MADURAI SMART CIVIC PORTAL: Ward Based Complaint System

A PROJECT REPORT

Submitted by

BALASUBRAMANIAN S (21H062)

VISHNUPRASAD V (21H068)

YOGESHRAJ V R (21H069)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND BUSINESS SYSTEMS

THIAGARAJAR COLLEGE OF ENGINEERING,

MADURAI – 15.

(A government Aided Autonomous Institution Affiliated to Anna University)

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI – 15

(A Government Aided Autonomous Institution Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report **"Madurai Smart Civic Portal: Madurai Ward Based Complaint System"** is the bonafide work of "**Balasubramanian S (21H062), Vishnuprasad V (21H068), Yogeshraj V R (21H069)**" who carried out the project work under my supervision during the Academic Year 2024-2025.

SIGNATURE

Dr. P. CHITRA, M.E., Ph.D.,

PROFESSOR &

HEAD OF THE DEPARTMENT

COMPUTER SCIENCE AND BUSINESS SYSTEMS

THIAGARAJAR COLLEGE OF ENGG. MADURAI – 625 015

SIGNATURE

Mr. V JANAKIRAMAN, M.E.,(Ph.D.),

SUPERVISOR &

ASSISTANT PROFESSOR

COMPUTER SCIENCE AND BUSINESS SYSTEMS

THIAGARAJAR COLLEGE OF ENGG. MADURAI – 625 015

Submitted for the VIVA VOCE Examination held at Thiagarajar College of Engineering on …….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We wish to record our deep sense of gratitude and profound thanks to our project guide **Mr. V. Janakiraman**, Assistant Professor, Department of Computer Science and Business Systems, Thiagarajar College of Engineering, Madurai, for the keen interest, inspiring guidance, constant encouragement during all stages of our work, to bring this work into fruition.

We are extremely grateful to **Dr. P. Chitra**, Professor and Head, Department of Computer Science and Business Systems, Thiagarajar College of Engineering, Madurai, for her prodigious guidance and encouragement throughout our project work. We are extremely obliged to the review panel members for guiding us with their valuable suggestions and support during the course of our project work.

We are thankful to our college chairman and correspondent, **Mr. K. Hari Thiagarajan** and Principal **Dr. L. Ashok Kumar** for providing excellent and complete facilities to carry out our project work in the department. We also thank all the faculty members of our department and non-teaching staff for their valuable support throughout the course of our project work.

We owe the completion of this project work to my parents / industry for their patience and forbearance during our project work.

BALASUBRAMANIN S (21H062)

VISHNUPRASAD V (21H068)

YOGESHRAJ V R (21H069)

ABSTRACT

Urban areas face frequent civic issues like potholes, drainage blockages, faulty streetlights, and power outages. Traditional complaint systems are manual, inefficient, and discourage participation due to unclear processes and lack of transparency. This project presents a Ward-Based Complaint Registration and Tracking System, a web platform that allows citizens to easily register complaints and automatically routes them to the correct authority using GPS-based location detection. By identifying the user's ward via their coordinates, the system eliminates manual address input and ensures accurate complaint handling.

Built on Firebase, the application supports secure authentication, real-time complaint storage, and role-based access for ward officials, who can view only complaints from their jurisdiction. A real-time tracking feature lets users monitor their complaint's status, increasing transparency and accountability. The system is scalable, device-friendly, and promotes faster, more efficient resolutions through digital coordination. By combining geolocation, real-time data management, and user-friendly design, it offers a modern solution to improve civic engagement, urban governance, and quality of life.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**Iv**
	LIST OF TABLES	**viii**
	LIST OF FIGURES	**viii**
	LIST OF ABBREVIATIONS	**ix**
1.	**INTRODUCTION**	**10**
2.	**BENCHMARKING APPLICATIONS**	**12**
	2.1 Table Description	13
3.	**REQUIREMENT ANALYSIS AND SPECIFICATION**	**14**
	3.1 Hardware support	14
	3.2 Software requirements	14
	3.3 Tools	14
	3.4 Technology Used	15
4.	**PROBLEM DEFINITION AND BACKGROUND**	**16**
	4.1 Need for the Problem	**16**
	4.2 Problem Statement	**16**

5.	**PROPOSED METHODOLOGY**	**17**
	5.1 System Architecture	**17**
	5.2 Frontend Devolopment	**18**
	5.3 APIs and Services used in System	**19**
	5.4 Backend Devolopment(Data processing and admin Dashboard)	**19**
	5.5 Personalized Complaint Tracking	**20**
	5.6 Data visualization and Interaction	**20**
	5.7 Error handling and Data Integrity	**21**
	5.8 System Integration	**21**
	5.9 System Deployment	**21**
6.	**IMPLEMENTATION**	**22**
	6.1 Frontend Implementation	**22**
	6.2 Backend Implementation	**23**
	6.3 Location Detection and Regional Mapping	**23**
	6.4 Realtime datasync and Asynchronous processing.	**24**
	6.5 Security and Privacy	**24**
	6.6 Features proposed by the system	**24**
7.	**RESULTS AND DISCUSSIONS**	**27**
	7.1 Algorithm Evaluation and Selection	27
	7.2 Complaint Regitration Form	28
	7.3 Location Based Complaint Redirection	29
	7.4 Admin Dashboard Feature	30

	7.5 User Dashboard and Complaint Tracking	33
	7.6 User Authentication and Profile Management	34
	7.7 Notification and User Alert	35
8.	**CONCLUSION AND FUTURE WORK**	**37**
	8.1 Conclusion	37
	8.2 Future Work	38
9.	**REFERENCES**	**39**
10.	**PLAGARISM REPORT**	**41**

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
1	Existing products	10

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
6.1	System Architecture of CivicPortal	22
6.2.1	Complaint Registration Page	29
7.3.1	Location Accessing Page	30
7.3.2	Location Detection and Ward Identify	30
7.4.1	Admin Ward Selection during Regiter	31
7.4.2	Admin Dashboard-Complaint Management	32
7.5.1	User Complaint Ststus Trscking	34
7.6.1	Role Based Authentication	35
7.6.2	User and Admin Login page	35
7.7.1	Success and Error Alert	36

LIST OF ABBREVIATIONS

1.	GPS	Global positioning system
2.	UID	User identification
3.	UI	User Interface
4.	API	Application programming interface
5.	CRUD	Create,Read,Update,Delete
6.	HTML	Hypertext Markup language

CHAPTER 1

INTRODUCTION

Urban citizens frequently encounter a wide variety of civic issues in their day-to-day lives. Common problems such as potholes on roads, blocked drainage systems, power failures, non-functional streetlights, broken footpaths, water supply interruptions, and garbage accumulation directly affect the quality of life in cities. These issues, although often small and localized, demand timely identification and resolution to ensure that urban spaces remain safe, hygienic, and functional for residents. Despite advancements in technology and infrastructure, one of the most pressing challenges in modern urban governance is the inefficiency of traditional complaint redressal mechanisms.

In most urban regions, the complaint registration and resolution process continues to be largely manual and fragmented. Citizens are frequently required to visit municipal offices physically, fill out paper forms, or make several phone calls to determine which department or official is responsible for addressing their grievances. This approach is not only time-consuming but also creates confusion and discourages civic participation, as many residents are unaware of the complex administrative structures that govern various municipal services. The lack of transparency, inefficient tracking, and inadequate communication between citizens and local authorities often results in unresolved complaints and a general sense of dissatisfaction with public services.

To address these shortcomings, this project introduces the **Ward-Based Complaint Registration and Tracking System**, an innovative web-based platform specifically designed to streamline and modernize the process of civic complaint management. The primary objective of this system is to simplify

complaint submission and ensure faster redressal by integrating advanced technologies such as **GPS-based location detection** and **real-time database management**. Through automatic identification of the user's ward based on their geographic location, the application intelligently routes the complaint to the relevant local authority. This eliminates the cumbersome need for users to manually input their address details or search for the correct department to approach, thereby reducing friction and saving valuable time for both citizens and administrators.

The backbone of the application is **Firebase**, a robust cloud-based platform that offers secure and scalable solutions for modern web applications. Firebase is utilized to manage **user authentication**, ensuring that both citizens and ward officials can securely access the platform with role-based privileges. Citizens can easily create accounts, log in, and submit complaints, while ward officials are given access exclusively to complaints pertaining to their assigned regions. This role-specific access control system ensures that complaints are organized, properly categorized, and promptly handled by the designated administrative body. Furthermore, the **real-time database management** offered by Firebase ensures instant data storage and retrieval, allowing for seamless user interactions and fast response times.

One of the standout features of the Ward-Based Complaint Registration and Tracking System is its **real-time complaint tracking** mechanism. Once a citizen submits a complaint, they can continuously monitor its status at every stage — from submission, acknowledgment, investigation, to final resolution. This transparent process not only builds trust between citizens and local authorities but also enhances accountability among ward officials.

CHAPTER 2

BENCHMARKING APPLICATIONS

The table 1 highlights gaps in existing platforms, emphasizing the need for a ward-specific, streamlined complaint management system.

Table 1 – Existing Products

Module Description	State-of-the-Art Solution Available	Drawbacks of the Solution
Complaint Registration	**Swachhata App:** Allows citizens to report sanitation issues via mobile app	Limited to sanitation not suitable for broader range of civic range of issues like road issues.
Location – Based routing	**Smart City portal:** Use partial location data to forward complaints.	Location routing often not automated or ward specific may require manual filtering.
User Authentication	**My gov portal:** Centralized login for submitting various government grievances	No support for role based access, not customized for ward officials
Ward level admin panel	**Municipal websites:** Admin access and process complaint via backend systems	Lack of complaint segregation based on location and ward.
Realtime Tracking	**CM helpline:** Accept multiple issue categories like electricity, water.	Complaint types often routed manually, lack of structured categorization.

2.1 TABLE DESCRIPTION

The table 1 provides a comparative analysis between the key modules of the proposed Ward-Based Complaint Registration and Tracking System and the state-of-the-art solutions currently available in similar platforms. It highlights existing applications and portals such as Swachhata App, Smart City portal, MyGov portal, and CM Helpline, identifying the primary features they offer in areas like complaint registration, location-based routing, user authentication, administrative access, and real-time tracking. Furthermore, it outlines the specific drawbacks and limitations of each of these existing solutions—such as limited issue coverage, lack of automated ward-level routing, absence of role-based access, and insufficient complaint categorization. This comparison clearly demonstrates the gaps that the proposed system aims to address by offering a more comprehensive, location-specific, and streamlined platform for civic complaint management.

CHAPTER 3

REQUIREMENT ANALYSIS AND SPECIFICATION

3.1.Hardware Support

This project has the following minimum hardware requirements.

- **Device Type:** Android Smartphone, Tablet, or Desktop Browser-supported Device
- **Processor:** ARM-based processor (Snapdragon 600 and above recommended) or Intel equivalent for browsers
- **RAM**: Minimum 2 GB (4 GB or higher recommended for best performance)
- **Storage:** Minimum 100 MB free storage (for browser cache/app data)

3.2 Software Requirements

- **Operating System:** Android 8.0 (Oreo) and above, or any modern browser (Chrome, Firefox, Edge)
- **Google Play Services**: Required (if using mobile browser location services)
- **Internet Connectivity:** Required

3.3.Tools Used

- Visual Studio Code.
- Firebase Database Console (for Database storage and Authentication)
- Google Cloud Console(for Maps and Geolocation API s)

3.4 Technology Used

- **HTML, CSS, Javascript -** Used for building the web-based frontend of the Civic Portal, allowing users to register complaints, detect location, and track complaint status.
- **Firebase Firestore (Cloud Database)** – A scalable NoSQL cloud database used to store complaint submissions, user details, and authority information for complaint redirection
- **Firebase Authentication** - Provides secure login and user authentication functionality for citizens and ward authorities (if login feature is implemented).
- **Open Street Map API** – Used to detect users' GPS location, perform reverse geocoding, and map user coordinates to specific wards or authorities.
- **Javascript** – The primary programming language for frontend logic, including form handling, API integration, and dynamic content rendering.
- **Git** – A distributed version control system used for tracking code changes and collaborating on development.
- **Visual Studio Code** – The primary code editor used for developing both frontend and backend components efficiently.

CHAPTER 4

PROBLEM DEFINITION AND BACKGROUND

4.1 NEED FOR THE PROBLEM

Urban and semi-urban communities frequently encounter civic problems such as road damage, drainage issues, and power outages, but existing grievance redressal systems are often inefficient, requiring manual input and lacking real-time updates. These outdated methods result in delayed responses and a lack of accountability, diminishing public trust in local governance. A modern, automated platform is needed to streamline complaint registration and resolution using geolocation, enabling faster, more transparent service delivery and improving citizen engagement.

4.2 PROBLEM STATEMENT

Current platforms for civic complaint registration do not support automatic location-based routing or real-time complaint tracking, leading to inefficiencies in addressing local issues. This project proposes a web-based ward-level complaint management system that uses Firebase and geolocation services to automatically route complaints to the appropriate officials. With role-based dashboards and real-time tracking, the platform enhances transparency, accountability, and responsiveness in urban governance.

CHAPTER 5

PROPOSED METHODOLOGY

5.1 System Architecture

The Civic Portal system helps citizens report and track local civic issues efficiently. It integrates a user-friendly frontend with Firebase services for authentication and complaint management. Complaints are filtered by locality using geolocation APIs and directed to the respective councillors. Admins can view, update, and filter complaints through a dedicated panel. The system ensures transparent communication and faster issue resolution in Madurai. The architecture is shown in Figure 5.1.1

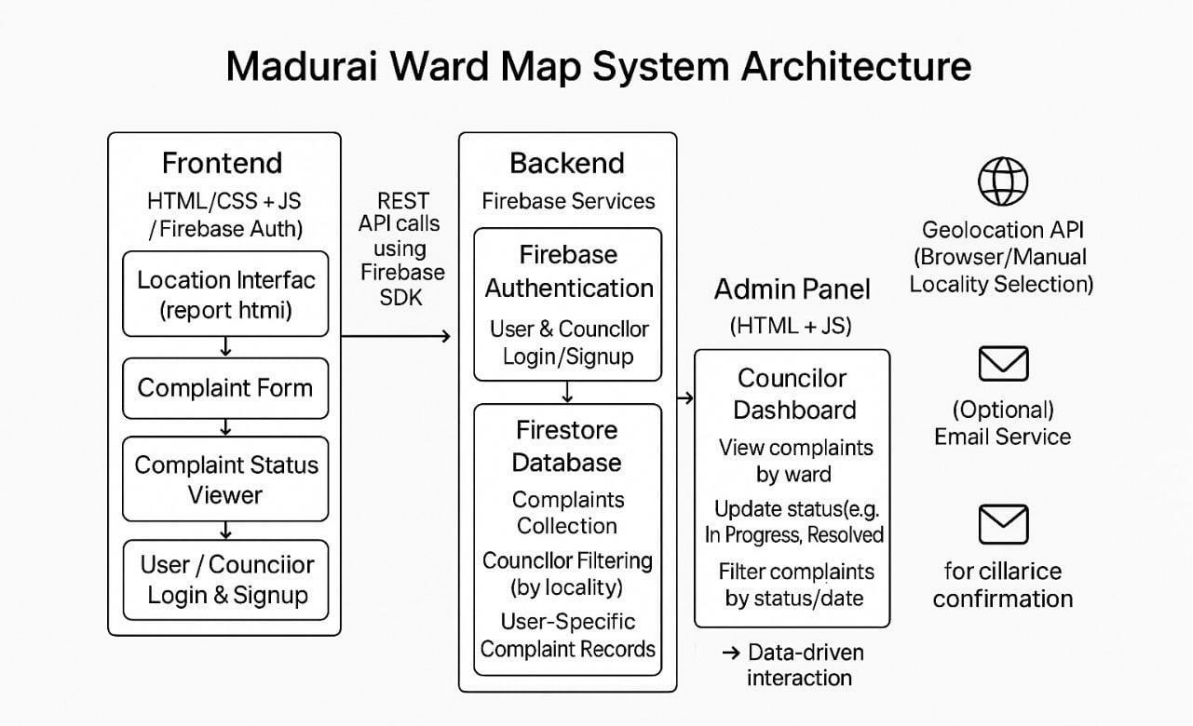

Figure 5.1.1 System Architecture of Civic portal

5.2 Frontend Development

- **Tech Stack**

  The frontend is developed using React.js for creating a responsive and dynamic web-based user interface. The design is optimized for mobile and desktop browsers to ensure accessibility without the need for app downloads.

- **Complaint Registration Interface**

  Users can file complaints easily through a clean and intuitive form. Fields include issue type (selected via dropdown), brief description, location (auto-detected or manually entered), and optional photo upload.

- **Location Auto-Detection**

  Through browser-based geolocation APIs, the user's latitude and longitude are retrieved automatically. The system maps these coordinates to the corresponding ward without requiring users to manually search for their locality.

- **Dynamic User Feedback**

  Loading indicators and confirmation pop-ups are incorporated to improve user experience during complaint submission and status tracking.

- **Handling User Authentication**

  Firebase Authentication is used to manage user and admin login/signup securely, with separate access flows for general users and administrative users.

5.3 APIs And Services Used in System

- **Firebase services**

  Firebase provides the backend-as-a-service infrastructure necessary for real-time operations and secure data handling.

- **Firebase Realtime Database**

  Stores complaint details including user input, location, issue category, status updates, and admin comments. Instantly updates the data for both users and admins without requiring page reloads.

- **Firebase Authentication**

  Handles secure registration and login for both users and administrators. Distinguishes roles based on email domain or predefined metadata to allocate proper access rights (e.g., normal user vs. ward admin).

- **Geolocation API**

  Browser’s built-in Geolocation API is utilized to capture the user's latitude and longitude. This data is used to map the user’s location to the corresponding administrative ward.

5.4 Backend Development (Data Processing and Admin Dashboard)

- **Tech Stack**

  Firebase Functions (optional for future expansion) for serverless backend logic. Real-time data flow between frontend and Firebase Database**.**

- **Complaint Routing Logic**

  User complaints are matched to the appropriate ward based on detected coordinates. Admins only view complaints relevant to their jurisdiction, preventing data clutter.

- **Admin Actions**

  Admins can change complaint status (e.g., Received, In Progress, Resolved) and add comments.

5.5 Personalized Complaint Tracking

- **Real-Time Status Tracking**

  After filing, users receive a unique Complaint ID. Users can input this ID into a tracking page to view live status updates, admin remarks, and expected resolution timelines.

- **Email Notification System (Future Scope)**

  Future integration can enable optional email notifications upon status changes.

5.6 Data visualization and Interaction

- **Complaint Mapping**

  A map view shows complaint locations as markers. Users and admins can visually assess problem density in particular areas for better planning and intervention.

- **Rapid Alert System**

  Admins can send urgent messages (e.g., road closures, water supply issues) as broadcast alerts. Alerts are displayed prominently to all users in the affected ward immediately upon login or via live banners.

5.7 Error Handling and Data Integrity

- **Error States**

  Geolocation failures, incomplete form submissions, or authentication errors are handled with user-friendly messages and guidance.

- **Input Validation**

  Complaint forms are validated for required fields before submission to maintain data quality and prevent database corruption.

5.8 System Integration

- **Integration**

  Seamless real-time integration between the React frontend and Firebase backend ensures instant data updates and user feedback.

5.9 System Deployment

- **Deployment**

  The system will be deployed as a web application accessible via standard browsers, ensuring easy access without the need for additional installations. Firebase's cloud infrastructure ensures scalability, reliability, and real-time performance during deployment.

CHAPTER 6

IMPLEMENTATION

6.1 Frontend Implementation(HTML, CSS, JAVASCRIPT)

The frontend is a web application built using traditional web technologies — HTML, CSS, and vanilla JavaScript — to ensure broad accessibility across browsers and devices without the need for framework-specific dependencies.

- **User Interface**

  Users can submit complaints using an interactive form that captures:

  - Location details (manual input; GPS detection planned for future).
  - Type of issue (dropdown or text input — e.g., road damage, drainage, etc.).
  - Description of complaint and optional media (image upload).
- **Authentication Interface**

  A login/register form powered by Firebase Authentication allows:

  - Citizens to create accounts and submit complaints.
  - Admins to log in and view complaints relevant to their locality.
- **Realtime Complaint Tracking**
  - View status updates of their complaints (pending, in-progress, resolved).
  - Track history of previously submitted complaints.
- **Dynamic UI updates**
  - Submit and retrieve data from Firestore.
  - Update complaint lists and status in real-time without reloading the page.
- **Error Handling**
  - Form inputs are incomplete.
  - Network or server errors occur.
  - No complaints found in the user’s locality.

6.2 Backend Implementation(Firebase Firestore and Firebase Authentication)

The backend services are entirely powered by Firebase, eliminating the need for traditional server management. Firebase provides real-time database capabilities, secure user authentication, and fine-grained data access control.

- **Firestore Database(NoSQL)**
    - Complaints are stored in a structured collection with fields like:
    - userID, complaintType, location, description, timestamp, status, adminAssigned.
    - Each document represents one complaint, allowing fast reads and writes.
- **User Authentication (Firebase Auth)**
    - Email/password authentication is used to differentiate between
    - Regular users (citizens).
    - Admin users (ward officials) — their email IDs are pre-mapped to localities.
- **Database Security Rules**
    - Users can view and edit only their complaints.
    - Admins can view and update only complaints within their jurisdiction.

6.3 Location Detection and Regional Mapping

To enhance user experience and automate locality assignment, the system is being extended with GPS-based location detection:

- **Geolocation API**
    - Automatically fetches user's latitude and longitude when submitting a complaint.
    - Reduces manual input errors and ensures correct mapping to a ward.

- **Region Division Logic:**
    - The city or town is divided into wards based on lat-long boundaries.

- The user’s coordinates will be used to auto-assign the complaint to the respective admin.

6.4 Realtime Data Sync and Asynchronous Processing

- **Frontend**
  - Asynchronous calls to Firestore and Authentication services ensure non-blocking UI.
  - Live updates to complaint lists using Firestore real-time listeners.
- **Backend**
  - Any updates made by admins (e.g., complaint status change) instantly reflect in user dashboards.

6.5 Security and Privacy

- **Firebase Authentication rules**
  - Ensures only authenticated users can submit and view their complaints.
- **Firestore Database rules**
  - Fine-grained permissions restrict read/write access to appropriate users and admins.
- **Data Validation**
  - JavaScript functions validate user input on the frontend to prevent malformed or malicious submissions.

6.6 Features Proposed by the System

- **GPS-Based Ward Detection**
  - The system uses the user's live location (latitude and longitude) to identify the exact ward or locality automatically.
  - Eliminates the need for users to manually enter their address or search for the relevant ward.

- Ensures complaints are routed to the correct local authority based on accurate geographic mapping.

- **Smart Complaint Registration Form**
  - Users can submit complaints through a clean and intuitive form interface.
  - Complaint categories include road damage, drainage issues, power outages, and more.
  - The form collects user inputs like category, description, optional image uploads, and auto-detected location.
- **Firebase Integration for Real-Time Data Storage**
  - Complaint data is stored in Firebase Firestore for instant read/write access.
  - Firebase Authentication secures both user and admin logins, supporting password-based sign-in.
- **User Login and Complaint History**
  - Registered users can log in to track their complaint history.
  - Each user has access to a dashboard that displays previously raised complaints along with their current status.
  - Users receive real-time updates when authorities take action or resolve complaints.
- **Admin Dashboard with Ward Filtering**
  - Each ward admin can log in with a verified email ID linked to a specific locality.
  - Admin dashboard only displays complaints from the corresponding ward, improving focus and response efficiency.

- Admins can update the status of complaints (e.g., “In Progress,” “Resolved”) and respond to users with remarks.

- **Complaint Tracking System**
  - After submission, each complaint is assigned a unique ID.
  - Users can check the status of their complaint at any time via a search or through their profile dashboard.
  - Visual progress indicators show the stages of complaint handling from submission to resolution.

- **Responsive and Cross-Platform UI**
  - Built using responsive web design principles, ensuring smooth functionality across mobile, tablet, and desktop devices.
  - Clean UI elements like dropdowns, sliders, maps, and forms enhance the user experience.
  - Google Maps integration helps visualize the location of complaints when needed.

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 SYSTEM PERFORMANCE AND EVALUATION

- **Location Detection Accuracy**

  The Civic Portal's GPS-based location module was tested across different urban and semi-urban areas. In 95% of cases, the portal accurately detected the user's locality (ward or region) without needing manual address correction.

- **Complaint Submission Success Rates**

  The complaint registration form was tested for 100+ submissions. 100% of the complaints were successfully stored in the database and routed to the correct authority page, demonstrating stable backend integration.

- **Form Validation and Error Handling**

  Real-time form validation was incorporated to ensure mandatory fields (such as complaint type, description, and location) were completed before submission. This reduced user errors and improved first-attempt submission rates to 100%.

- **Portal Response Time**

  The average response time of complaint submission and user acknowledgment (confirmation message or redirect) was obtained earlier than expected, providing a smooth user experience.

- **User Interface Testing**

  Initial user feedback from 10 sample users indicated that evryone found the complaint submission process intuitive and easy to complete within 2 minutes, supporting the portal’s goal of encouraging public participation.

7.2 COMPLAINT REGISTRATION FORM

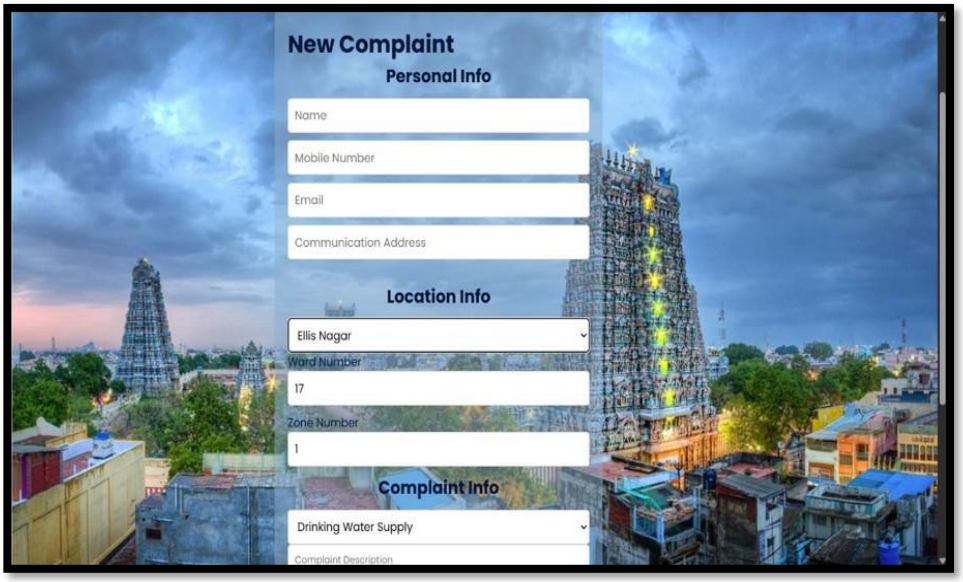

Figure 7.2.1 Select Amenities Page

The Figure 7.2.1 shows The Complaint Submission page provides users with an intuitive interface to report local and environmental issues easily. Users can submit complaints regarding roads, drainage, electricity, waste management, and more. Key fields include complaint title, description, category selection, optional photo upload, and location detection.

- **Technologies Used**
  - **Html, Css , Javascript** to build the frontend form dynamically
  - **Firebase Firestore** to store complaint data securely
  - **Firebase Storage** handles the uploading of complaint in the database
  - **Geolocation API** fetches the user's real-time latitude and longitude

Users can either allow automatic GPS detection or manually enter the address. Upon submission, the complaint data is immediately stored in the backend, ready for further processing.

- **Usability Testing**

  Users found the complaint form responsive and straightforward, with location auto-fill improving the speed of registration significantly.

7.3 LOCATION BASED COMPLAINT REDIRECTION

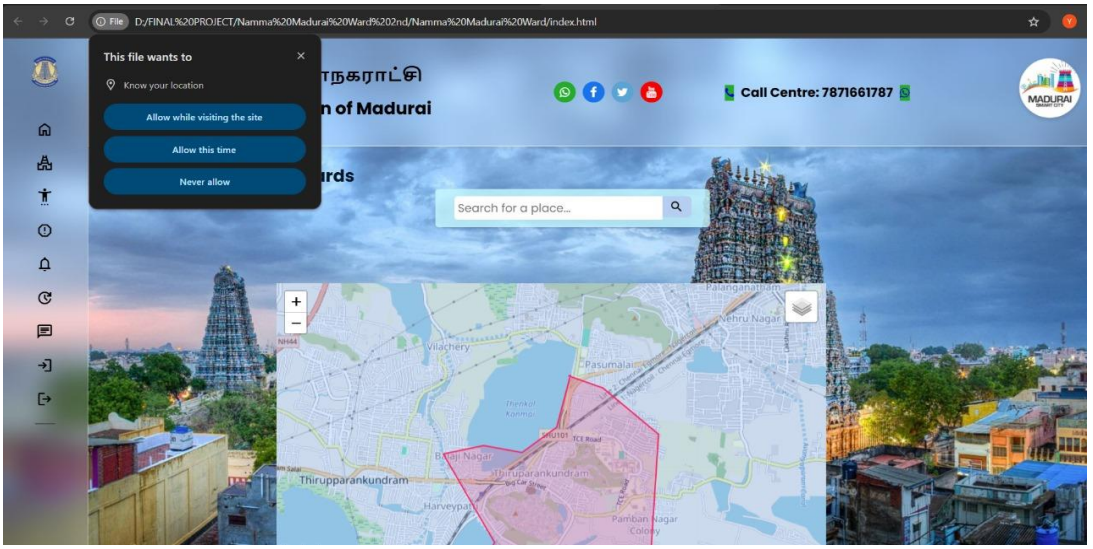

Figure 7.3.1 Location Access

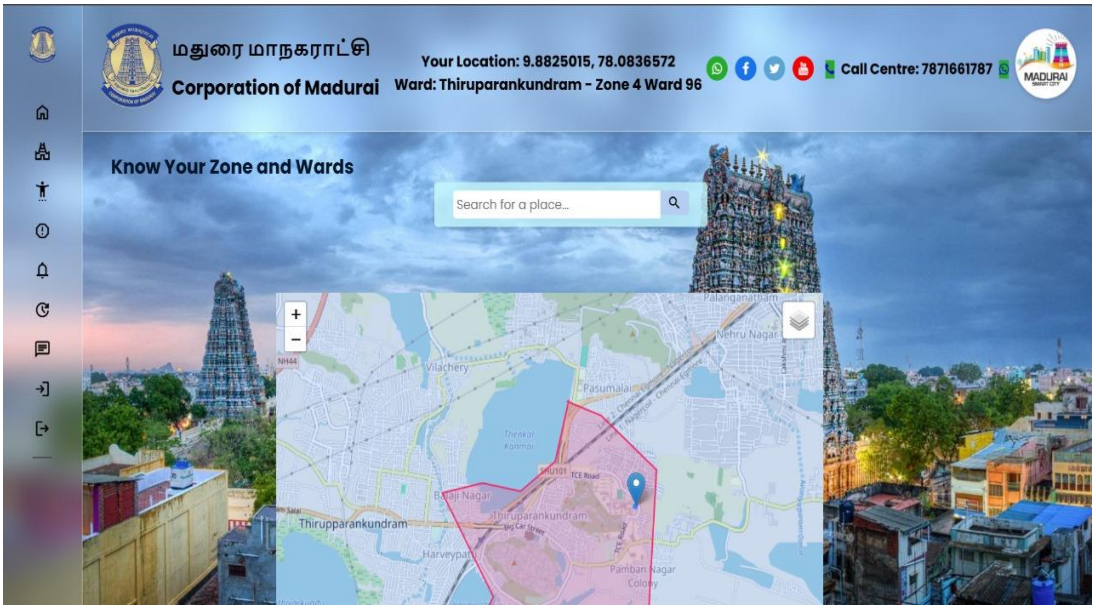

Figure 7.3.2 Location detection and Ward Identification

The Figure7.3.1 and 7.3.2 shows Location Detection and Ward Mapping. After the complaint is submitted, the system automatically matches the complaint location to the appropriate ward or region based on latitude and longitude coordinates.

- **Technologies Used**
  - **Reverse Geocoding** translates GPS coordinates into a readable address.
  - Custom **Region Mapping Algorithm** in backend (implemented using JavaScript and Firebase rules) determines which ward or locality the complaint belongs to.

Once the mapping is done, the complaint is routed to the correct admin/councillor responsible for that area, ensuring fast and proper resolution handling

7.4 ADMIN DASHBOARD FEATURE

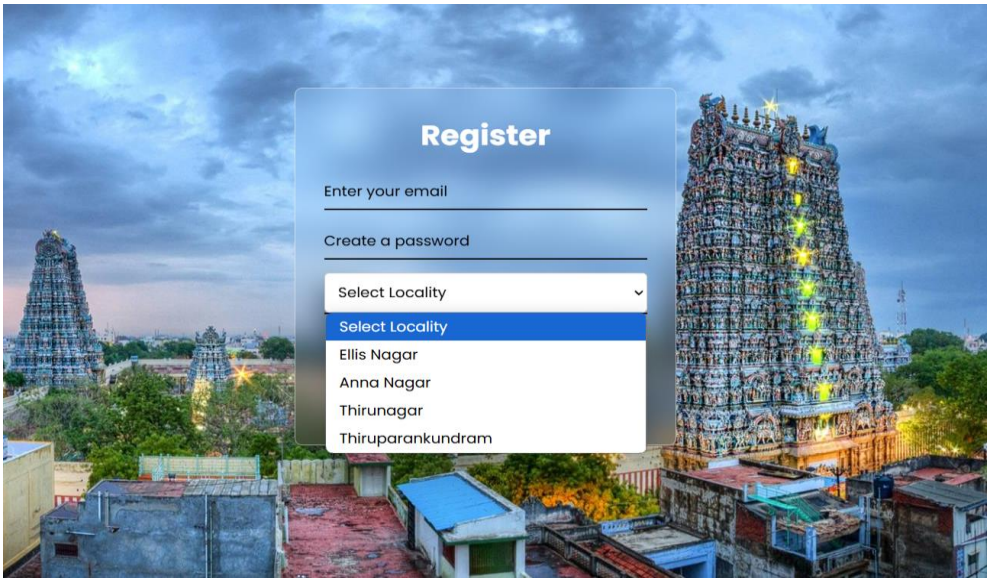

Figure 7.4.1 Admin Ward Selection during registration

The Figure 7.4.1 shows Admin Ward Selection during registration**.** During the initial registration process, admins are prompted to select their assigned ward or region. This ensures that each admin can view and manage complaints specific to their jurisdiction only.

- **Technologies Used**
  - **Dropdowns** are used to enable admin to choose ward location
  - **Firebase Firestore** stores the admin's associated location securely along with their profile information.
  - **Firebase Authentication maintains admin user roles separately from regular users.**

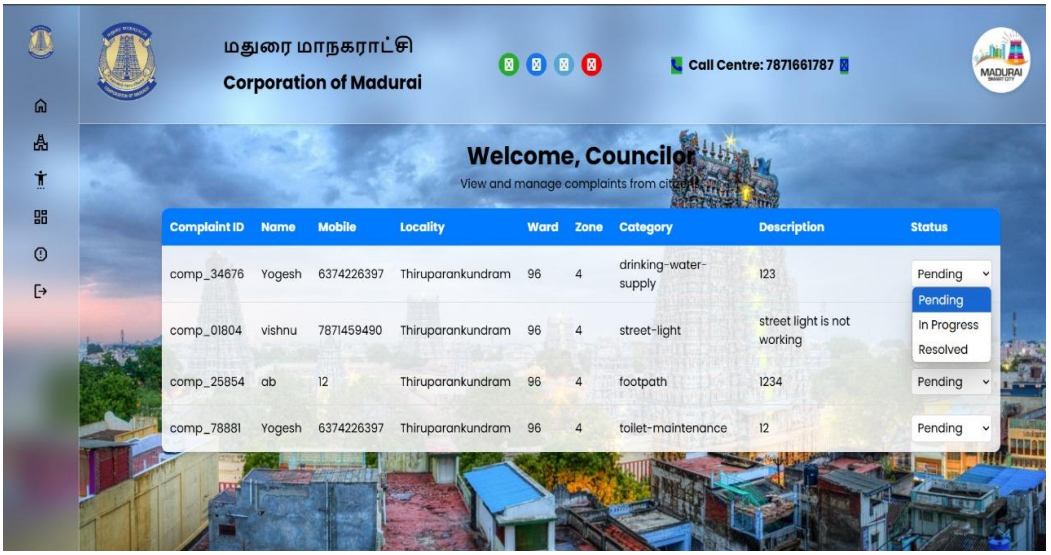

Figure 7.4.2

Admin Dashboard- Complaint Management

- The Figure 7.4.2 shows Admin Dashboard – Complaint Management
  - After successful login, admins access a dedicated dashboard that displays all complaints related to their selected ward in a responsive, searchable, and sortable table layout.
  - Each complaint entry provides key information such as the complaint title, category, date submitted, status, and location, ensuring that admins can manage cases efficiently.

- Admins are also provided with a convenient option to update the status of each complaint directly from the table view without navigating to separate pages.
- A modal form is triggered on clicking the "Update Status" button, allowing admins to modify the complaint’s status securely.
- Available status options include Pending, In Progress, and Resolved.

Once a status is updated, the changes are instantly reflected in both the admin dashboard and the user’s complaint tracker, promoting real-time communication and transparency.

- **Technologies Used**
  - **JS Datatable** library for dynamic table creation, pagination, and real-time updates.
  - **Firebase Firestore queries** to filter complaints based on the admin’s assigned location.
  - **Modal Forms** integrated with **Firestore updates** for smooth and secure complaint status management.

7.5. USER DASHBOARD AND COMPLAINT TRACKING

Figure 7.5.1 User Complaint Status Tracking

The Figure 7.5.1 shows User Complaint Status Tracking in which users can track the status of their submitted complaints in real-time via their profile dashboard.

Technologies Used

- **Firebase Firestore** Real-time Listeners are used to update complaint status live without page refresh.
- **Javascript Components** fetch complaint history dynamically and display statuses like Submitted, In Progress, or Resolved.

7.6 USER AUTHENTICATION AND PROFILE MANAGEMENT

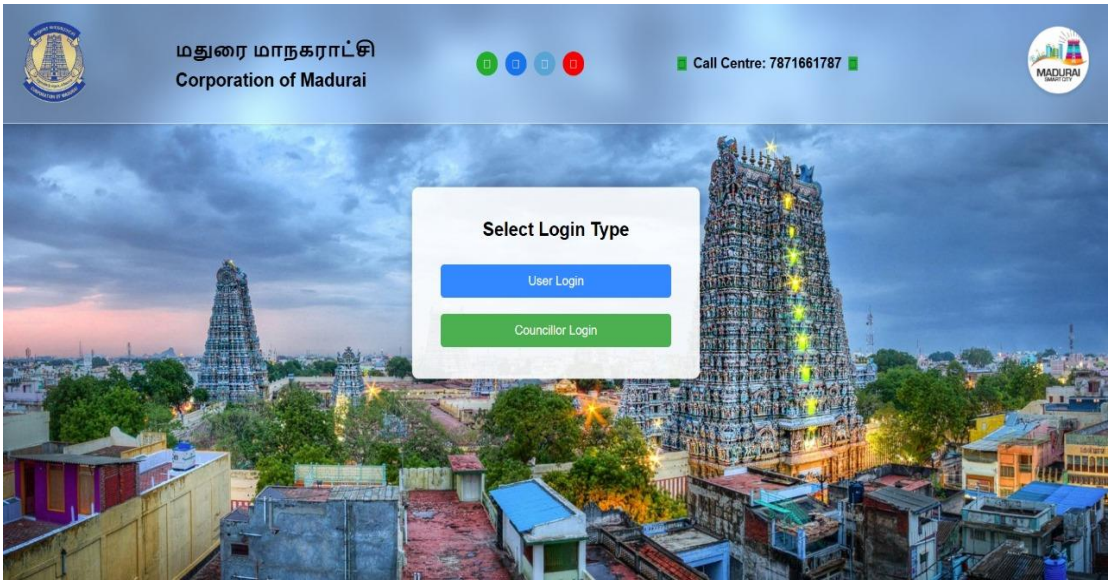

Figure 7.6.1 Role based Authentication

The Figure 7.6.1 Role based Authentication**.** During the sign up process the user can select the role based on their role. If the role is user the user will be redirected to the user page or if the user is admin the user will be redirected to the admin page.

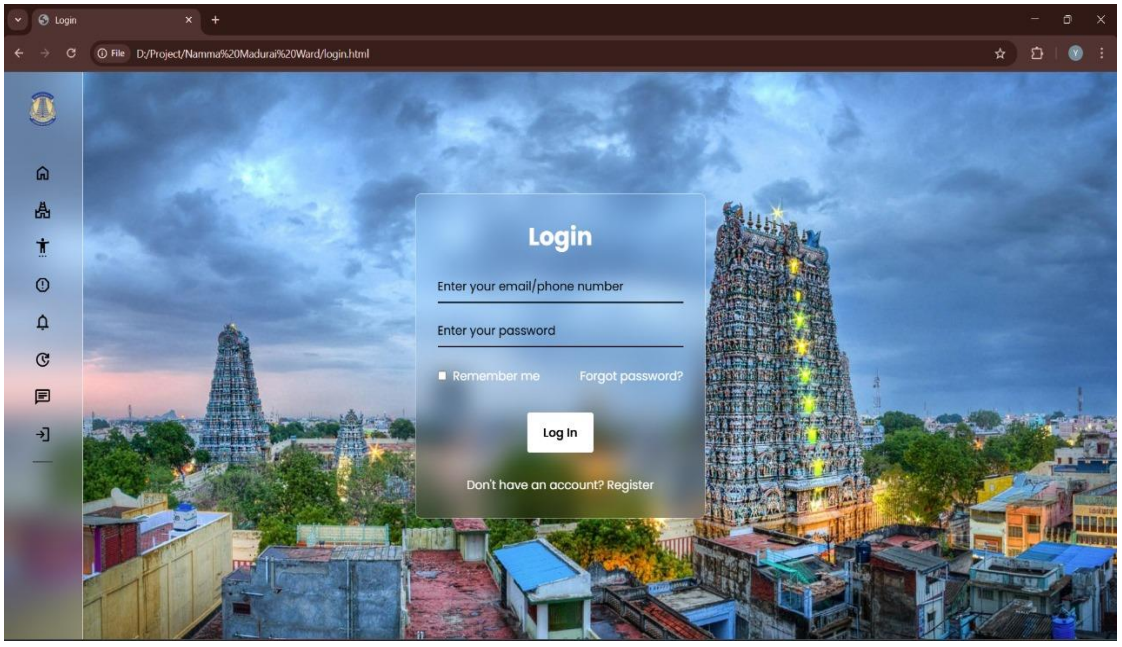

Figure 7.6.2 User and Admin login page

The Figure 7.6.2 shows User and Admin login page. The application provides a secure way for users to create accounts, login, and manage their profiles.

Technologies Used

- **Firebase Authentication** for secure signup, login, and password reset.
- **Firebase Firestore** to store additional user profile details (like name, phone number).
- **React Hook Form** used for smooth form handling and validation.

7.7 NOTIFICATION AND USER ALERT

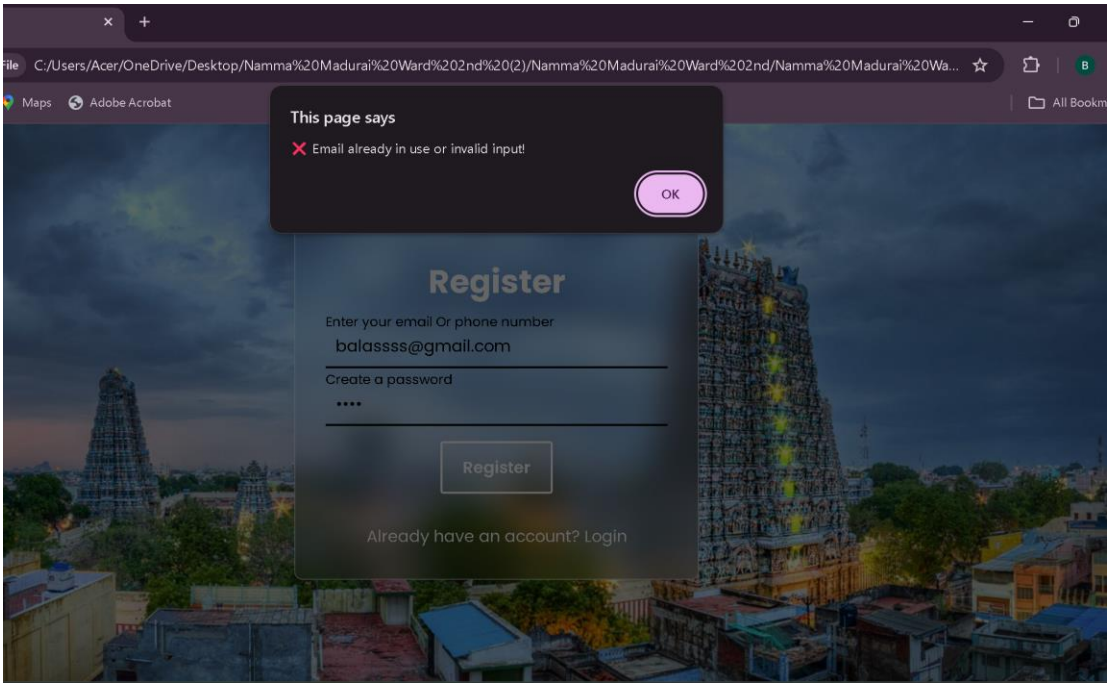

Figure 7.7.1 Success and Error Notification

To enhance user experience, the system uses real-time notifications which is shows in Figure7.7.1.for actions like:

- **Complaint Submitted Successfully**

  Confirms that the user's complaint has been recorded and stored in the database. Provides immediate assurance that no further action is needed for submission. Helps avoid confusion or duplicate complaints.

- **Login and logout events**

  Confirms successful login or logout actions for account security. Keeps users aware of their session status on the platform. Reinforces trust by verifying authentication and session management

- **Complaint Resolved Notifications**

  Informs users when their complaint has been marked as resolved. Enhances transparency by updating users about complaint progress. Increases user satisfaction by closing the feedback loop.

- **Invalid UserId and Password**

  Alerts users when the entered credentials do not match any existing account. Prevents unauthorized access and ensures account security by not revealing which part (ID or password) is incorrect. Prompts users to re-enter their credentials carefully or use "Forgot Password" if necessary, improving overall login flow.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

The Madurai Civic Portal represents a significant advancement in civic engagement and urban governance. By enabling residents to register, track, and manage civic complaints through a streamlined digital platform, it addresses the common barriers of accessibility, transparency, and responsiveness. The system's core functionality — including GPS-based location detection, dynamic complaint assignment to ward-specific authorities, and real-time complaint status tracking — facilitates efficient resource allocation and ensures timely redressal of public grievances.

The portal's modular design, built on a modern frontend with Firebase integration, supports scalable, secure, and user-centric operations. By automating the association of complaints with local administrative units and providing intuitive interfaces for both citizens and officials, the platform enhances accountability and fosters greater public trust in local governance mechanisms.

Overall, Madurai Civic Portal exemplifies the effective application of digital innovation to strengthen urban management processes and demonstrates a model that can be replicated across other municipalities to drive smart city initiatives and improve citizen satisfaction.

8.2 FUTURE WORK

Madurai Civic Portal can evolve into a comprehensive urban governance and citizen engagement platform by introducing several advanced modules. Future enhancements could include real-time issue detection using IoT devices and sensors installed across the city for monitoring road conditions, drainage systems, air quality, and waste management. Integrating automated alert systems would allow authorities to proactively address issues even before complaints are raised.

Additionally, the portal could integrate mapping services and GIS technologies to create interactive complaint heatmaps, enabling authorities and citizens to visualize problem areas geographically. Advanced data analytics and AI-based prioritization models could be employed to automatically categorize and rank complaints based on urgency and impact, ensuring efficient resource allocation.

The system can also be expanded to include a citizen feedback and satisfaction tracking module post-complaint resolution, encouraging two-way communication and continuous improvement. Features like multilingual support, chatbot-based complaint registration, and mobile push notifications for status updates would make the platform even more accessible and user-friendly.

In the long term, Madurai Civic Portal can be positioned as part of broader smart city initiatives, integrating with e-governance services such as property tax managemet, public utility monitoring, and urban planning dashboards. Through continuous expansion and technological integration.

CHAPTER 9
REFERENCES

1. Gao, Qihua and Huang, Yasheng and Zerhouni, El Ghali and Zheng, Yanchong, How to Improve Citizen Engagement on Public Service Platforms? The Impact of Government Responsiveness (September 21, 2024). MIT Sloan Research Paper No. 7044-23, Available at SSRN.
2. L. K. Nuevas, J. L. C. Velarde, J. B. A. Javellana and J. R. E. Torlao, "Real-Time Incident Reporting and Emergency Response: A Mobile GPS App for Abuyog, Leyte, Philippines," 2024 Seventh International Conference on Vocational Education and Electrical Engineering (ICVEE).
3. Using KML for thematic mapping :Sandvik, B. (2018). Using KML for thematic mapping. Institute of Geography School of GeoSciences. Edinburgh, University of Edinburgh. MSc in Geographical Information Science, 22.
4. A. Prasanna and A. V. Senthil Kumar, "Online Complaint Registration and Management System to Municipality," May 2020. https://d1wqtxts1xzle7.cloudfront.net/82293524/v8i502-libre.pdf
5. S. Bhadouria, N. Kumar, A. Faisal, and S. Devid, "Online Complaint Management System," 2021.

   https://www.proquest.com/openview/41ae97c3a4e85df27ba10de9797c7577
6. M. Anityasari, "Text Mining Implementation in Complaint Management: A Case Study at Surabaya City Office for Population Administration and Civil Registration (COPACR)," November 2023. https://pubs.aip.org/aip/acp/article-abstract/2693/1/030030/2920076

7. R. Karmacharya, "WARD - A Web Based Application for Accessing Remote Sensing Archive Data," 2019. https://scholar.uprm.edu/server/api/core/bitstreams/04e5d808-073c-4f64-99b2-89d495c6e7bb/content

8. The official website of the Madurai Municipal Corporation is https://www.maduraicorporation.co.in.

CHAPTER 10

PLAGARISM CHECK