



ONLINE COMPLAINT REGISTRATION AND MANAGEMENT SYSTEM USING MERN STACK



PRESENTED BY

SAKTHIDHASAN K (412021104006)

HARIPRASATH D (412021104003)

PALANIBHARATHI M(412021104303)



TABLE OF CONTENTS

- INTRODUCTION
- PURPOSE AND GOALS
- KEY FEATURES
- ARCHITECTURE
- TESTING
- KNOWN ISSUES
- FUTURE ENHANCEMENTS
- CONCLUSION



INTRODUCTION

The Complaint Management and Registration System is a comprehensive full-stack application, leveraging React for an interactive and user-friendly frontend and Node.js with Express for a robust and scalable backend. The frontend empowers users to easily file complaints, view updates, and track the resolution progress in real-time. Meanwhile, the backend handles critical operations such as data processing, user authentication, secure complaint storage, and efficient assignment of complaints to agents. By integrating these technologies, the system offers a seamless and reliable platform for users to address and resolve their issues effectively.



PURPOSE AND GOALS

The Complaint Registration and Management System is designed to streamline the process of registering, managing and resolving complaints. It provides a user-friendly platform for users to file complaints, track their status, and communicate with agents. Admins can efficiently assign complaints to agents while agents can update progress and resolve issues promptly.



KEY FEATURES AND FUNCTIONALITIES

- User-Friendly Complaint Registration
- Real-Time Complaint Tracking
- Admin Dashboard for Comprehensive Management
- Agent-Specific Views and Workflows
- Data Analytics and Reporting



FRONTEND ARCHITECTURE

The frontend architecture is designed to efficiently handle three user roles—Admin, Ordinary User, and Agent—each with distinct functionalities. The architecture includes modular component structures, role-based routing, state management, and a robust UI framework, making the application scalable and user-friendly.



ORDINARY USER

- File a new complaint by providing required details (e.g., name, address, issue description).
- View the status of submitted complaints (e.g., pending, assigned, completed).
- Communicate with the assigned agent through the integrated messaging feature.



AGENT

- View complaints assigned by the admin.
- Communicate with users to gather additional details or provide updates via messaging.
- Update the status of complaints (e.g., in-progress, completed) to reflect progress.



ADMIN

- View all registered complaints in the admin dashboard.
- Assign complaints to specific agents based on workload or expertise.
- Monitor the status and progress of complaints assigned to agents.
- Manage the overall flow and resolution of complaints in the system.



BACKEND ARCHITECTURE

The backend architecture supports the primary user roles—Admin, Ordinary User, and Agent—focusing on modularity, scalability, and role-based permissions. The architecture is implemented using Node.js and Express.js, with MongoDB for the database, Mongoose for ORM, and JWT for secure authentication.



DATABASE DESIGN

- User Model: Stores user details (name, email, role, password). Users can create complaints and interact with messages.
- Complaint Model: Represents complaints with fields like title, description, status, and references to the user and assigned staff.
- Assigned Model: Tracks which admin or staff member is assigned to a complaint, with fields for assignment status and timestamps.
- Message Model: Stores messages related to complaints, including sender, content, and timestamps.



TESTING

- ❖ **Responsive Testing:** We used browser developer tools to ensure the application is responsive and functions properly across different screen sizes and devices, providing a seamless user experience on mobile, tablet, and desktop.
- ❖ **API Testing with Postman:** We used Postman to test the backend APIs, ensuring that all endpoints for complaint registration, user authentication, complaint tracking, and message handling are working correctly. This included testing for correct responses, error handling, data validation, and performance under various conditions.



KNOWN ISSUES

There are a few issues within the application that may affect user experience. These include problems with data persistence, such as certain changes not being reflected after page refreshes. Login and signup functionalities may also experience validation errors. Efforts are ongoing to address and resolve these issues in future updates.



FUTURE ENHANCEMENTS

- Advanced Search and Filtering
- Real-Time Notifications
- User Role Permissions
- Analytics Dashboard
- AI-Driven Complaint Routing
- Mobile App Integration
- Multilingual Support



CONCLUSION

The Complaint Management System streamlines the process of registering, assigning, and resolving complaints, fostering better communication between users, agents, and administrators. This project highlights the importance of technology in simplifying complex workflows and demonstrates a scalable approach to complaint management



THANK YOU