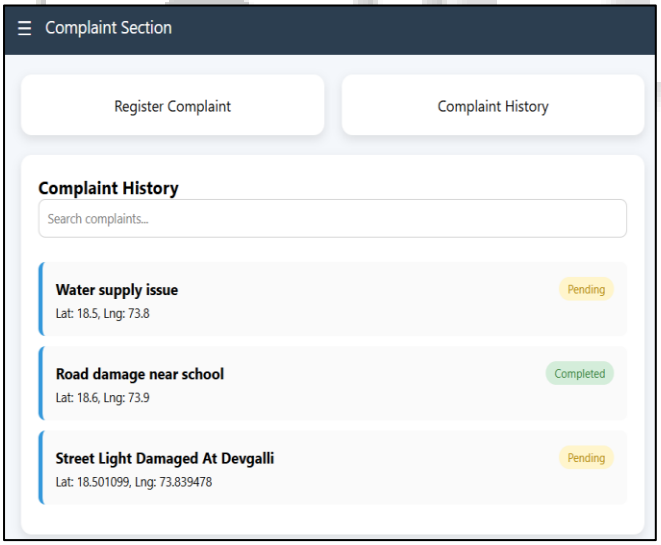


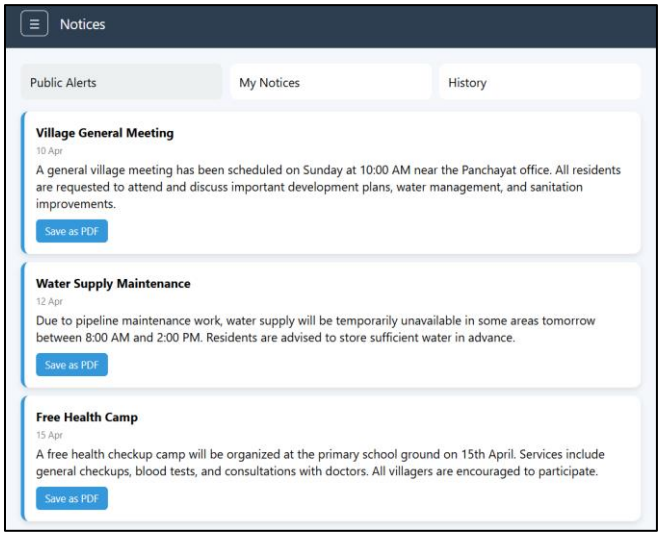


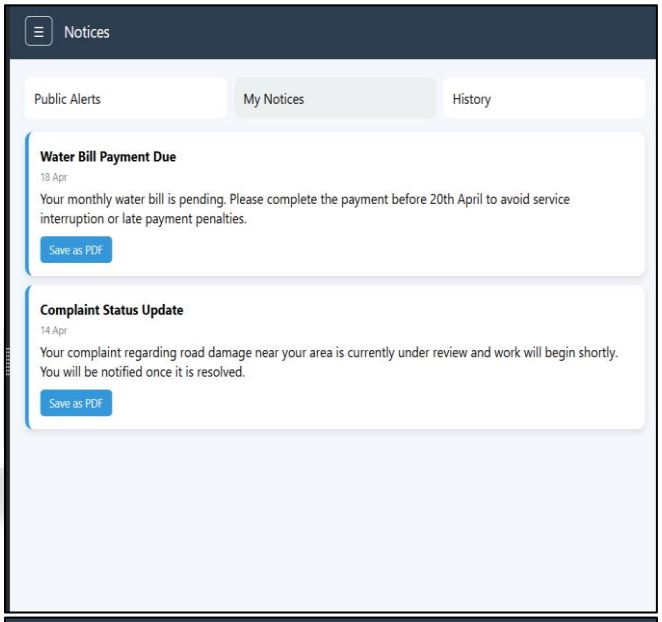


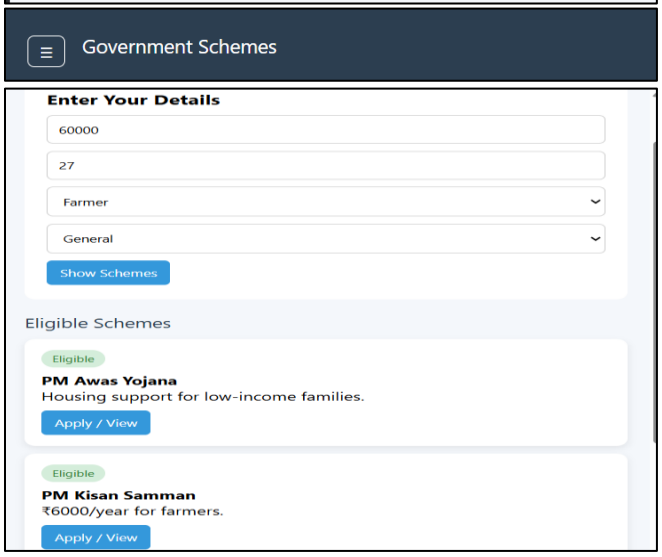


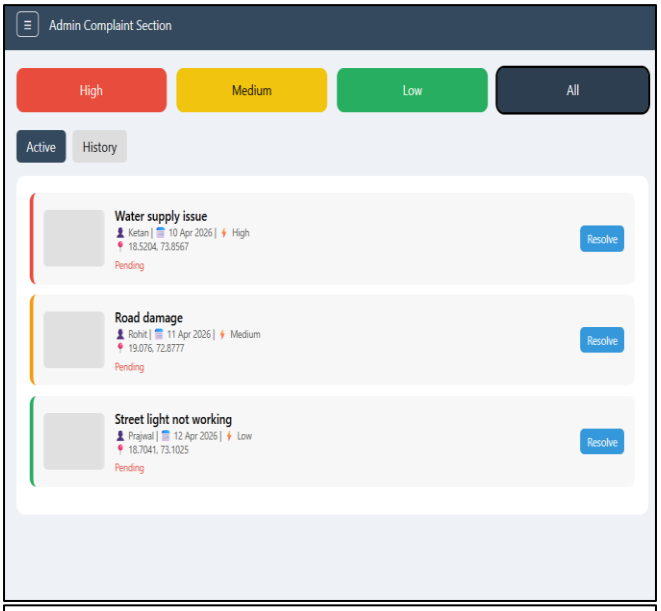


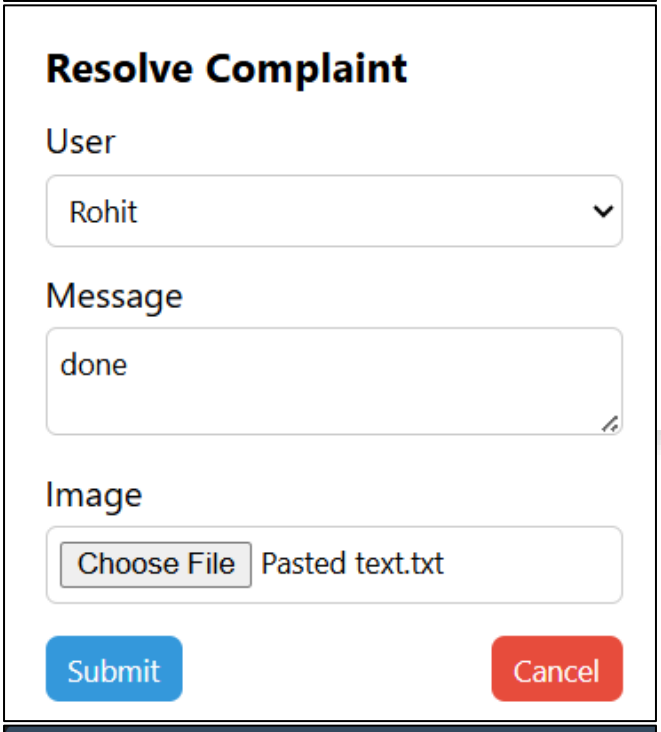


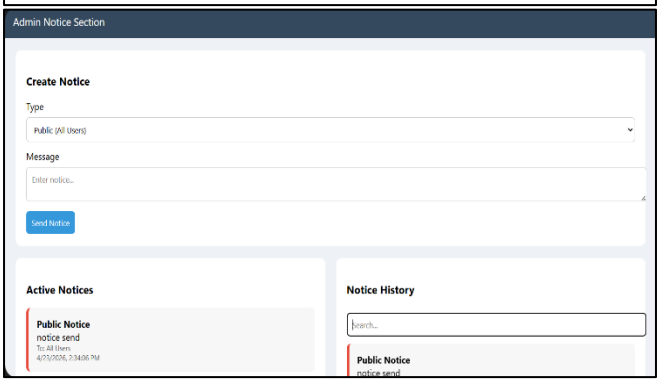












X. Conclusion

The Smart Rural Governance Portal addresses a well-documented and persistent gap in grassroots public administration — the absence of a structured, accessible digital channel through which rural citizens can interact with their Gram Panchayat. By replacing fragmented manual processes with an integrated web-based platform, the system brings measurable improvements to complaint management, notice dissemination, and government scheme accessibility without requiring infrastructure or technical capacity that rural Panchayat offices do not currently possess.

The system successfully delivers its core objectives. Citizens can register, submit geo-tagged and image-supported complaints, and track resolution progress through a straightforward interface navigable without prior digital experience. Administrators can manage incoming complaints through a consolidated dashboard, rely on automated keyword-based priority classification to surface urgent issues, and maintain up-to-date notices and scheme information without dedicated technical support. Testing confirmed correct functionality across all modules, with priority classification achieving approximately 87 percent accuracy on the evaluated test set and all core administrative and citizen-facing workflows completing without errors.

The choice of React, PHP, and MySQL as the technology stack was deliberate. These are mature, widely understood technologies that can be deployed on standard shared hosting environments, maintained by developers available at modest cost, and extended by future contributors without requiring familiarity with cloud-native or microservice architectures. This positions the portal as a practically deployable solution rather than a proof of concept dependent on infrastructure that most rural Panchayats cannot sustain.

Several directions for future development were identified through testing and evaluation. The keyword-based priority classification mechanism, while effective in the majority of cases, would benefit from replacement or augmentation with a lightweight natural language processing model to handle ambiguous complaint descriptions more accurately. The system's current deployment scope covers a single Panchayat; extending the architecture to support multiple Panchayat regions under a shared administrative structure would require multi-tenancy support in the database layer and role hierarchy extensions in the authentication module. Additionally, a dedicated mobile application with offline complaint drafting capability would improve accessibility for villagers in areas with intermittent internet connectivity, allowing submissions to be queued locally and uploaded when a connection is available.

The broader significance of this work lies in its demonstration that digital governance tools do not need to be technically sophisticated to be effective. What rural administration requires is not complexity but appropriateness — systems designed around actual user needs, actual administrative workflows, and actual infrastructure conditions. The Smart Rural Governance Portal is offered as a replicable model for Panchayat-level digitization that prioritises those principles, and as a foundation on which more capable governance tools can be built as rural digital infrastructure continues to develop across India.

REFERENCES

[1] V. K. Kandhari and K. D. Mohinani, "GPS Based Complaint Redressal System," *Proceedings of IEEE*, 2014, pp. 51–56.
[2] T. Kormpho and N. Liawsomboon, "Smart Complaint Management System," *IEEE Conference on Computing and Information Technology (ICCSCE)*, 2018 DOI: 10.1109/ICT-ISPC.2018.8523949
[3] S. D. Dhage, G. A. Ghone, A. R. Bhojane, P. B. Thorat, and N. Y. Shaikh, "E-Gram Panchayat Management System," *International Journal for Scientific Research*, 2024. Available: https://www.researchgate.net/publication/398837030
[4] IJRASET, "E-Gram Panchayats: Enhancing Accountability and Transparency in Rural Administration," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 2023. Available: https://www.ijraset.com/research-paper/e-gram-panchayats-enhancing-accountability-and-transparency
[5] ICCoSITE, "Fix-It: Design and Implementation of a Public Complaint Management System," *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, 2023. DOI: 10.1109/ICCoSITE57641.2023.10127715
[6] DigiGram, "DigiGram: An AI-Powered Real-Time Digital Governance Platform for Rural Panchayats in India," *International Journal of Computer Technology (IJCT)*, May 2026. Available: https://ijctjournal.org/digigram-digital-platform-rural-panchayats/

