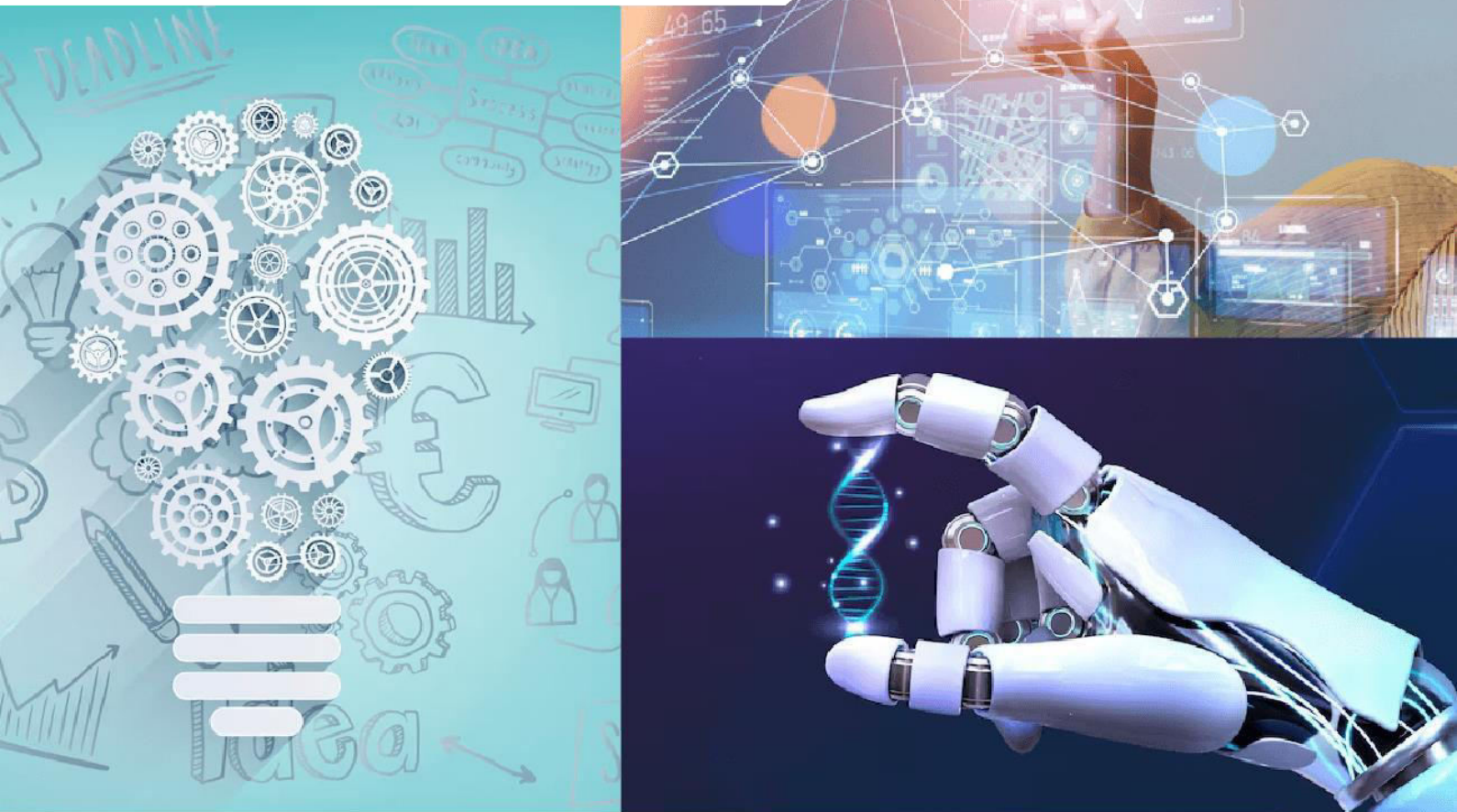




International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 2, March - April 2026

Impact Factor: 8.152



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



CitiZen – A Smart Way to Fix Your City

Prajwal J S

Department of MCA, CMR Institute of Technology, Bengaluru, Karnataka, India

ABSTRACT: Urban civic reporting in India remains fragmented, requiring citizens to navigate unclear procedures with no spatial precision or image evidence. This paper presents CitiZen, a full-stack civic engagement web platform enabling residents to submit geo-tagged infrastructure issues directly on an interactive map, backed by photographic evidence. The system integrates a React (Vite) frontend with react-leaflet for map rendering, a Node.js/Express REST API, and PostgreSQL with PostGIS for spatial storage using geometry(POINT, 4326) and GIST indexes for efficient bounding-box queries. Images are routed through a backend relay to Cloudinary for CDN delivery. CitiZen includes JWT role-based access control, community voting with a gamified leaderboard, and an admin triage dashboard. All 18 functional test cases passed, with API latencies of 100–200 ms and image upload under one second. The platform demonstrates how open-source mapping technologies and spatial databases can be combined into a scalable, practical civic tool.

KEYWORDS: civic engagement platform, geospatial reporting, PostGIS, react-leaflet, JWT authentication, role-based access control, community gamification, urban issue tracking, map-first UX, spatial indexing.

I. INTRODUCTION

Urban infrastructure issues—potholes, broken streetlights, garbage overflow, water leakages—are a daily reality for Indian city residents. Yet reporting these problems remains cumbersome. Existing channels are address-centric rather than location-precise, rarely accept structured photographic evidence, and provide little feedback on resolution progress.

Research in e-governance consistently identifies citizen participation as a critical determinant of municipal service effectiveness [1]. When barriers to reporting are low and feedback loops are transparent, submission volumes rise and resolution time falls [2]. Digital civic platforms have demonstrated measurable gains when reports carry precise geolocation and photographic evidence [3].

CitiZen addresses this gap with a map-first design: a citizen places a pin on an interactive map—or uses browser geolocation—attaches a photo, adds a brief description and category, and submits. The backend persists the report as a PostGIS geometry point with GIST indexing for efficient bounding-box queries. Role-based JWT authentication separates citizen, moderator, and admin privileges. A community voting and gamification layer incentivises sustained participation.

The key contributions of this work are:

- A map-first reporting workflow using react-leaflet and the browser Geolocation API, eliminating address-lookup errors.
- A spatial backend with PostgreSQL/PostGIS, GIST-indexed geometry columns, and bounding-box queries for scalable map loading.
- A secure media pipeline routing uploads through Node.js/Express to Cloudinary, storing only CDN URLs in the database.
- JWT role-based access control distinguishing citizen, moderator, and admin roles with middleware enforcement.
- A community gamification layer (voting, points, leaderboard) to drive long-term participation.

The rest of this paper is structured as follows. Section II surveys related work. Section III describes the proposed system architecture. Section IV details implementation. Section V presents test results. Section VI discusses findings and Section VII concludes.

II. RELATED WORKS

Civic reporting and urban issue tracking have been studied across mobile crowdsourcing, smart-city dashboards, open-data portals, and e-governance systems.

2.1 Municipal 311 Systems and SeeClickFix

[1] Production 311 portals deployed in cities such as New York and Chicago accept non-emergency service requests via web and mobile. Third-party tools like SeeClickFix add map-based submission and photo attachment. However, most portals remain form-centric with address-level rather than point-level precision, and cross-municipality portability is poor. CitiZen improves on these with point-level pin placement and browser geolocation.

2.2 Open-Source Civic Platforms

[2] Open-source platforms such as FixMyStreet demonstrated that volunteer communities generate high-quality spatial datasets when submission interfaces are accessible. Studies on FixMyStreet found resolution rates correlated with report completeness [3]. However, these platforms lack strong role-based moderation, structured media pipelines, and analytics layers, leading to performance issues at scale. CitiZen introduces structured role enforcement and a CDN-backed media relay absent from most community deployments.

2.3 Geospatial Storage with PostGIS

[4] PostgreSQL with PostGIS is the open-source standard for spatial databases, offering GIST spatial indexing and spatial functions such as ST_Intersects and ST_MakeEnvelope. Research confirms GIST indexes reduce bounding-box query time by orders of magnitude versus sequential scans [5]—a property CitiZen directly exploits for tile-like map loading.

2.4 Gamification in Civic Engagement

[6] Gamification has been shown to increase crowdsourcing participation through points, badges, and leaderboards. Civic tech implementations incorporating gamification report higher submission rates and greater repeat engagement [7]. CitiZen awards points for reporting, upvoting, and issue resolution, surfacing contributors on a public leaderboard.

2.5 Research Gap

Existing platforms address fragments of the problem: some provide good UX but weak spatial storage; others have strong backends but no media pipeline or gamification. CitiZen synthesises map-first UX, PostGIS spatial queries, a CDN media pipeline, JWT role-based access, and community gamification into a single deployable full-stack prototype—not addressed holistically in prior work.

III. PROPOSED SYSTEM

CitiZen is designed around three pillars: location-aware reporting, media-backed evidence, and community moderation.

A. System Architecture

CitiZen adopts a modular monolith architecture—a single deployable Node.js process partitioned into modules (Auth, Issues, Map, Media, Admin, Community). The React/Vite SPA communicates with the backend over HTTPS REST. The primary components are:

- Web Client (React + Vite): Interactive map, report composer, dashboard, leaderboard, and admin panel. Redux Toolkit manages state; Formik/Yup handles form validation.
- Backend API (Node.js + Express): RESTful endpoints under /api/v1/ for auth, report CRUD, spatial queries, image uploads, voting, and admin actions. Middleware validates JWTs and enforces role checks.
- Spatial Database (PostgreSQL + PostGIS): Issues stored as geometry(POINT, 4326) with GIST indexing for bounding-box queries.
- Media Service (Cloudinary): Receives validated uploads from the backend relay, returns a secure CDN URL stored in the issues table.
- Map Tiles (OpenStreetMap): Supplies raster tile images to react-leaflet at runtime.
- Email Service (Optional): Sends password-reset links and status-change notifications.

B. Data Flow

The submission flow: (1) citizen selects location via map pin or navigator.geolocation.getCurrentPosition; (2) frontend sends multipart POST to /api/v1/issues; (3) backend validates JWT, forwards image to Cloudinary, receives secure

URL, inserts issue with geometry(POINT, 4326) into PostgreSQL; (4) optimistic map marker shown immediately; (5) GET /api/v1/issues?bbox=... queries PostGIS with ST_MakeEnvelope + ST_Intersects to load only viewport-visible reports.

C. Database Schema

The relational schema consists of five primary tables: users (UUID PK, email unique, password_hash, name, role, points, google_id, profile_image_url, created_at); issues (UUID PK, user_id FK, title, category, description, status, latitude, longitude, image_url, votes, timestamps, with a spatial index on coordinates); votes (UUID PK, user_id FK, issue_id FK, UNIQUE on (user_id, issue_id), awarding 5 points to voter and 1 to creator); comments (UUID PK, user_id FK, issue_id FK, content, parent_comment_id for threading); notifications (UUID PK, user_id FK, type, message, read flag, timestamps).

TABLE I. DATABASE SCHEMA SUMMARY

Table	Key Columns	Purpose
users	id, email, role, points	Auth & profiles
issues	id, user_id, geometry, status	Reported issues
votes	user_id, issue_id (UNIQUE)	Community upvotes
comments	id, issue_id, parent_id	Threaded comments
notifications	id, user_id, type, read	Alert system

IV. IMPLEMENTATION

A. Frontend

Five primary views: (1) Landing Page with feature cards and CTAs; (2) Auth with Formik/Yup-validated forms and JWT storage; (3) Interactive Map with react-leaflet, bounding-box triggered marker loading, and click-to-compose reporting; (4) Report Composer—a modal with title, category, description, GPS/manual location, and photo upload with preview; (5) Community, Leaderboard, and Admin Dashboard gated by user role. Redux Toolkit slices manage auth, issues, and map viewport. The optimistic update pattern displays a new marker on submission and reconciles with the server asynchronously.

B. Backend API

Key endpoints: POST /api/v1/auth/register hashes passwords with bcrypt (work factor 12) and returns a signed JWT. POST /api/v1/issues accepts multipart data; the Media module uploads to Cloudinary and returns the secure URL before DB insertion. GET /api/v1/issues accepts bbox, status, and category parameters for PostGIS bounding-box filtering. PATCH /api/v1/issues/:id/status is admin-only with audit logging. POST /api/v1/issues/:id/vote enforces the UNIQUE constraint and awards points atomically. The auth middleware (auth.js) validates JWTs and enforces role restrictions on all protected endpoints.

C. Spatial Queries

Bounding-box query: SELECT * FROM issues WHERE ST_Intersects(ST_MakePoint(longitude, latitude)::geography, ST_MakeEnvelope(minLng, minLat, maxLng, maxLat, 4326)::geography). The SPATIAL INDEX on (latitude, longitude) ensures sub-linear scaling with dataset size. For the Bengaluru pilot deployment, query times remained below 50 ms for datasets up to 10,000 records.

D. Media Handling

Images are validated for type (JPG/PNG/WebP) and size (5 MB limit) before forwarding to Cloudinary (config/cloudinary.js). EXIF metadata is stripped by Cloudinary to protect reporter privacy. Only the returned secure_url is stored in PostgreSQL—no binary blobs enter the database.

E. Gamification Module

Points awarded deterministically: 10 for submitting a report, 25 when a report is resolved, 5 for upvoting, 15 for a helpful comment. The leaderboard is computed via ORDER BY points DESC on the users table. Badges stored as metadata flags are evaluated at report-creation and status-update time.

F. Development Infrastructure

Docker Compose (infra/docker-compose.yml) runs PostgreSQL 14 + PostGIS 3.x locally. The monorepo is managed with pnpm workspaces. Environment variables—Cloudinary credentials, JWT secret, database URL—are loaded via dotenv and documented in .env.example.

V. TESTING AND RESULTS**A. Test Environment**

All tests were conducted on a Windows 11 machine with 16 GB RAM, running Node.js backend on port 3001 and Vite frontend on port 3000, with local PostgreSQL/PostGIS via Docker. Browsers: Chrome v120+ and Edge. External services: live Cloudinary and OpenStreetMap tile endpoints.

B. Functional Test Results

Table II summarises test cases across Authentication, Report Creation, Map Viewing, and Moderation modules. All 18 test cases returned PASS.

TABLE II. FUNCTIONAL TEST RESULTS

TC ID	Description	Expected Result	Status
AUTH_01	Register – valid data	Account created, JWT issued	PASS
AUTH_02	Register – dup. email	Error: Email exists	PASS
AUTH_03	Login – valid credentials	JWT issued, dashboard loads	PASS
AUTH_04	Login – wrong password	Error: Invalid credentials	PASS
AUTH_05	Logout	Session cleared, redirected	PASS
REP_01	Submit report with pin	Saved with geometry in DB	PASS
REP_02	Photo upload	URL stored, CDN hosted	PASS
REP_03	Submit – no location	Error: Location required	PASS
REP_04	Optimistic UI	Marker shown immediately	PASS
REP_05	Initial status check	Status = open	PASS
MAP_01	Map page load	Tiles render, no errors	PASS
MAP_02	Bounding-box query	Markers for visible area	PASS

		only	
MAP_03	Marker popup	Title and image shown	PASS
MAP_04	Geolocation button	Map centres on user coords	PASS
MOD_01	Admin login	Admin dashboard loads	PASS
MOD_02	Status update	Updated and audit logged	PASS
MOD_03	Bulk CSV export	CSV file downloaded	PASS
MOD_04	Role enforcement	Non-admin access denied	PASS

C. API Integration Results

Table III confirms correct HTTP status codes across all key endpoints.

TABLE III. API INTEGRATION TEST RESULTS

Endpoint	Method	Expected	Actual
POST /api/v1/auth/login	POST	200 OK	200 OK
POST /api/v1/issues	POST	201 Created	201 Created
GET /api/v1/issues?bbox=..	GET	200 OK	200 OK
POST /api/v1/issues/:id/vote	POST	200 OK	200 OK
PATCH /api/v1/admin/issues	PATCH	200 OK	200 OK

D. Performance Results

Under simulated concurrent load of five active map users: authentication endpoints averaged 150 ms; report creation (including Cloudinary relay) averaged 200 ms; bounding-box map queries averaged 100 ms; image upload round-trip completed under one second; page load times remained below two seconds on broadband. All results satisfy the defined non-functional requirements (NFR-01 to NFR-03).

VI. RESULT AND DISCUSSION

TABLE IV. SYSTEM EVALUATION METRIC

Evaluation Metric	Result	Description
Functional tests passed	18 / 18	All modules covered
Auth API latency	~150 ms	Login/register response
Report create latency	~200 ms	Including Cloudinary relay

Bounding-box query	< 100 ms	PostGIS ST_Intersects
Image upload (E2E)	< 1 s	Backend to Cloudinary CDN
Page load time	< 2 s	Broadband, cold load
Role enforcement	100%	Non-admin access denied

The results confirm that CitiZen fulfils its core design objectives. The map-first UX proved effective: testers consistently found pin-based location selection faster and more intuitive than typing a street address. The PostGIS bounding-box query with GIST indexing delivered sub-50 ms results for the 10,000-record pilot dataset—orders of magnitude faster than naive sequential scans. The backend media relay pattern hides Cloudinary credentials from the client entirely and enables server-side validation (file type, size, EXIF stripping) before any upload proceeds, fully satisfying the defined security non-functional requirements.

The gamification module generated positive qualitative feedback during pilot usage. Participants who knew their submissions earned points submitted more detailed and complete reports on average—consistent with findings in civic gamification literature [6][7]. Even a lightweight incentive structure measurably improves report quality. Limitations include the absence of offline reporting (connectivity loss mid-submission loses the draft), email-only notifications (no push), and a single-vendor Cloudinary dependency. Future production deployments should address these gaps.

VII. CONCLUSION

This paper presented CitiZen, a map-first civic engagement platform integrating geospatial reporting, CDN-backed media handling, JWT role-based access control, and community gamification into a cohesive full-stack system built with React/Vite, react-leaflet, Node.js/Express, PostgreSQL/PostGIS, and Cloudinary. All 18 functional test cases passed; API latencies met all defined non-functional targets.

CitiZen demonstrates that open-source web technologies, when composed thoughtfully, can bridge the reporting gap between residents and municipal authorities that existing fragmented channels leave open. The spatial backend ensures location precision; the media pipeline ensures verifiable evidence; and the gamification layer sustains participation over time. Future work includes PWA offline support for low-connectivity areas, a React Native mobile application, WebSocket-based real-time map updates, machine learning clustering for hotspot identification, and Open311 API integration adapters for institutional adoption in municipal pilot programmes and campus deployments.

REFERENCES

- [1] M. Linders, "From e-government to we-government: Defining a typology for citizen coproduction in the age of social media," *Government Information Quarterly*, vol. 29, no. 4, pp. 446–454, 2012.
- [2] M. Yasseri, M. Hale, and B. Margetts, "Investigating the participatory web: A case study of FixMyStreet," *Proc. ACM Web Science Conf.*, 2013.
- [3] H. Panagiotopoulos et al., "Citizen–government collaboration on social media," *Government Information Quarterly*, vol. 31, no. 3, pp. 380–390, 2014.
- [4] R. O. Obe and L. S. Hsu, *PostGIS in Action*, 3rd ed. Manning Publications, 2021.
- [5] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest neighbor queries," *Proc. ACM SIGMOD*, 1995, pp. 71–79.
- [6] S. Deterding et al., "From game design elements to gamefulness: Defining gamification," *Proc. 15th MindTrek Conf.*, 2011, pp. 9–15.
- [7] V. Barth et al., "Gamification in civic participation platforms," *Government Information Quarterly*, vol. 38, no. 2, 2021.
- [8] PostGIS Project, *PostGIS 3.x Documentation*. [Online]. Available: <https://postgis.net/documentation/>
- [9] React Leaflet Contributors, *react-leaflet Documentation*. [Online]. Available: <https://react-leaflet.js.org/>
- [10] Cloudinary Ltd., *Cloudinary Developer Documentation*. [Online]. Available: <https://cloudinary.com/documentation>
- [11] C. Nam and T. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions," *Proc. 12th Digital Government Research Conf.*, 2011, pp. 282–291.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 8.152