International Research Journal of Education and Technology

Peer Reviewed Journal, **ISSN: 2581-7795**

INTELLIGENT ONLINE COMPLAINT MANAGEMENT SYSTEM

LINGESH KARTHIK D and R RESHMA

UG Student(III B.Sc. COMPUTER SCIENCE), Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore, India

Assistant Professor, Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore, India

Abstract:The **Intelligent Online Complaint Management System** is a web-based platform designed to modernize the process of registering and managing public complaints between citizens and government authorities. It replaces traditional manual complaint methods such as paper forms and phone calls with a centralized digital system. Citizens can register, submit complaints with supporting evidence like images or videos, and track their complaint status in real time. The system also provides an administrative dashboard for authorities to view, categorize, and resolve complaints efficiently. It is developed using the MERN Stack (MongoDB, Express.js, React, Node.js) for high performance and scalability. The platform is deployed using modern cloud services to ensure accessibility, transparency, and faster grievance resolution.

Keywords:
Complaint Management System,MERN Stack,Citizen Grievance Redressal,Web-Based Application,Real-Time Complaint Tracking,Cloud Deployment

.

1. INTRODUCTION

In today's rapidly evolving digital world, effective and transparent communication between citizens and government authorities has become a fundamental requirement for good governance and efficient civic management. Traditional methods of complaint filing—including paper forms submitted at government offices, telephone helplines, and personal visits—are increasingly inadequate for modern demands. These conventional approaches are time-consuming, lack transparency, and are inherently difficult to track or audit. Citizens are often left without meaningful feedback on whether their complaints have been received, reviewed, or acted upon. At the same time, administrators and municipal officers are burdened with managing large volumes of unstructured, handwritten, or verbally recorded complaints, making systematic analysis and prioritization nearly impossible.

To address these challenges, the proposed system provides a centralized digital platform that enables citizens to register complaints easily and allows authorities to categorize, prioritize, and respond to them efficiently. The platform ensures transparency and accountability throughout the complaint resolution process. Citizens can track the status of their complaints in real time, which

increases trust and engagement between the public and government authorities.

The system is developed using the **MERN Stack**, which includes **MongoDB, Express.js, React.js, and Node.js**, one of the most popular technology stacks for modern full-stack web application development. This architecture provides a unified JavaScript environment across both client and server layers, simplifying development, deployment, and maintenance. React.js is used on the frontend to create a dynamic and responsive user interface, while the Node.js and Express.js backend provides a fast and scalable API server.

MongoDB is used as a flexible NoSQL document database, which is well suited for storing various types of complaint data, including optional GPS coordinates, embedded media files, and user comments. This flexible structure allows the system to handle diverse types of complaints while maintaining efficient data management.

The deployed version of the system, **ComplainIQ**, is hosted on industry-standard cloud platforms—**Render** for the backend API server and **Vercel** for the frontend React application. This deployment architecture demonstrates real-world production readiness and ensures that the system remains accessible to users at all times.

Additionally, the platform is fully responsive and optimized for both desktop and mobile devices. This ensures accessibility for a wide range of users and promotes inclusivity across different user demographics and device types.

1.1 ABOUT THE PROJECT

The **Intelligent Online Complaint Management System** is a web-based platform developed to simplify and modernize the process of registering and resolving public complaints. In traditional systems, citizens often need to visit government offices, fill out paper forms, or make phone calls to report issues, which leads to delays, lack of transparency, and difficulty in tracking complaint status. To overcome these challenges, the proposed system provides a digital platform that enables citizens to register, log in securely, and submit complaints related to public services or civic issues. Users can attach supporting evidence such as images, videos, and location details, and they can track the progress of their complaints and receive updates on the actions taken by the authorities.

On the administrative side, the system offers a dedicated dashboard that allows officials to view, categorize, prioritize, and update the status of submitted complaints efficiently. The application is developed using the **MERN Stack** (MongoDB, Express.js, React, and Node.js), which supports a responsive user interface and scalable backend services. The system is deployed on modern cloud platforms such as **Render** for the backend and **Vercel** for the frontend, ensuring high availability and accessibility. Overall, the project demonstrates how modern web technologies can enhance transparency, accountability, and efficiency in public grievance management.

1.2 OBJECTIVES OF THE PROJECT

The primary objective of the **Intelligent Online Complaint Management System** is to provide a fast, transparent, and convenient digital platform for citizens to report civic issues and communicate with government authorities through a single user-friendly web interface. The system allows citizens to register securely, log in, and submit structured complaints that include a title, detailed description, category, GPS location, and optional media evidence such as images or videos. It also enables users to track the status of their complaints in real time through stages such as **Pending, In Progress, and Completed**, ensuring transparency and

accountability throughout the grievance redressal process.

Another important objective is to provide an efficient administrative environment where officials can manage complaints systematically. The system includes a feature-rich **Admin Dashboard** that supports searching, filtering by category or status, sorting by date or priority, and exporting complaint data. It also integrates modern technologies such as **Mapbox GL JS** for visualizing complaint locations on an interactive map and **OpenStreetMap Nominatim API** for reverse geocoding of GPS coordinates into readable addresses.

1.3 PROBLEM STATEMENT

In many existing complaint management processes used by municipal bodies, government departments, and educational institutions, citizens face several challenges when reporting civic issues and tracking their resolution. The absence of a centralized digital platform leads to inefficiencies, delays, and a lack of transparency throughout the complaint lifecycle. Citizens often have no visibility into the status of their complaints once they are submitted through paper forms or telephone helplines, and there is usually no feedback mechanism unless they personally follow up. Manual record keeping can result in data loss, duplicate entries, and slow processing. In addition, traditional systems lack geo-spatial capabilities, making it difficult for administrators to visualize complaint locations or identify areas with frequent issues.

The **Intelligent Online Complaint Management System** addresses these problems by providing a structured and transparent digital workflow. The system includes several key modules such as the **Authentication Module**, which manages secure user registration and login using encrypted passwords and JWT-based authentication; the **User/Complaint Module**, which allows citizens to submit complaints with descriptions, categories, GPS coordinates, and media evidence; and the **Admin Dashboard Module**, which enables administrators to view, filter, prioritize, and manage complaints efficiently. Additionally, the **Map Module** integrates map technology to display complaint locations visually and assist administrators in identifying and navigating to reported issues. Together, these modules create a modern platform that improves transparency, accountability, and efficiency in handling public grievances.

2. SYSTEM SPECIFICATION

2.1 Hardware Specification:

The system requires a minimum **Intel Core i3 processor**, **4 GB RAM**, and **256 GB HDD**, while the recommended configuration includes an **Intel Core i5 processor**, **8 GB RAM**, and **512 GB SSD** for better performance. A **64-bit operating system**, **keyboard and mouse**, **minimum display resolution of 1024×768**, and **internet connectivity of at least 10 Mbps** are required. The system can be accessed using modern browsers such as **Chrome, Firefox, or Microsoft Edge**.

2.2 Software Specification:

The system is developed using the **MERN Stack**, which includes **MongoDB Atlas** for database management, **Express.js** for backend development, **React.js** for building the frontend user interface, and **Node.js** as the server-side runtime environment. The application can run on **Windows, macOS, or Ubuntu operating systems** and is deployed on cloud platforms for reliable access and scalability.

3. SYSTEM STUDY

3.1 Existing System:

Most government departments, municipal bodies, and institutions still rely on traditional complaint handling methods such as paper-based forms, telephone helplines, emails, or

basic web forms. These systems are often slow, inefficient, and lack transparency. Citizens usually need to visit offices physically or make phone calls to report issues, and once the complaint is submitted, there is little or no information about its progress. Manual record keeping also increases the chances of data loss, duplicate entries, and delays in processing complaints.

These existing systems have several limitations. There is usually no role-based access control, meaning complaint records may not be securely managed. Citizens also do not receive real-time updates or acknowledgments after submitting complaints, which reduces trust in the system. Additionally, traditional methods do not support geographic mapping of complaints, multimedia evidence attachments, or structured communication between citizens and authorities,

3.2 Proposed System:

The proposed **Intelligent Online Complaint Management System** is designed to overcome these limitations by providing a centralized, web-based platform for managing complaints digitally. The system allows citizens to register, submit complaints with structured information such as title, description, and category, and optionally attach images, videos, or GPS location details. Each complaint follows a clear workflow with status stages such as **Pending, In Progress, and Completed**, allowing users to track the progress of their complaints in real time.

The system also provides an **Admin Dashboard** where administrators can view, categorize, filter, and respond to complaints efficiently. Role-based access control secured through JWT authentication ensures that only authorized administrators can manage complaint records. Additional features such as map-based visualization, multimedia evidence support, and data export options improve transparency, accountability, and decision-making. Overall, the proposed system provides a modern, efficient, and user-friendly solution for managing public complaints.

4. SYSTEM DESIGN

4.1 Input Design:

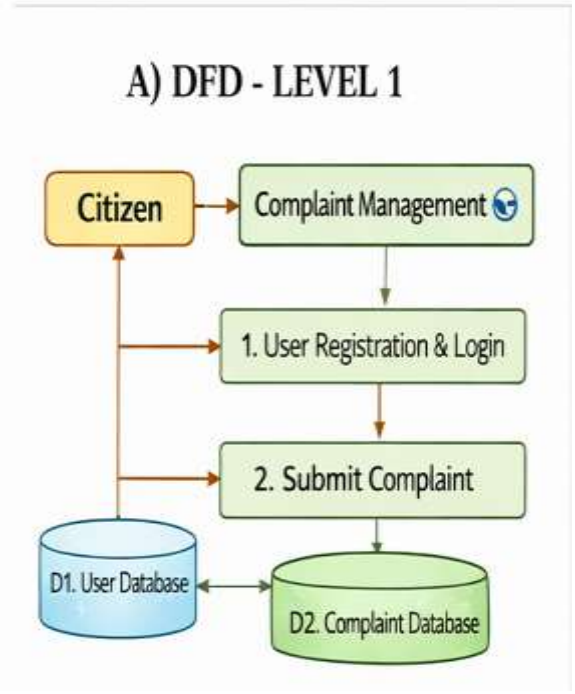
Input design in the **Intelligent Online Complaint Management System** focuses on collecting data from citizens and administrators in a structured and user-friendly manner. The system provides well-organized forms for user registration, login, complaint submission, and admin replies. Each form includes clear labels, placeholder text, and validation checks to guide users and reduce data entry errors. Client-side validation is implemented using React form components, while server-side validation is handled through Express.js and Mongoose to ensure data accuracy and security.

4.2 Output Design:

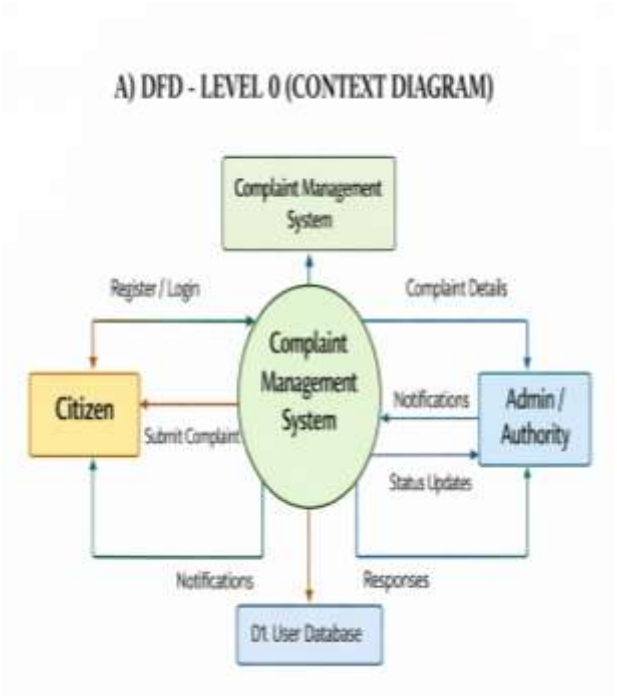
Output design ensures that processed information is displayed in a clear and meaningful format for both citizens and administrators. Citizens can view complaints through an interactive complaint feed with status indicators such as **Pending, In Progress, and Completed**, along with notifications confirming their actions. Administrators access outputs through the **Admin Dashboard**, which displays complaint statistics, a filterable complaint list, map-based location views, and downloadable CSV reports. These outputs help users easily monitor complaint status and assist administrators in efficient complaint management and reporting.

4.3 **A) DFD – Level 0 (Context Diagram):**

The Level 0 Data Flow Diagram represents the overall view of the Complaint Management System. It shows how external entities like the **Citizen** and **Admin/Authority** interact with the system. Citizens can register or log in and submit complaints, which are processed by the Complaint Management System. The system stores user information in the **User Database**



and sends complaint details to the Admin. The Admin reviews the complaint, provides responses, and updates the status. Notifications and status updates are then sent back to the citizen through the system.

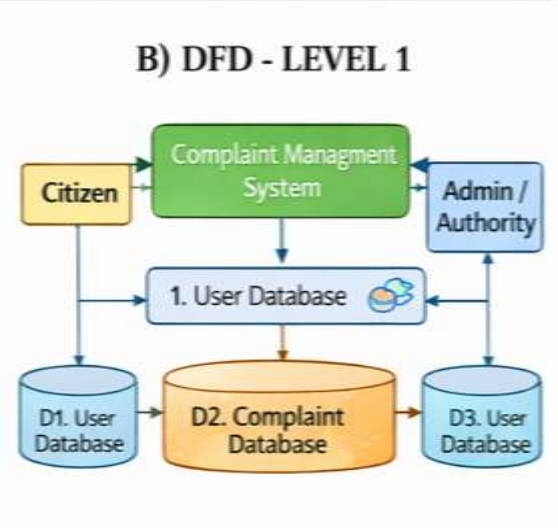


B) DFD – Level 1

The Level 1 DFD provides a more detailed breakdown of the Complaint Management System processes. It divides the system mainly into **User Registration & Login** and **Submit Complaint**. Citizens first register or log in to the system, after which they can submit their complaints. The system stores user data in the **User Database (D1)** and complaint information in the **Complaint Database (D2)**. The Admin/Authority can access these complaints, review them, and manage the complaint process using the stored data.

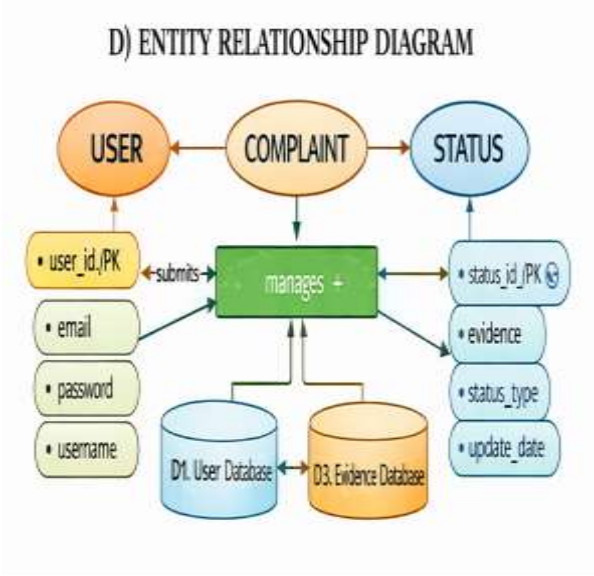
C) DFD – Level 2

The Level 2 DFD further explains the **Submit Complaint** process in detail. First, the citizen enters complaint details through the system (**2.1 Enter Complaint Details**). Then the user uploads supporting files or proof such as images or documents (**2.2 Upload Evidence**). These files are stored in the **Evidence Database (D3)**. The Admin/Authority can then view the complaint, categorize it, respond to it, and update its status based on the evidence and details provided.



D) Entity Relationship Diagram (ER Diagram)

The ER Diagram shows the database structure of the Complaint Management System and how different entities are related. The main entities are **User**, **Complaint**, and **Status**. The User entity contains attributes like user ID, email, and password. A user can submit complaints, and each complaint is managed within the system. The Status entity keeps track of the complaint's progress, including status type and update date. These entities are connected through relationships that represent how users submit complaints and how the system manage.



5. TESTING

1. Testing Overview

Testing is an essential phase in the software development lifecycle of the Intelligent Online Complaint Management System. A comprehensive testing strategy was implemented to ensure that the system functions correctly, securely, and reliably in real-world conditions. The process included unit testing, integration testing, validation testing, white box testing, black box testing, user acceptance testing, and performance testing to verify all modules and system interactions.

2. Unit and Integration Testing

Unit testing was performed on individual modules such as the authentication module and complaint module to ensure each function works correctly in isolation. Features like user registration, login verification, token generation, complaint creation, and status updates were thoroughly tested. Integration testing was then conducted to confirm that the frontend and backend components interact properly. Scenarios like register → login → dashboard, complaint submission → public feed display, and admin updates → user status view were successfully verified.

3. Validation and Security Testing

Validation testing ensured that the entire system met both functional and non-functional requirements. Security measures were also tested, including protection against JWT token forgery, NoSQL injection attacks, and cross-site scripting. Passwords were securely stored using bcrypt hashing, and role-based access control ensured that only authorized users could access protected system features.

4. White Box and Black Box Testing

White box testing was applied to analyze the internal structure of the backend code, including Express.js routes, middleware functions, and database validation logic, ensuring all logical paths and error conditions work properly. Black box testing focused on the system's functionality from the user perspective by testing the frontend interface, including registration, login, complaint submission, profile management, and admin dashboard operations.

5. User Acceptance and Performance Testing

User Acceptance Testing (UAT) was conducted with eight participants including citizens, administrators, and technical staff. Participants completed real-world tasks such as submitting complaints, uploading evidence, and updating complaint status. The system received an average satisfaction score of **4.5/5**. Performance testing also confirmed efficient

response times, with login requests under **350 ms**, complaint retrieval under **700 ms**, and complaint creation under **400 ms**, demonstrating that the system performs efficiently under typical usage conditions.

6. SYSTEM IMPLEMENTATION AND MAINTENANCE

1. System Implementation Overview

System implementation is the final phase of the software development lifecycle where the Intelligent Online Complaint Management System is deployed into a live environment for real users. This stage involved configuring cloud platforms, deploying frontend and backend applications, connecting the system to the MongoDB Atlas database, setting up environment variables and API keys, and performing final end-to-end testing to ensure the system works correctly in the production environment.

2. Frontend Implementation

The frontend of the system was developed using React.js and bootstrapped with Vite, which provides fast development with features like Hot Module Replacement and optimized production builds. The frontend application was deployed on Vercel, a serverless hosting platform that offers global content delivery, automatic HTTPS security, and seamless deployment directly from the project's Git repository.

3. Backend Implementation

The backend API was built using Node.js and Express.js with a structured architecture. The main server file (server.js) manages API requests, while database models are organized in the models directory and authentication logic is handled through JWT middleware. The backend was deployed on Render, a cloud platform that supports Node.js applications with automatic deployment, environment variable configuration, HTTPS security, and health monitoring.

4. Database Implementation

The system uses MongoDB Atlas as its cloud database. A free-tier cluster was created in the AWS Mumbai region to reduce latency for South Asian users. The database connection is secured through IP allow-listing and authentication credentials stored as environment variables. Mongoose schema validation is used to ensure proper data structure and maintain data integrity.

5. Media Storage and Data Handling

All complaint-related media files such as images and videos are stored directly within MongoDB documents as base64-encoded data. This approach simplifies the system architecture by eliminating the need for external storage services like cloud file storage platforms. As a result, the system remains lightweight while still supporting evidence uploads as part of the complaint submission process.

7. CONCLUSION

The **Intelligent Online Complaint Management System (ComplainIQ)** was successfully designed, developed, tested, and deployed as a full-stack web application that digitalizes the entire process of citizen complaint management. The project aimed to replace traditional complaint systems that are often slow, unorganized, and lacking transparency with a modern digital platform that improves efficiency and accountability. Through this system, citizens can easily submit complaints online while authorities can manage and resolve them more effectively.

The system provides **secure authentication using JWT**, allowing both citizens and administrators to access the platform according to their roles. Citizens can submit detailed complaints along with multimedia evidence and GPS location data, making complaints more reliable and verifiable. The system also implements a **three-stage complaint workflow — Pending, In Progress, and Completed**, which allows

users to track the progress of their complaints and increases transparency in the complaint resolution process.

From a technical standpoint, the system was developed using the **MERN stack (MongoDB, Express.js, React.js, and Node.js)**, which proved to be efficient for building a scalable and responsive web application. These technologies allowed seamless integration between the frontend, backend, and database while maintaining good performance and flexibility for future improvements.

Overall, the Intelligent Online Complaint Management System demonstrates how modern open-source technologies can solve real-world civic problems by making governance systems more transparent and efficient. The platform has the potential to strengthen the relationship between citizens and authorities by improving communication, accountability, and trust.

In the future, the system can be enhanced with advanced features such as **AI-based complaint classification, push notifications, mobile applications, and service-level agreement (SLA) monitoring**. These improvements could further expand the system's capabilities and make it a powerful tool for smart city governance and digital public service management.

8. FUTURE ENHANCEMENT

The **Intelligent Online Complaint Management System** has been developed using a modular and scalable architecture, allowing it to support many future improvements. Although the current system successfully achieves its main objectives and has passed extensive testing and user acceptance evaluations, there is still strong potential to enhance its intelligence, usability, and governance capabilities. Future developments will focus on integrating advanced technologies and new features to further improve the efficiency and effectiveness of the platform.

One major future enhancement is the integration of **Artificial Intelligence for automatic complaint classification**. Using Natural Language Processing (NLP) models such as BERT or DistilBERT, the system can automatically analyze the text of a complaint and assign it to the appropriate category. This will reduce the manual work required by administrators and improve the accuracy of complaint routing. Additionally, computer vision technologies such as Google Vision API or AWS Rekognition could analyze uploaded images to identify issues like potholes, garbage dumping, or damaged infrastructure from visual evidence.

9. REFERENCE

1. Books

• Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.
• Mardan, A. (2018). *React Quickly: Painless Web Apps with React, JSX, Redux, and GraphQL*. Manning Publications.
• Hossain, M. (2019). *Full-Stack React Projects*. Packt Publishing.
• Brown, E. (2019). *Web Development with Node and Express: Leveraging the JavaScript Stack* (2nd ed.). O'Reilly Media.

2. Research Papers / Journals

• Bhatt, R. K. (2021). Modern Web Application Development with MERN Stack. *International Journal of Computer Science and Engineering*, 9(3), 45–52.
• Srivastava, P., & Mehta, A. (2022). Civic Complaint Management Using Digital Web Technologies. *Journal of Information Technology Research,* 15(2), 118–130.
• Kumar, S., & Rao, P. (2020). Role of Web Technologies in E-Governance and Citizen Services. *International Journal of Advanced Computer Science and Applications*, 11(4), 78–85.

3. Web References

• MongoDB Inc. (2024). *MongoDB Manual –*

Document Model. https://docs.mongodb.com/manual/

• React Documentation. (2024). *Getting Started with React*. https://react.dev/

• Express.js Foundation. (2024). *Express.js Guide and API Reference*. https://expressjs.com/

• Auth0. (2024). *Introduction to JSON Web Tokens*. https://jwt.io/introduction/

• Mapbox. (2024). *Mapbox GL JS API Reference*. https://docs.mapbox.com/mapbox-gl-js/api/

• OpenStreetMap Nominatim. (2024). *Reverse Geocoding API*. https://nominatim.org/release-docs/

• Node.js Foundation. (2024). *Node.js v18 API Documentation*. https://nodejs.org/en/docs/

• Vercel Inc. (2024). *Vercel Platform Documentation*. https://vercel.com/docs/

• Render Inc. (2024). *Render Web Services Deployment Guide*. https://render.com/docs/

• Tailwind Labs. (2024). *Tailwind CSS Utility-First Fundamentals*. https://tailwindcss.com/docs/

If you want, I can also **convert this into a perfect IEEE / APA reference format** (some colleges specifically require that for project reports).