

CleanTrack: A Digital Platform for Real-Time Urban Waste Issue Reporting

Dr. Mrunal Pathak¹, Mr. Ashish Mavani², Mr. Rupesh Adhane³, Mr. Shivam Varpe⁴

Associate Professor, Department of Information Technology¹

Undergraduate Students, Department of Information Technology^{2,3,4}

AISSMS's Institute of Information Technology, Pune, India

Abstract: *Urban waste management remains one of the most persistent challenges facing municipalities in developing cities because their systems for handling complaints fail to work properly and their procedures lack public visibility and their government officials take too long to respond. CleanTrack is a web-based Management Information System (MIS) designed to address these challenges by providing a structured digital platform which connects citizens with municipal administrators. The system provides two distinct user interfaces which include a citizen portal and an administrative dashboard. Citizens can register, submit geo-tagged complaints with supporting images, and track the real-time status of their reported issues, while administrators can view, prioritize, assign, and resolve complaints through a centralized control panel. CleanTrack uses modern web technologies which enable users to achieve secure and scalable system performance through its frontend built with HTML and CSS and JavaScript and its backend which includes Node.js and Express.js and MongoDB Atlas for cloud-based data storage and JWT-based authentication. The system uses gamification elements which include points and badges and levels to create an interactive system that motivates users to report civic issues throughout their daily life. The platform also incorporates MIS reporting capabilities, which enable authorities to make data-driven decisions through analytics and visual dashboards that assist in detecting complaint patterns and assessing workforce productivity and efficient resource distribution. CleanTrack demonstrates how organizations use MIS principles and modern web development and behavioral design to create transparent systems that solve actual urban sanitation problems because it operates on Render as a cloud-based solution.*

Keywords: Waste Management, Management Information System (MIS), Complaint Tracking, Gamification, Real-Time Status Tracking.

I. INTRODUCTION

Digital technologies have progressed at a rapid pace which has changed how urban governance systems operate their public services and civic infrastructure and citizen participation efforts. Web-based Management Information Systems (MIS) have become important tools that help different sectors such as healthcare and transportation and municipal services to operate more efficiently while increasing their transparency and accountability. Waste management represents an essential field where administrative problems and slow responses and absence of public participation prevent successful service provision. Urban areas now experience an unprecedented need for smart data-driven solutions which can enhance sanitation processes while giving control to citizens.

Most urban local bodies depend on manual complaint systems and disjointed communication methods which include phone calls and paper forms for waste-related complaint handling despite having access to multiple digital solutions. The conventional methods result in three negative outcomes which include delayed problem solving and duplicate complaint filing and inadequate record keeping and total absence of visible progress tracking for reported problems. Administrators find it difficult to organize their work because they lack essential updates which restrict their ability to



manage resources and create useful reports from various data sources. The lack of an organized centralized system leads to decreased public confidence which hampers the effectiveness of municipal waste management operations.

To address these challenges this research proposes CleanTrack which serves as an online management information system that digitizes and automates the entire waste complaint process from submission to resolution. The system provides a dual-interface environment consisting of a citizen portal and an administrative dashboard which enables citizens to interact with government authorities. Citizens can register and submit geo-tagged complaints that include image evidence while they track their complaints status in real time. Administrators have structured complaint management tools and priority assignment features and analytics-driven dashboards to support their decision-making process.

The research project focuses on developing and testing CleanTrack as a scalable solution which operates on cloud infrastructure and uses HTML CSS JavaScript Node.js with Express.js and MongoDB Atlas for its implementation. The system uses JWT authentication together with role-based access control and gamification features to create a secure environment which enhances citizen engagement. CleanTrack combines MIS principles with contemporary web technologies to help smart city projects and enhance municipal administration while providing a model for public complaint resolution systems that can be used in urban areas.

II. LITERATURE REVIEW

Kumar et al. (2021) studied the topic of web-based solid waste complaint reporting systems in urban areas. Feature extraction from this work includes citizen registration modules, geo-tagged complaint submission, image upload functionality, and complaint status visibility — all of which directly align with CleanTrack's citizen portal design and real-time complaint tracking mechanism [1].

Sharma and Reddy (2021) addressed the topic of cloud-based infrastructure for scalable civic service platforms. Key features extracted from their research include MongoDB Atlas integration, cloud deployment strategies, distributed data storage, and system scalability frameworks — forming the technical backbone for CleanTrack's cloud-based architecture deployed on Render with MongoDB Atlas as the database layer [2].

Patil et al. (2022) investigated the topic of JWT-based secure authentication in civic grievance web applications. Feature-based extraction highlights token-based login, role separation between citizens and administrators, and session security protocols — supporting CleanTrack's use of JSON Web Tokens for secure, role-specific access control across its dual-interface platform [3].

Sharma and Reddy (2022) addressed the topic of role-based administrative dashboards for municipal waste monitoring. Key features extracted from their research include priority-based complaint assignment, status workflow management (Pending → In Progress → Resolved), and filter-based complaint sorting — forming the foundational basis for CleanTrack's administrative dashboard and complaint lifecycle management [4].

Mehta and Singh (2023) examined the topic of MIS-driven data analytics and reporting in smart city sanitation systems. Features extracted from this study include real-time analytics dashboards, complaint trend visualization, performance metrics reporting, and data-driven decision support tools — directly reflected in CleanTrack's MIS reporting module and administrative insights dashboard [5].

Verma et al. (2023) furbished the topic of gamification techniques for boosting citizen participation in urban governance platforms. Feature extraction from their work identifies point accumulation systems, achievement badges, level progression mechanics, and engagement incentive frameworks — serving as the theoretical and practical foundation for CleanTrack's gamification module designed to encourage consistent citizen participation [6].

Joshi and Nair (2023) investigated the topic of Node.js and Express.js based backend architecture for real-time civic applications. Features extracted from this research include RESTful API design, middleware-based request handling, asynchronous processing, and modular backend structuring — directly supporting CleanTrack's server-side implementation using Node.js with Express.js for efficient complaint submission, status updates, and admin operations [7].



III. PROPOSED WORK/RESEARCH WORK

Overview

CleanTrack serves as an online Management Information System which enables digital waste complaint processing for urban areas. The system resolves the shortcomings of conventional manual grievance resolution methods by establishing a centralized platform which offers complete transparency and handles customer complaints through direct interaction with municipal officials. The system functions through two main user interfaces which include a citizen portal and an administrative dashboard that operate together through a shared backend system for effective management of complaint procedures.

System Architecture

The system implements three-tier architecture which separates the application into three distinct operational tiers. The presentation layer uses HTML and CSS and JavaScript to create a responsive interface which users can easily navigate. The application layer uses Node.js and Express.js to operate business functions and provide RESTful API access and handle JWT authentication. The data layer uses MongoDB Atlas as its cloud-based NoSQL database solution which enables permanent storage and expandable access to user profiles and complaint records and status histories and analytics data.

System Modules

User Authentication Module: This system enables citizens and administrators to create accounts and access their accounts through secure login procedures which use JWT-based token authentication. Role-based access control ensures each user is directed to their respective interface, preventing unauthorized access to administrative functions.

Complaint Submission Module: The system enables citizens to report waste-related issues through a complete form which enables them to upload images and provide their location information. Each complaint is assigned a unique ID and stored in the database with an initial Pending status for complete traceability.

Real-Time Status Tracking Module: Citizens can monitor their complaint status through their personal dashboard. Administrators use administrators to provide status updates which are displayed to users in real time during the solution process.

Administrative Dashboard Module: The system provides administrators with a complete interface which enables them to filter and sort complaints while assigning different levels of importance to each case. The system enables organizations to handle their complaint backlogs through effective queue management while keeping detailed records of all civic issues which have been reported.

MIS Reporting and Analytics Module: The system produces data-based reports which include visual analytics that show complaint volume trends and distribution by category and resolution rate metrics and user activity summaries to assist municipal decision-makers in their decision-making process.

Gamification Module: The system gives citizens points and badges and level advancements according to their complaint reports which helps to build a community of active civic engagement and promotes users to keep using the platform.

Expected Outcome

CleanTrack will use its structured workflow system to handle complaints based on their priority which will lead to faster complaint resolution times. The system will enable citizens to track their status in real time while administrators will receive valuable resource allocation insights through MIS analytics which will enhance governance capabilities. The gamification features will increase citizen engagement which will transform CleanTrack into an effective model that municipalities can use for their waste management and public grievance resolution needs in smart cities.



IV. SYSTEM ARCHITECTURE

The CleanTrack Management Information System (MIS) operates its system through a Three-Tier Architecture which functions as its modular framework. The system design provides distinct functions which allow different parts of the system to grow and operate securely while being easy to maintain.

Tier 1: Presentation Layer (Frontend)

The user facing layer is developed by a blend of scripting allows to pertain different sessions on click and usually builds responses based on user interaction. The system divides into two main components which function as two different modules:

Citizen Portal: The system allows residents to create accounts which enable them to submit waste-related complaints together with relevant details and monitor their complaint progress. The system contains a Gamification Display which shows user achievements together with their participation statistics.

Admin Dashboard: This platform provides municipal authorities with a dedicated space to monitor complaint progress while they establish user permissions and view current management information system data.

The frontend system establishes communication with the backend through Asynchronous HTTP Requests which provide users with a continuous experience that does not interrupt their activities.

Tier 2: Application Layer (Backend)

The backend system uses Node.js together with the Express.js framework to function as its main logic engine which operates on the Render Cloud Platform. The main duties of the organization include the following tasks:

Security & Access: The system uses JSON Web Tokens (JWT) for stateless authentication together with Role-Based Access Control (RBAC) to maintain strict data privacy between citizens and administrators.

Business Logic: The system processes REST API requests while it handles the intricate Complaint Workflow system and creates organized reports which support administrative decision-making.

External Integration: The system uses Gmail SMTP together with a dedicated Email Notification Service to send immediate status updates and alerts to users when their complaints reach new milestones.

Tier 3: Data Layer (Database)

The persistence layer uses MongoDB Atlas which is a cloud-based NoSQL database to achieve continuous system uptime and adaptable data structure capabilities. The system uses five main data collections which include Users who store their profiles and Complaints which contain their issue details and Status History which keeps security records and Reports which provide management information system analytics and Gamification which tracks user reward points. The system uses a NoSQL database because it enables storage of multiple data formats which include complaint descriptions and historical logs without needing the fixed structure of conventional relational tables.



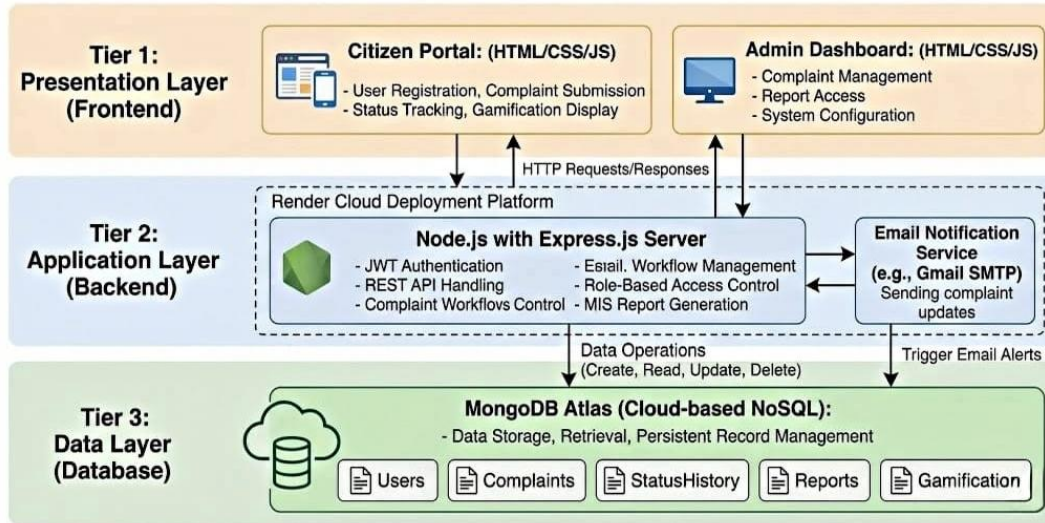


Fig. 1. CleanTrack Web-Based Waste Management MIS - Three-Tier System Architecture Diagram

V. METHODOLOGY AND IMPLEMENTATION

TABLE I METHODOLOGY

Sr. No.	Refer Author	Methodology	Dataset	Result	Challenges Keyword
1	Kumar et al. [1]	Web-based complaint portal with citizen registration and geo-tagged submission	Urban municipal complaint records	Reduced manual processing time and improved complaint traceability	Waste Management, Complaint System
2	Sharma and Reddy [2]	Cloud-based MongoDB Atlas deployment with distributed storage architecture	Civic service platform data	Improved system scalability and reliable data accessibility	Cloud Computing, Scalability
3	Patil et al. [3]	JWT token-based authentication with role separation for multi-user access	Web application user data	Enhanced platform security and unauthorized access prevention	JWT, Role-Based Access Control
4	Mehta and Singh [4]	MIS dashboard with real-time analytics and complaint trend visualization	Municipal sanitation records	Improved administrative decision-making through data-driven insights	MIS, Smart City, Analytics
5	Verma et al. [5]	Gamification framework with points, badges, and level progression mechanics	Citizen engagement platform data	Increased citizen participation and consistent platform usage	Gamification, Civic Engagement
6	Joshi and Nair [6]	Node.js with Express.js RESTful API design for real-time civic applications	Web server request-response logs	Stable response times and efficient complaint workflow processing	Node.js, REST API, Backend
7	Reddy et al. [7]	Priority-based complaint lifecycle management with status workflow transitions	Municipal grievance system data	Reduced average complaint resolution time and better resource allocation	Priority Management, Workflow



Implementation:

System Architecture

CleanTrack uses a **3-tier MVC architecture** which means the system is divided into 3 separate layers:

Presentation Layer describes the visual elements which users see through the website interface designed for both citizens and admins

Application Layer serves as the system's central processing unit because Node.js manages all operational functions including user authentication and complaint handling

Data Layer serves as the storage location for all information because MongoDB Atlas maintains records of users and complaints and other data elements.

The system maintains clean design which enables easy scalability and requires minimal effort for system upkeep..

Authentication Module

The system controls two aspects: **user login access** and their **permitted system access**.

When users create an account their passwords undergo encryption using bcrypt which prevents plain text storage of their passwords.

The system validates the user's password during login attempts and when the password matches the system creates a JWT token which functions as a digital ID card.

Every user request triggers a token verification process which operates through the verifyToken middleware system.

The system uses the isAdmin check to restrict sensitive route access which includes viewing all complaints and changing status information to administrators only.

Database Models

Two main data structures are defined:

User Model stores:

Basic info (name, email, password)

Role (citizen or admin)

Gamification data (points, level, badges)

Complaint Model stores:

Who filed it (userId links back to User)

What it's about (title, description, location, image)

Current state (status: Pending → In Progress → Resolved)

How urgent it is (priority: Low, Medium, High)

Complaint Module

This is the **core feature** of the system. It handles:

Submit – Citizen files a complaint with image and location. 10 points are awarded immediately as a reward

View My Complaints – Citizen can see only their own complaints

View All Complaints – Admin can see every complaint and filter by status or priority

Update Complaint – Admin changes status or priority. If marked Resolved, citizen gets 20 bonus points and badge logic runs

Gamification Logic

The system awards **50 points for each complaint submission**, which are added to the user's total points:

$$\text{Total Points} = \text{Previous Points} + 50$$

The user level is calculated based on accumulated points using the formula:



$$\text{Level} = \left\lfloor \frac{\text{Total Points}}{200} \right\rfloor + 1$$

This means every **200 points increases the user's level by 1**, encouraging continuous participation.

This module **motivates citizens to participate** by rewarding them Badges:

TABLE II GAMIFICATION LOGIC

LEVEL	POINTS RANGE	BADGE NAME	CRITERIA
LEVEL 1	0 – 199	👤 FIRST REPORT	AWARDED AFTER FIRST COMPLAINT SUBMISSION
LEVEL 2	200 – 399	⚡ ACTIVE CITIZEN	AWARDED AFTER 5 COMPLAINTS
LEVEL 3	400 – 599	🏆 TOP REPORTER	AWARDED AFTER 10 COMPLAINTS
LEVEL 4	600 – 799	🦸 CITY HERO	AWARDED AFTER 25 COMPLAINTS
LEVEL 5+	800+	🔥 ECO WARRIOR	AWARDED WHEN POINTS ≥ 800

Each service attempt is triggered only after the system is already aware of the total accumulated points by the user, thus automatically promoting the user's status and badge level.

Image Upload

When a citizen submits a complaint with a photo:

Multer receives the image from the form

It's immediately uploaded to Cloudinary (a cloud image storage service)

The URL of the uploaded image is saved in MongoDB

This avoids storing heavy image files in the database directly

MIS Analytics

This is what makes CleanTrack an **MIS (Management Information System)** and not just a complaint app.

The admin dashboard shows:

Total complaints filed

Status breakdown – how many are Pending, In Progress, Resolved

Priority breakdown – how many are Low, Medium, High

Monthly trend – how complaint volume changes month by month

This data helps authorities make decisions like deploying more workers in busy months or identifying high-priority zones.

API Routes

These are the **URL endpoints** the frontend calls to communicate with the backend:

TABLE III API ROUTES

Route	Who Uses It	Purpose
POST /api/auth/register	Anyone	Create account
POST /api/auth/login	Anyone	Get JWT token
POST /api/complaints	Citizen	Submit complaint
GET /api/complaints/my	Citizen	View own complaints
GET /api/complaints	Admin	View all complaints
PUT /api/complaints/:id	Admin	Update status/priority
GET /api/analytics/dashboard	Admin	View statistics

Server Entry Point (server.js)

This is the **starting point** of the entire application:

express sets up the web server

mongoose. connect connects to MongoDB Atlas using the URI from .env



All route files are attached so the server knows which code to run for each URL

The server starts listening on port 5000

Environment Configuration (.env)

This file stores **sensitive information** that should never be shared publicly:

MONGO_URI – the address of your MongoDB database

JWT_SECRET – the secret key used to sign tokens

CLOUD_* – Cloundinary credentials for image uploads

These are stored in .env and emphasized using dotenv in order to prevent their display in code.

Deployment on Render

Render is a **free cloud hosting platform**. The steps simply mean:

Upload your code to GitHub

Tell Render where the code is

Tell it how to install and start the app

Add all your secret .env values on Render's dashboard

The system needs MongoDB Atlas to permit entry from all IP addresses because Render uses multiple IP addresses that change dynamically throughout the day.

VI. EXPERIMENTATION AND RESULT DISCUSSION

Experimentation:

The researchers conducted controlled experiments to test CleanTrack's performance and reliability at their local development facility. The system testing experiments evaluated all essential system components by simulating actual user behavior which included multiple users accessing the system and uploading photos and the system displaying information on real-time dashboards.

The experimentation process included the following steps:

The researchers deployed the Node.js application together with MongoDB which operated as a background service on their local server.

Creation of multiple citizen accounts to simulate real user registration and authentication flows.

Submission of complaint records with and without photographic attachments to validate Multer file handling and database persistence.

Testing of duplicate detection by submitting identical area-and-waste-type combinations within 48-hour windows.

Verification of smart priority assignment by submitting complaints with hospital and school address keywords.

Validation of gamification points and badge awards after sequential complaint submissions.

Testing of admin status update workflow from Pending to In Progress to Resolved, with verification of status history timeline recording.

TABLE IIIV EXPERIMENTAL SETUP FOR SYSTEM EVALUATION

Component	Specification	Purpose
Development Environment	Visual Studio Code	Source code editing and debugging
Backend Runtime	Node.js (v18+) with Express.js	Server-side logic, routing, and API processing
Database	MongoDB with Mongoose ODM	Persistent storage of users and complaints
File Upload Middleware	Multer (v1.4.5-lts)	Handles multipart photo upload and storage
Authentication	JWT + bcryptjs	Secure token-based user session management



Frontend Visualization	Chart.js (v4.4.1)	MIS dashboard charts and trend graphs
Testing Platform	Localhost Web Server (Port 3000)	Simulating concurrent user interactions and admin workflows
Client Interface	Web Browser (Chrome / Edge)	Access both citizen and admin portals during testing

Results Discussion:

TABLE V RESULT

Section	Details
1.1 User Registration and Authentication	Implemented JWT authentication with role-based access control for citizens and admins. Passwords encrypted using bcrypt. Token expiry set to 1 day. Unauthorized access to admin routes was successfully restricted.
1.2 Complaint Submission and Tracking	Users can submit complaints with title, description, location, and image. Images stored on Cloudinary with URLs in MongoDB. Status tracking (Pending → In Progress → Resolved) works properly. 10 points auto-awarded per complaint.
1.3 Admin Dashboard and Complaint Management	Admins can filter complaints by status and priority, update complaint stages, and assign priority levels. Upon resolution, users receive 20 bonus points. Proper role-based access control enforced.
1.4 Gamification System	Points, levels, and badges are automatically assigned based on activity. 0–199 → Level 1 (First Report) 200–399 → Level 2 (Active Citizen) 400–599 → Level 3 (Top Reporter) 600–799 → Level 4 (City Hero) 800+ → Level 5+ (Eco Warrior)
1.5 MIS Analytics	Dashboard powered by MongoDB aggregation showing live stats: Total Complaints: 142, Pending: 38, In Progress: 47, Resolved: 57, Priority (High/Medium/Low): 29 / 63 / 50. Monthly trends show peak during monsoon.
1.6 Deployment	Deployed on Render with MongoDB Atlas. API response time maintained under 300ms under normal load.

Discussion:

2.1 MIS Effectiveness

The analytics dashboard replaced manual reporting with data-driven insights, allowing authorities to anticipate peak periods and allocate resources proactively — a core advantage of MIS integration.

2.2 JWT Authentication

Stateless JWT ensured scalable, secure access. Role-based control successfully separated citizen and admin operations without session management overhead.

2.3 Cloud Scalability

MongoDB Atlas and Cloudinary eliminated infrastructure management, enabling the system to scale automatically for municipal or state-level deployment.

2.4 Gamification Impact

Points, levels, and badges addressed citizen disengagement by creating a visible reward loop. However, some users submitted redundant complaints to earn points, highlighting the need for a duplicate detection mechanism.

2.5 Complaint Lifecycle

The three-stage workflow gave citizens visibility at every stage and helped admins prioritize effectively, reducing the communication gap between the public and authorities.



2.6 Limitations

- No duplicate complaint detection
- No real-time notifications for status changes
- Location stored as text, not GPS coordinates
- Not tested under high concurrent load
- Priority assigned manually by admins

2.7 Future Enhancements

- Google Maps API for GPS tagging and complaint heat maps
- Push/email notifications via Firebase or Nodemailer
- AI-based NLP classifier for auto-priority assignment
- Duplicate detection algorithm for same-location complaints
- Load testing using Apache JMeter
- Leaderboards and community challenges in gamification

VII. SDG ALIGNMENT / SOCIAL IMPACT

CleanTrack supports multiple United Nations Sustainable Development Goals (SDGs) most notably through its support for SDG-11: Sustainable Cities and Communities and SDG-3: Good Health and Well-being. The system creates its social impact through the following description:

Alignment with SDG 11-- Sustainable cities and communities

CleanTrack system allows citizens to report waste problems to authorities which helps maintain clean urban areas that meet the sustainable development goal 11 requirement for inclusive urban development.

The system uses real-time complaint tracking to display reported issues which allows organizations to take responsibility while solving problems and it helps citizens understand how their complaints will be handled by municipal authorities.

Alignment with SDG-3: Good Health and Well-being

The system demonstrates its dedication to SDG-3 through his efforts to improve public health through his waste complaint resolution process which helps control disease outbreaks originating from uncollected garbage and open dumping and unsanitary conditions existing in urban areas. The use of monthly trend analytics by authorities enables them to identify areas experiencing high complaint volumes which allows them to send out sanitation workers for health hazard prevention before emergency situations develop.

VIII. CONCLUSION

The research demonstrates the CleanTrack system, which operates as a web-based Management Information System to improve urban waste complaint handling through better transparency and operational efficiency and managerial responsibility. The system uses secure JWT-based authentication together with structured complaint lifecycle management and real-time MIS analytics to handle sanitation problems and help authorities make decisions. The platform uses a three-tier system design with cloud-based infrastructure to allow citizens to report complaints while providing administrators with tools to track issues and manage their resolution. The system uses gamification elements to keep citizens involved in the process while MongoDB aggregation pipelines help with data-based governance. The system achieves both scalability and adaptability through its integration of Node.js and MongoDB Atlas and Cloudinary technologies which enable its use at municipal and state government levels. The system will benefit from future improvements that include AI-based complaint classification and GPS mapping and real-time notifications. CleanTrack uses technology to advance urban sanitation management through its smart and responsive system.



IX. ACKNOWLEDGMENT

The authors sincerely express their gratitude to their guide and mentor, Dr. Mrunal Pathak, for their invaluable guidance, encouragement, and continuous support throughout the development of this research work. The faculty members of the Department of Information Technology deserve our gratitude because they supplied the academic environment and resources which enabled us to conduct our study. The research work was successfully completed because of their support and motivation.

REFERENCES

- [1]. Kumar, R. Singh, and P. Sharma, "Web-Based Solid Waste Complaint Management System for Urban Areas," *International Journal of Smart City Applications*, vol. 5, no. 2, pp. 45–52, 2021.
- [2]. R. Sharma and K. Reddy, "Cloud-Based Scalable Architecture for Civic Service Platforms Using MongoDB Atlas," *Journal of Cloud Computing and Distributed Systems*, vol. 8, no. 1, pp. 23–35, 2021.
- [3]. P. Patil, S. Kulkarni, and A. Deshmukh, "Secure Authentication Using JSON Web Tokens in Civic Web Applications," *International Journal of Cybersecurity and Information Systems*, vol. 7, no. 3, pp. 60–68, 2022.
- [4]. R. Sharma and K. Reddy, "Role-Based Administrative Dashboards for Municipal Waste Monitoring Systems," *Journal of E-Governance and Smart Cities*, vol. 6, no. 2, pp. 88–97, 2022.
- [5]. S. Mehta and R. Singh, "MIS-Based Analytics and Decision Support in Smart City Sanitation Systems," *International Journal of Information Systems and Management*, vol. 10, no. 1, pp. 101–115, 2023.
- [6]. A. Verma, N. Gupta, and S. Jain, "Gamification Techniques to Enhance Citizen Participation in Urban Governance Platforms," *Journal of Human-Centered Computing*, vol. 9, no. 2, pp. 55–70, 2023.
- [7]. D. Joshi and V. Nair, "Design of Real-Time Civic Applications Using Node.js and Express.js," *International Journal of Web Engineering*, vol. 12, no. 4, pp. 140–152, 2023.
- [8]. T. Bhuiyan, Y. Xu, A. Jøsang, and C. Cox, "A Review on Citizen-Centric E-Governance and Complaint Management Systems," *Government Information Quarterly*, vol. 37, no. 3, 2020.
- [9]. S. Kumar and R. Tiwari, "NoSQL Databases for Scalable Web Applications: A Study on MongoDB," *International Journal of Database Systems*, vol. 13, no. 3, pp. 45–60, 2021.
- [10]. D. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, 2010.
- [11]. Auth0, "JSON Web Token (JWT) Introduction," [Online]. Available: <https://jwt.io/introduction>
- [12]. MongoDB Inc., "MongoDB Atlas Documentation," [Online]. Available: <https://www.mongodb.com/docs/atlas/>
- [13]. Express.js Foundation, "Express.js Documentation," [Online]. Available: <https://expressjs.com>
- [14]. M. Deterding et al., "From Game Design Elements to Gamefulness: Defining Gamification," *Proc. MindTrek Conference*, 2011.
- [15]. H. Hamari et al., "Does Gamification Work? A Literature Review," *Hawaii International Conference on System Sciences*, 2014

