NAGAR NIRMAN

Menu About Us @rnagarnirrman

NagarNirman

A Smart City Issue Reporting & Management System

A Complete End-to-End Project Built with Python and Streamlit.

Live Demo

NAGAR NIRMAN

Menu About Us @rnagarnirrman

The Problem : Barriers to Civic Engagement

Difficulty in Digital Reporting

Citizens lack user-friendly, centralized platforms to digitally report civic issues like potholes, poor lighting, or sanitation problems.

Lack of Transparency & Tracking

Once reported, there is no proper mechanism for tracking the status of the issue, leading to citizen frustration and distrust.

Manual System Inefficiency

Traditional manual reporting processes result in significant delays, high operational costs, and often, issues falling through the cracks.

Communication Gap

A persistent divide between the concerns of the community and the awareness/response of municipal authorities hinders effective city management.

NAGAR NIRMAN

Menu About Us @rnagarnirrman

Project Objectives: Enhancing Transparency and Accountability

User-Friendly Platform

Design and deploy a simple, intuitive platform for citizens to submit reports quickly.

Issue Tracking Capability

Enable real-time tracking for every submitted issue, from submission to resolution.

Authority Dashboard

Develop a dedicated administrative interface for efficient issue monitoring and management.

Ensure Accountability

Foster greater transparency in the city management process, holding responsible parties accountable.

Technology Stack: Modern and Efficient Tools

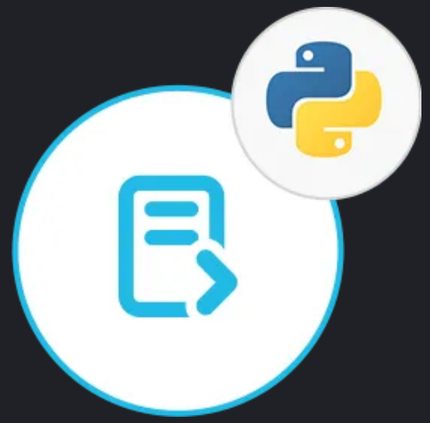
Our project leverages a robust and efficient technology stack chosen for rapid development and maintainability.

Python

The core programming language, utilized for all application logic and data handling.

Streamlit

The web application framework used for fast prototyping and creating the interactive user interface.



JSON

Selected for its lightweight structure and ease of use in storing application data and configurations.

Streamlit Community Cloud

The platform chosen for seamless, continuous deployment and hosting of the application.

System Architecture Overview

Streamlit UI

The front-end interface where citizens and administrators interact with the system.

app.py Router

The main controller handling routing, application initialization, and user sessions.

Views & Utils

Modular separation: Views handle the presentation layer (UI pages), and Utils encapsulate core business logic (authentication, data management).

JSON Files

Serve as the lightweight data persistence layer for all reports and user information.

NAGAR NIRMAN

Menu About Us @rnagarnirrman

Folder Structure

- **app.py**

  The central entry point and main application controller; initializes the application state and navigates between different views.

- **views/**

  Holds individual Python files for each major page or UI component (e.g., login, dashboard, specific report views).

- **utils/**

  Contains reusable modules for business logic, such as user authentication (`auth.py`), persistent data operations (`data_handler.py`), and application logic.

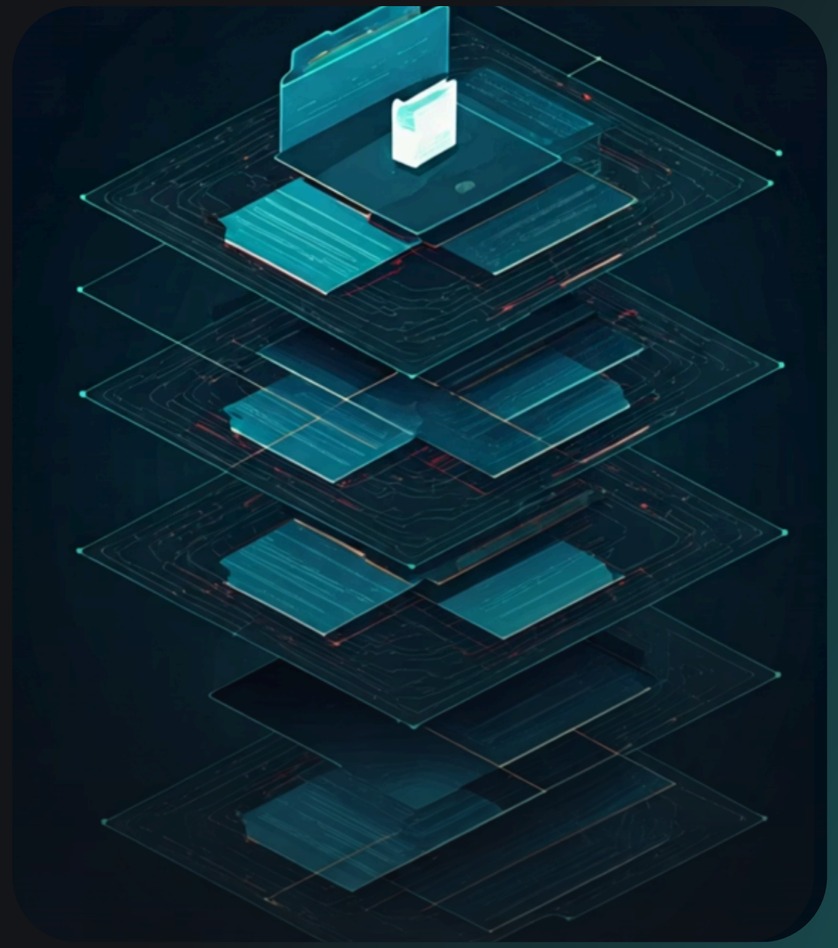
- **assets/**

  Stores static resources, including custom CSS for styling and any images used in the application interface.

- **JSON Files**

  Act as the lightweight, file-based database for storing structured application data, including user credentials and all issue reports.

```
├── app.py
├── views/
│   ├── login.py
│   ├── dashboard.py
│   └── reports.py
├── utils/
│   ├── auth.py
│   ├── data_handler.py
│   └── logic.py
├── assets/
│   ├── style.css
│   └── images/
├── reports.json
└── users.json
```



NAGAR NIRMAN

Menu About Us @rnagarnirrman

Core Functionalities: A Citizen-Centric Approach

User Authentication & Sessions

Secure login mechanism and robust session management using Streamlit's `session_state` to maintain user context across interactions.

Issue Reporting with Location

A simplified form allowing users to submit new civic issues, categorized by type (e.g., waste, infrastructure) and pinpointed by location data.

User-Specific Report History

Citizens can access a dedicated history view to monitor the status—such as "Reported," "In Progress," or "Resolved"—of only the issues they submitted.

Admin Monitoring Dashboard

A comprehensive view for administrators to track all incoming reports, filter by category/status, and manage the issue resolution workflow.

Challenges and Engineering Solutions

Addressing key technical hurdles during development ensured the application's stability and reliability.

Data Persistence

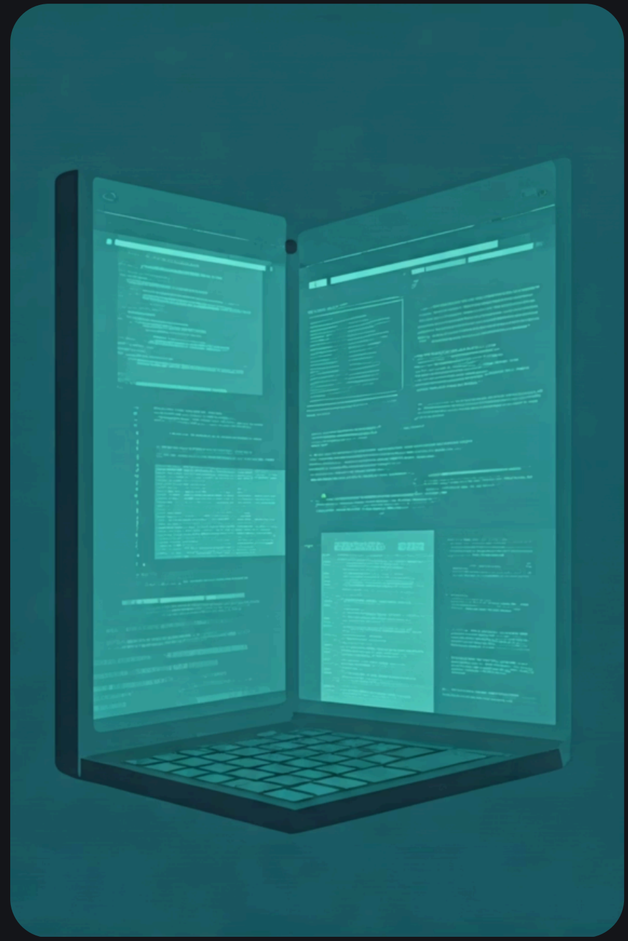
Ensuring the JSON data was reliably saved and loaded, especially given the ephemeral nature of cloud-hosted applications after restarts.

Solution: Structured JSON Read/Write Logic with error handling to guarantee data integrity across sessions.

Session Management

The default behavior of Streamlit often results in the loss of user session data upon browser refresh or widget interaction.

Solution: Extensive use of Streamlit's `session_state` to maintain user login state and application data integrity.



Deployment Process: Continuous Integration

Code Push to GitHub

All updated code is version-controlled and pushed to the designated GitHub repository.

Dependency Manifest

The `requirements.txt` file is verified to ensure all necessary Python libraries are listed for the deployment environment.

Cloud Connection

The GitHub repository is connected directly to the Streamlit Community Cloud platform.

Main File Selection

The `app.py` file is selected as the entry point for the deployed application.

Automatic CI/CD

Automatic Continuous Integration/Continuous Deployment ensures that every push to the main branch triggers an automatic build and update.

NAGAR NIRMAN

Menu About Us @rnagarnirrman

Meet the NagarNirman Development Team

Habibur Rahman Zihad

ID: E241024

Saimon Uddin Imam

ID: E241014

Rohit Ahemed

ID: E241013

Minhajun Noor Miraj

ID: E241027

NAGAR
NIRMAN

@rnagarnirrman

python beta version

thank you

If you have any question.....
please keep it for yourself.....I’m not chatgpt😑