International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Village Problem Reporting System

A.Kavya[1],Bommana Akshitha[2],.Domalapalli Mounika[3],Veeraboina Pooja Pranathi[4]
[1]Assistant Professor; Department Of CSE(AI & Ml) Bhoj Reddy Engineering College For Women, Hyderabad, India
[2,3,4]B.Tech Students; Department Of CSE(AI & Ml) Bhoj Reddy Engineering College For Women, Hyderabad, India.
akshithabommana21@gmail.com

Abstract
Efficient governance in rural areas requires timely identification and resolution of local issues such as road damage, water supply interruptions, electricity failures, and sanitation problems. Traditional complaint reporting methods in villages are largely manual, requiring citizens to visit government offices or submit written complaints, which often results in delays, lack of transparency, and inefficient problem management. This paper proposes a web-based Village Problem Reporting System that enables villagers to register, submit, and track complaints online through a centralized platform. The system allows administrators to manage complaints efficiently, update statuses, and analyze recurring issues for better decision-making. By providing easy access, real-time updates, and transparent communication between citizens and authorities, the proposed system enhances administrative efficiency, reduces manual work, and promotes community engagement. The implementation of such a digital platform can significantly improve rural governance and support effective management of village infrastructure and public services.

Keywords: *Village Problem Reporting System, e-governance, digital complaint management, rural development, transparency, web-based application*

Introduction

The Village Problem Reporting System is a web-based platform designed to enable villagers to report local issues directly to relevant authorities. In line with modern e-governance principles, such digital platforms enhance transparency, communication, and administrative efficiency. Through this system, citizens can submit complaints regarding road damage, water supply issues, sanitation, electricity failures, and other public services using a user-friendly web interface.

Traditionally, rural populations rely on manual complaint methods, including visiting local offices or submitting written forms, which are often slow, inefficient, and lack proper tracking mechanisms. A digital reporting system streamlines the process by allowing citizens to submit, monitor, and receive updates on complaints, significantly improving responsiveness.

The proposed system is a web application that allows villagers to register, log in, and submit complaints online. Complaints are stored in a centralized database, which administrators can access to review, update, and resolve issues. This approach enhances communication between villagers and authorities and reduces resolution time.

Benefits of the Village Problem Reporting System

Support for Rural Development

The system facilitates monitoring and maintenance of critical village infrastructure, including roads, water supply, drainage, and electricity. By tracking complaints digitally, authorities can prioritize and resolve issues more efficiently.

Support for Government Authorities

Officials can view all reported problems through a centralized dashboard, update statuses, manage multiple complaints, and maintain records for future reference.

Citizen Engagement

The platform encourages active community participation by enabling villagers to report problems and provide feedback, fostering improved communication between citizens and local governance bodies.

Easy Access to Information

Users can submit complaints from any location via a web browser and monitor the progress of their issues without visiting government offices.

Centralized Complaint Management

All complaints are recorded in a structured database, facilitating data analysis, tracking resolution times, and ensuring transparency in public service management.

Existing System

In many rural areas, complaints regarding road damage, water shortages, electricity outages, drainage issues, and sanitation are reported through traditional manual methods. Villagers typically visit local offices such as the village panchayat or municipal office, submit written forms, or communicate issues verbally.

This approach has several limitations:

- Physical visits are inconvenient and time-consuming.
- There is no efficient system for tracking complaints, leaving villagers uncertain about progress.

- Lack of centralized documentation hinders proper analysis, prioritization, and transparency.
- Communication delays between citizens and authorities result in slower problem resolution.

These challenges highlight the need for a digital solution that simplifies complaint reporting and improves administrative efficiency.

Problems in the Existing System

The current manual reporting process poses several challenges:

- **Limited Accessibility:** Villagers must physically visit offices to register complaints.
- **Delayed Resolutions:** Manual record-keeping increases the likelihood of misplaced complaints and slows response times.
- **Lack of Transparency:** Citizens rarely receive updates on the status of their issues.
- **Absence of Centralized Records:** Authorities cannot efficiently track recurring problems or prioritize urgent issues.

Overall, the traditional system is inefficient and restricts effective communication between villagers and administrators, necessitating a digital alternative.

Proposed System

The proposed Village Problem Reporting System is a web-based application that allows villagers to report local issues efficiently. Users can register, log in, and submit complaints related to infrastructure and public services.

Complaints are stored in a centralized database accessible to administrators, who can review, categorize, and update complaint statuses as pending, in progress, or resolved. The system enhances communication, reduces manual work, and ensures faster resolution of issues.

By providing transparent and structured complaint management, the platform improves citizen engagement, administrative efficiency, and overall governance of village infrastructure.

1.5 Advantages of the Proposed System

The proposed system offers several benefits over traditional manual methods:

- **Ease of Access:** Villagers can submit complaints online without visiting offices physically.
- **Centralized Management:** All complaints are stored digitally, allowing administrators to organize, track, and prioritize issues effectively.
- **Transparency:** Citizens can monitor the status of their complaints in real-time.
- **Reduced Paperwork:** Digital records minimize manual documentation, save time, and reduce errors.
- **Improved Communication:** Faster interactions between villagers and authorities accelerate issue resolution.

Methodology

Functional Requirements

Functional requirements define the core operations and services the Village Problem Reporting System must provide. The system is structured into two primary modules: **User Module** and **Admin Module**.

User Module:

This module allows villagers to report and track local issues conveniently.

- **Registration and Login:** Users must register with basic personal details and log in to access system features.
- **Complaint Submission:** Users can submit complaints regarding village infrastructure, such as damaged roads, water shortages, electricity outages, or sanitation issues. The system also supports uploading images as evidence.
- **Complaint Tracking:** Users can monitor the progress of their submitted complaints and view statuses including *pending*, *in progress*, or *resolved*.

Admin Module:

This module enables authorities to manage, prioritize, and resolve complaints.

- **Admin Login:** Administrators access the system through secure credentials.
- **Complaint Review:** All submitted complaints, along with associated details and images, are available for review.
- **Status Update:** Administrators can update the status of complaints after verification and appropriate action.

Non-Functional Requirements

Non-functional requirements specify the quality and operational standards that ensure the system runs efficiently.

1. **Performance:** The system should provide fast response times for complaint submission and status tracking, supporting multiple concurrent users without degradation.
2. **Security:** Secure login mechanisms must protect user and administrator data from unauthorized access. Complaint information should remain confidential and tamper-proof.
3. **Usability:** The interface should be intuitive, allowing villagers to easily register, submit complaints, and track their progress.
4. **Reliability:** The system must function consistently, with minimal downtime, and ensure safe storage of complaint data.
5. **Scalability:** The platform should accommodate an increasing number of users and complaints as adoption grows.
6. **Maintainability:** The system design should allow easy updates, maintenance, and integration of new features.

Computational Resource Requirements

Efficient development and deployment of the Village Problem Reporting System require specific hardware and software resources.

Hardware Requirements:

- **Processor:** Intel Core i3 or higher
- **RAM:** Minimum 4 GB
- **Storage:** At least 500 GB hard disk
- **Display:** Standard monitor
- **Input Devices:** Keyboard and Mouse
- **Internet Connectivity:** Required for accessing the web application

Software Requirements:

- **Operating System:** Windows 10 or higher
- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Python with Flask framework
- **Database:** SQLite
- **Development Environment:** Visual Studio Code
- **Web Browser:** Google Chrome or any modern browser

Software Development Life Cycle (SDLC)

The project follows a structured **Software Development Life Cycle (SDLC)** to ensure systematic design, implementation, testing, and maintenance. For this system, the **Waterfall Model** has been adopted.

In the Waterfall approach, development progresses sequentially through the following phases:

1. **Requirement Analysis:** Identification and documentation of system requirements.
2. **System Design:** Designing architecture, modules, and database structure.
3. **Implementation:** Coding the system using the chosen technologies.
4. **Testing:** Verifying the system functions as intended and fixing defects.
5. **Deployment:** Releasing the system for user access.
6. **Maintenance:** Performing updates, enhancements, and troubleshooting over time.

Each phase is completed before proceeding to the next, ensuring clear documentation and structured project development.

System Design and Architecture

System Architecture

System architecture defines the internal structure of a software application and illustrates how its components interact to achieve desired functionality. The **Village Problem Reporting System** employs a **three-tier architecture**, which separates presentation, application logic, and data management to ensure modularity, scalability, and maintainability.

Presentation Layer (Frontend): This layer provides the user interface for both villagers and administrators. Villagers can submit complaints, track their status, and upload supporting images, while administrators can view and manage complaints. The frontend is implemented using **HTML5, CSS3, and JavaScript**, providing a responsive and interactive web experience.

Application Layer (Backend): The application layer handles the core business logic of the system. It processes user requests, manages authentication, handles complaint submissions, and communicates with the database. The backend is developed using **Python with the Flask framework**, ensuring efficient processing and secure operations.

DataLayer(Database): This layer is responsible for storing and retrieving all system data, including user information, complaint details, images, and complaint statuses. **SQLite** is used as the database for its simplicity and lightweight integration with web applications.

Technical Architecture

The technical architecture outlines the technologies and their interaction to implement the system:

- **Frontend:** HTML5, CSS3, JavaScript
- **Backend:** Python (Flask framework)
- **Database:** SQLite

The frontend sends HTTP requests to the backend server, which processes the requests, interacts with the database as needed, and returns responses to the user interface. This separation ensures that data processing, storage, and presentation are modular and maintainable.

Use Case Diagram

The use case diagram represents system functionality and user interactions. The primary actors in the system are the **Villager** and the **Admin/Officer**.

- **Villager:** Can register, log in, submit complaints, upload images, and track complaint status.
- **Admin/Officer:** Can log in, view complaints, update status, and manage reported issues.

The use case diagram provides a visual understanding of system operations and helps identify functional requirements.

Class Diagram

The class diagram depicts the static structure of the system, showing classes, attributes, methods, and relationships:

- **User Class:** Contains attributes such as name, email, and password.
- **Complaint Class:** Stores complaint details, including description, location, image, and status.
- **Admin Class:** Manages complaints and updates their statuses.

This diagram clarifies how data is organized and how system components interact.

Sequence Diagram

The sequence diagram illustrates interactions between system components in a specific sequence. For complaint submission:

1. Villager logs into the system.
2. Villager submits complaint details via the interface.
3. Backend server processes the request and stores the complaint in the database.
4. System sends a confirmation response back to the user.

This diagram helps understand the order of operations and data flow within the system.

Activity Diagram

The activity diagram represents the workflow of complaint reporting:

1. User logs in.
2. Complaint details are entered.
3. Complaint is submitted and processed by the system.
4. Data is stored in the database.
5. Admin reviews and updates the complaint status.

This diagram visualizes the sequence of actions and decision points in the complaint management process.

Implementation

Technologies Used

The Village Problem Reporting System is developed using a combination of modern web technologies that support efficient design, development, and deployment of the application. HTML (HyperText Markup Language) is used to define the structure and layout of the web pages, forming the backbone of the user interface. CSS (Cascading Style Sheets) is applied to enhance the visual appearance of the system by incorporating styles, colors, and responsive layouts, making the interface more user-friendly and accessible. JavaScript is utilized to introduce interactivity and handle client-side operations, such as validating user inputs and dynamically updating content without reloading the page.

On the server side, the backend is implemented using Python with the Flask framework, which is responsible for processing user requests, managing authentication, handling complaint submissions, and communicating with the database. SQLite is used as the database management system due to its lightweight nature and ease of integration, enabling efficient storage and retrieval of user information and complaint data. The development process is carried out using Visual Studio Code, which provides a flexible and efficient environment for writing, debugging, and maintaining the application code. Together, these technologies ensure that the system is reliable, scalable, and easy to maintain.

Pseudo Code

The operational workflow of the Village Problem Reporting System can be described through a structured sequence of steps. When the system starts, it displays the main interface with options for villager login, officer login, or exiting the application. The system continuously runs in a loop, allowing users to interact with it until they choose to exit. When a villager selects the login option, the system prompts for a username and password, which are then verified against the database. If the credentials are valid, the user gains access to features such as submitting a complaint, viewing complaint status, or logging out.

During complaint submission, the user is required to enter details such as the problem category and description. The system then generates a unique complaint identifier and stores the complaint information in the database with an initial status marked as "Pending." If the user chooses to view complaint status, the system retrieves relevant records from the database and displays the current status of submitted complaints. In case of invalid login credentials, the system notifies the user and prompts them to try again. Similarly, administrators can log in to manage complaints, review submitted issues, and update their status accordingly. The system continues to operate until the user selects the exit option, at which point the process terminates.

Testing and Validation

Overview of Testing

Testing is a critical phase in the software development lifecycle that ensures the system operates correctly and satisfies user requirements. Its primary objective is to detect errors, defects, and missing functionalities before the system is deployed in a real-world environment. In the Village Problem Reporting System, testing is carried out to validate the proper functioning of all major modules, including user registration, authentication, complaint submission, complaint tracking, and administrative management. It also verifies that data is processed accurately and that the system generates appropriate responses for user actions. Effective testing enhances system reliability, improves performance, and ensures usability, thereby providing a seamless experience for both villagers and administrators under various operating conditions.

Dimensions of Testing

Software testing can be evaluated from multiple perspectives, commonly referred to as testing dimensions. Functional testing focuses on verifying whether the system performs all intended operations correctly, including user registration, login, complaint submission, and status updates. Performance testing evaluates the responsiveness and efficiency of the system, ensuring that it can handle multiple users simultaneously without significant delays. Usability testing examines the user interface to ensure that it is intuitive and easy to navigate, particularly for users with limited technical knowledge. Security testing is conducted to protect sensitive information by ensuring that unauthorized access is prevented and that user credentials and complaint data are securely managed.

Stages of Testing

Testing is conducted in several stages to ensure comprehensive system validation. Unit testing involves evaluating individual components or modules, such as the registration or login functionality, in isolation to verify their correctness.

Integration testing is performed to ensure that different modules interact seamlessly and exchange data correctly. System testing evaluates the complete application as a whole to confirm that it meets the specified requirements and performs all functions as expected. Finally, acceptance testing is carried out to determine whether the system satisfies user expectations and is ready for deployment in a real-world environment.

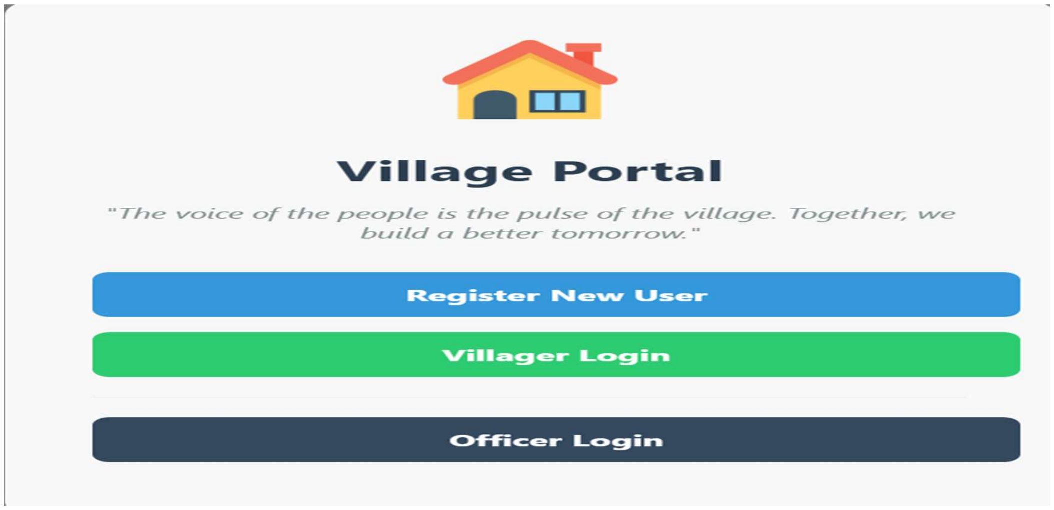
Types of Testing

Various testing approaches are used to assess different aspects of the system. Black box testing focuses on evaluating system functionality based on inputs and outputs without considering internal code structure. In contrast, white box testing examines the internal logic and structure of the code to ensure proper implementation. Manual testing is conducted by testers who interact with the system directly without the use of automated tools, allowing for detailed observation of system behavior. Additionally, user interface testing ensures that the web application interface is responsive, visually consistent, and user-friendly across different devices and browsers.

Test Cases

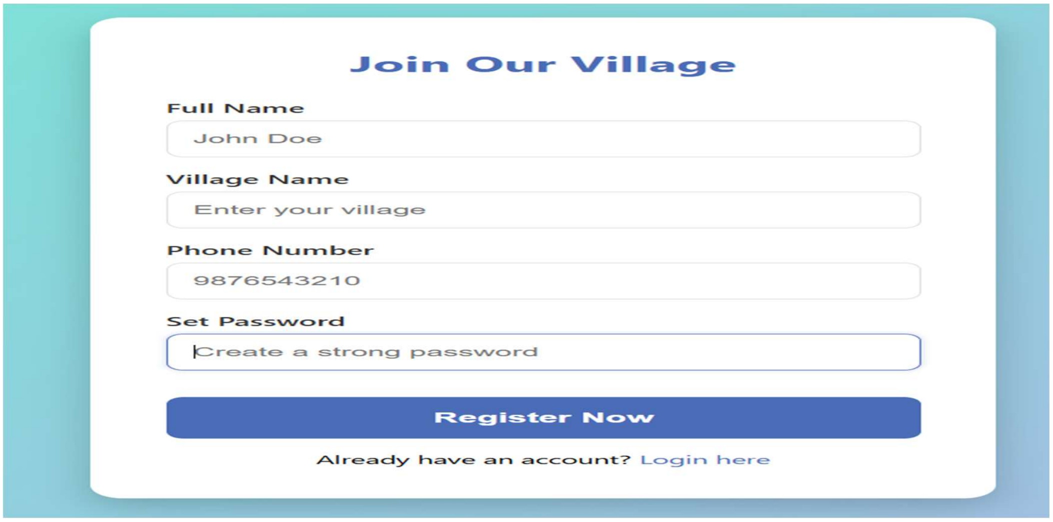
Test cases are designed to validate the functionality of the system under different input conditions. Each test case specifies the input data, expected output, and actual result to determine whether the system behaves as intended. By executing multiple test cases, developers can identify defects, verify system accuracy, and ensure that all features operate reliably. Properly designed test cases contribute to the overall quality of the system and ensure that it meets both functional and user requirements.

SCREENSHOTS



Village Portal

"The voice of the people is the pulse of the village. Together, we build a better tomorrow."



Join Our Village

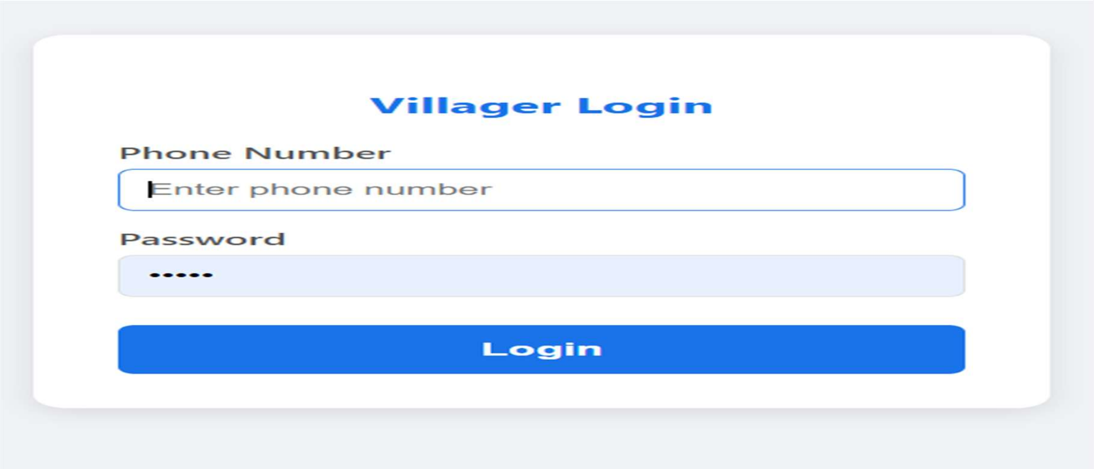
Full Name: John Doe

Village Name: Enter your village

Phone Number: 9876543210

Set Password: Create a strong password

Already have an account? Login here



Villager Login

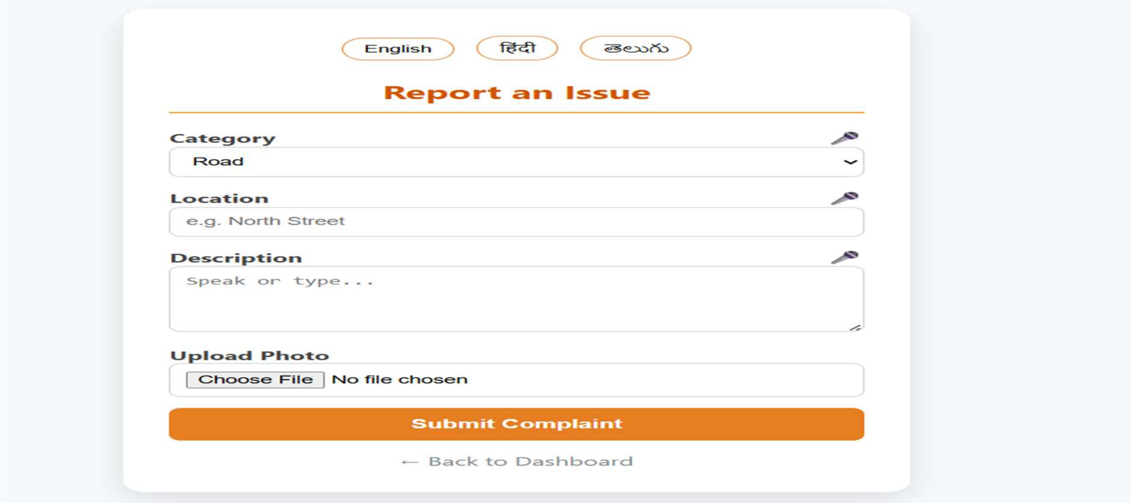
Phone Number

Enter phone number

Password

•••••

Login



English हिंदी తెలుగు

Report an Issue

Category

Road

Location

e.g. North Street

Description

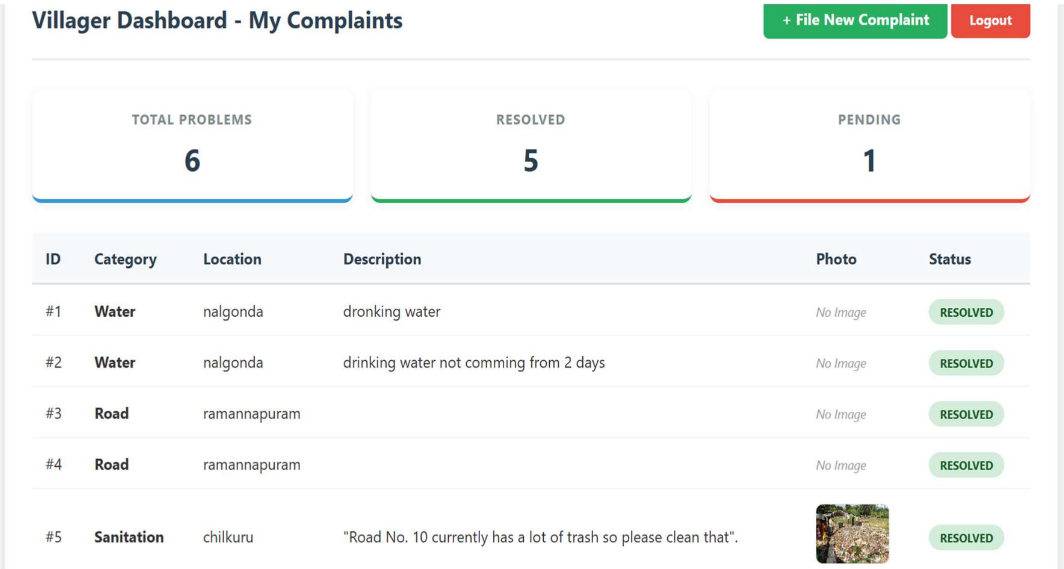
Speak or type...

Upload Photo

Choose File No file chosen

Submit Complaint

← Back to Dashboard



Villager Dashboard - My Complaints

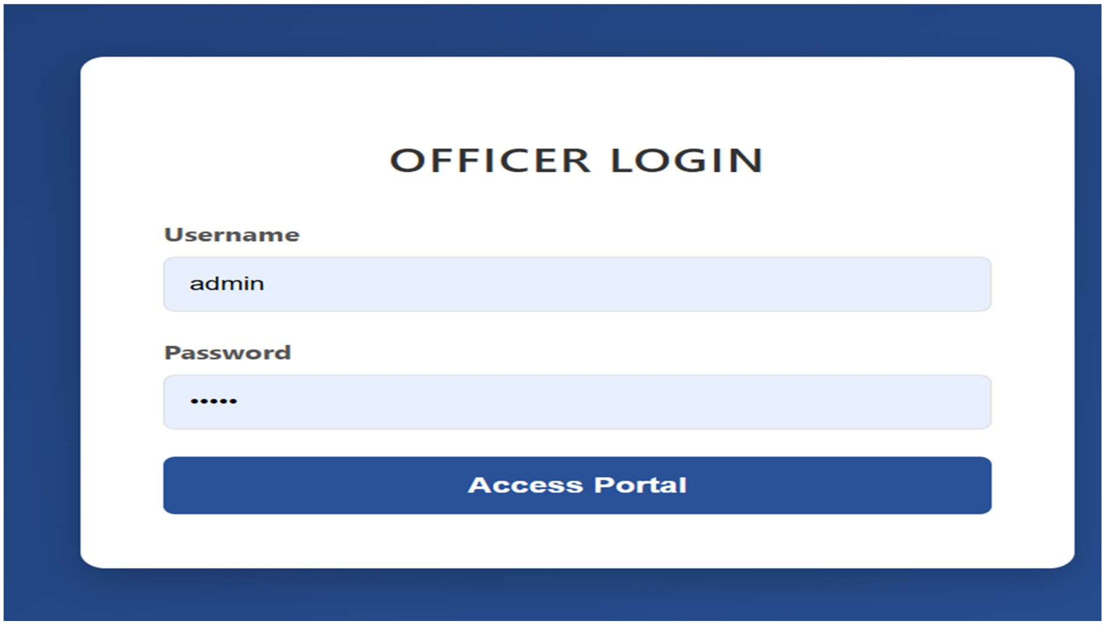
+ File New Complaint Logout

TOTAL PROBLEMS 6

RESOLVED 5

PENDING 1

ID	Category	Location	Description	Photo	Status
#1	Water	nalgonda	dronking water	No Image	RESOLVED
#2	Water	nalgonda	drinking water not comming from 2 days	No Image	RESOLVED
#3	Road	ramannapuram		No Image	RESOLVED
#4	Road	ramannapuram		No Image	RESOLVED
#5	Sanitation	chilkuru	"Road No. 10 currently has a lot of trash so please clean that".		RESOLVED



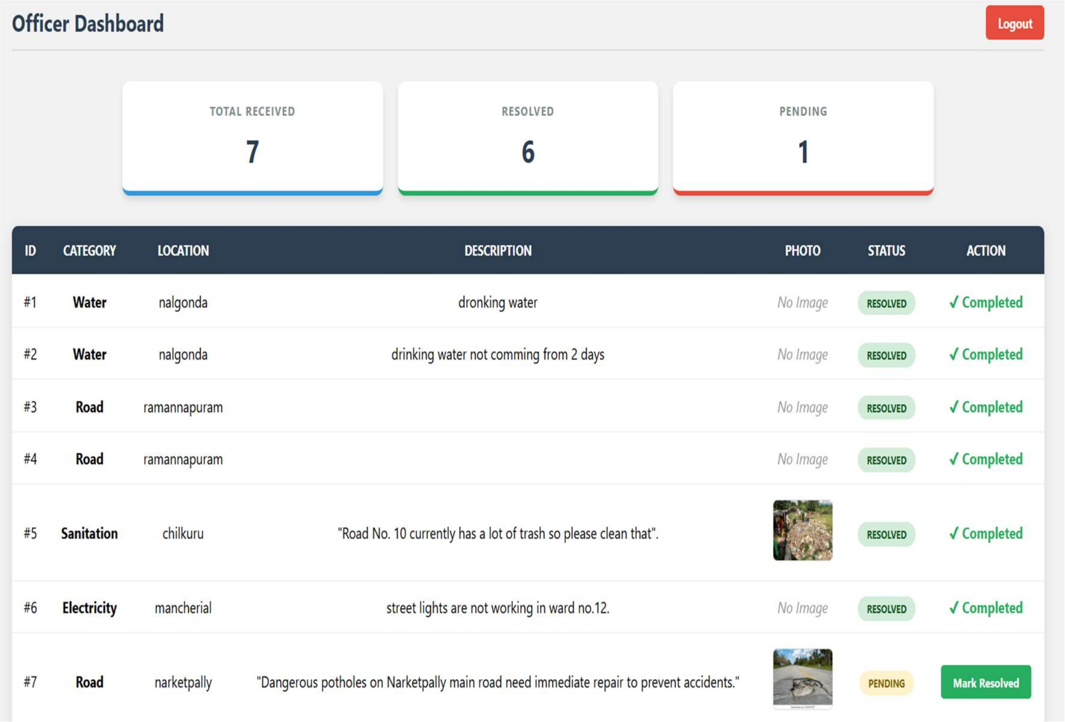
OFFICER LOGIN

Username

admin

Password

•••••



Officer Dashboard

TOTAL RECEIVED	RESOLVED	PENDING
7	6	1

ID	CATEGORY	LOCATION	DESCRIPTION	PHOTO	STATUS	ACTION
#1	Water	nalgonda	dronking water	No Image	RESOLVED	✓ Completed
#2	Water	nalgonda	drinking water not comming from 2 days	No Image	RESOLVED	✓ Completed
#3	Road	ramannapuram		No Image	RESOLVED	✓ Completed
#4	Road	ramannapuram		No Image	RESOLVED	✓ Completed
#5	Sanitation	chilkuru	"Road No. 10 currently has a lot of trash so please clean that".		RESOLVED	✓ Completed
#6	Electricity	mancherial	street lights are not working in ward no.12.	No Image	RESOLVED	✓ Completed
#7	Road	narketpally	"Dangerous potholes on Narketpally main road need immediate repair to prevent accidents."		PENDING	Mark Resolved

Conclusion

The Village Problem Reporting System provides an effective digital solution for addressing common issues faced in rural areas. By enabling villagers to report problems through an online platform, the system simplifies the complaint submission process and reduces dependence on traditional manual methods. It enhances transparency by allowing users to track the status of their complaints and ensures accountability among authorities. Furthermore, the system improves communication between villagers and administrative bodies, leading to quicker response times and efficient resolution of issues. Overall, the proposed solution contributes to better governance, streamlined operations, and improved quality of public services in rural communities.

Future Scope

The proposed system can be further enhanced by integrating advanced technologies and additional features to improve accessibility and efficiency. A dedicated mobile application can be developed to allow users to report issues conveniently using smartphones. Integration of GPS-based location tracking would enable precise identification of problem locations, assisting authorities in faster response and resolution. An audio-based complaint feature could also be introduced, allowing users—especially elderly or illiterate individuals—to record and submit complaints using voice input.

Additionally, the incorporation of Artificial Intelligence techniques can help in automatically analyzing uploaded images and categorizing complaints based on the type of issue. Deploying the system on cloud infrastructure would further improve scalability, data security, and overall performance. These enhancements would make the system more robust, inclusive, and adaptable to future technological advancements, thereby strengthening rural digital governance.

References

[1] A. Degada, H. Thapliyal, and S. P. Mohanty, “Smart Village: An IoT-Based Digital Transformation,” 2021.

[2] P. K. Sinha, M. A. K. Akhtar, and A. Kumar, “Impact of Internet of Things Applications in Smart Villages,” *CEUR Workshop Proceedings*, 2024.

[3] Q. Wang, S. Luo, J. Zhang, and K. Furuya, “Increased Attention to Smart Development in Rural Areas: A Scientometric Analysis of Smart Village Research,” *Land*, vol. 11, no. 8, 2022.

[4] G. Natarajan and L. Ashok Kumar, “Implementation of IoT-Based Smart Village for Rural Development,” *International Journal of Mechanical Engineering and Technology*, vol. 8, no. 8, pp. 1212–1222, 2017.

[5] V. Emerllahu and D. Bogataj, “Smart Villages as Infrastructure of Rural Areas: Literature Review and Research Agenda,” 2024.