Project proposal: Civic Lens – A Local Government Transparency Dashboard

23BCS10204

Mankaran Singh Tandon

1. Description

Civic Lens is a full-stack web application aimed at increasing public visibility into local government activities. The dashboard brings together municipal data—such as budgets, upcoming projects, public meetings, and citizen feedback—in a single, user-friendly platform. It's designed for people who care about their localities but struggle to stay updated or feel disconnected from civic decisions. The project focuses on converting raw administrative data into digestible, interactive formats that genuinely help residents understand what's happening in their area.

It's not just about transparency—it's about *comprehensibility* and *accessibility*.

2. Problem Statement

In most local governments, public data is either buried under poorly designed portals, uploaded as scanned PDFs, or never published in the first place. Even when data is technically "available," it's not actually usable for the average citizen. There's a missing link between local governance and the people it's supposed to serve.

Civic Lens is built around this idea: *What if understanding your municipal budget or knowing about an upcoming roadwork project was as easy as checking your weather app?*

By giving structure to chaos—through clean UI, clear data visualization, and contextual summaries—Civic Lens helps bridge the gap between civic action and civic awareness.

3. System Design

High-Level Design

- **Frontend**

    - Built with **React**, the UI is clean, responsive, and focused on visual clarity.

- Users can explore modules like **Budget Tracker**, **Project Map**, **Council Meetings**, and **Citizen Voice** without confusion or clutter.

- **Backend**

  - Powered by **Spring Boot**, handling user roles (citizen/admin), authentication, feedback submissions, and scheduled data syncs.

  - REST APIs handle communication between frontend and backend cleanly.

- **Database**

  - Using **MongoDB** for flexibility in storing diverse data types like JSON budget records, event details, and user feedback.

- **Data Flow**

  - User interacts with the React interface.

  - Requests hit Spring Boot APIs to fetch data.

  - Backend queries MongoDB and returns data in digestible chunks.

  - Data is displayed using components like graphs, tables, and maps.

Low-Level Modules

- **User Login & Roles**

  - Citizens access public modules.

  - Admins can upload datasets, push announcements, or moderate feedback.

  - Optional OAuth2 login (Google, institutional).

- **Budget Tracker**

  - Displays how local funds are allocated and spent.

  - Year-wise and department-wise breakdowns.

  - Graphical visualizations (bar, pie, trendlines).

- **Project Map**

  - Map view of current and proposed infrastructure/public service projects.

  - Search and filter by ward, department, or status.

- o   Timeline-based status updates.

- **Council Meetings**

    - o   List of upcoming municipal meetings with agenda points.

    - o   Archive of minutes from past meetings.

    - o   Commenting/discussion threads (optional).

- **Citizen Voice**

    - o   Submit feedback, suggestions, or complaints.

    - o   Vote or upvote on issues raised by others.

    - o   Admin view for triaging and responding.

4. Application Type

- **Type**: Full-stack web application using React + Spring Boot.

- **Domain**: Civic Tech / GovTech.

- **Target Users**:

    - o   Urban/rural residents

    - o   RTI activists and policy researchers

    - o   Civic-minded students and NGOs

    - o   Local government administrators

- **Platform**: Web-based, responsive on both desktop and mobile.

5. Why This Project Works

Civic Lens is not just a project—it's a *usable product idea*. It's realistic to build within a semester with iterative modules, but it also has real-world applicability. It blends coding, design, and meaningful data storytelling—ideal for someone who likes problem-solving, likes their work to *matter*, and isn't afraid to confront messy systems and make them a little less messy.

The architecture is scalable and could easily be extended to include:

- RTI request tracker

- Real-time pothole mapping (via citizen reports)

- Budget vs. outcome analytics

- Public grievance resolution timelines