ResolveNow: Your Platform For Online Complaints

Project Title: Online Complaint Registration and Management System

Team Members:
Kode Devi Sri Vallika (Lead)
Tupakula Yoshika Yadav
Shakina Dasari
Lukka Sanjay

1. INTRODUCTION

1.1 Project Overview

The Online Complaint Registration and Management System is a comprehensive, full-stack web application developed using the MERN stack (MongoDB, Express.js, React.js, Node.js). It is designed to modernize and digitize the grievance redressal process for organizations, municipalities, or service providers. The system features a robust Role-Based Access Control (RBAC) architecture, segregating functionalities into three distinct user modules: Customer, Administrator, and Field Agent. By providing a centralized digital portal, the application ensures that service requests are securely logged, intelligently routed, and swiftly resolved.

1.2 Purpose

The primary purpose of this project is to eliminate the inefficiencies associated with traditional, manual complaint tracking methods—such as paper forms, unorganized email threads, and redundant phone calls. The system aims to establish absolute transparency by providing users with real-time status updates (e.g., Pending, Assigned, Resolved) regarding their issues. Furthermore, it empowers administrative staff with a dynamic dashboard to oversee operations, track agent workloads, and ensure timely issue resolution, thereby significantly enhancing overall customer satisfaction.

2. IDEATION PHASE

2.1 Problem Statement

Citizens and consumers frequently face long delays and a complete lack of transparency when reporting service issues (e.g., utility failures, hardware malfunctions). Administrators struggle to manage these incoming complaints due to chaotic, paper-based workflows, resulting in lost requests and inefficient task delegation to field agents.

2.2 Empathy Map Canvas

During the ideation phase, we analyzed the user's perspective:

Thinks & Feels: Frustrated by slow responses; feels ignored when there is no follow-up on a submitted complaint.
Sees: Long queues at service centers and a lack of digital infrastructure.
Says & Does: Frequently calls helplines to ask, "What is the status of my issue?"; wastes time visiting offices physically.
Pains: Unpredictable resolution times and lack of direct communication with the assigned technician.
Gains: Peace of mind through automated tracking and guaranteed issue resolution.

2.3 Brainstorming

To address these pain points, the team brainstormed a transition from offline methods to a cloud-based digital ecosystem. Key ideas included implementing a secure login system, categorizing complaints by location/type, and creating a visible "status badge" system so users instantly know if their problem is being worked on.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Discovery: User experiences a service failure and accesses the web portal.
Onboarding: User registers an account and securely logs in.
Action: User submits a detailed grievance form (Title, Location, Description).
Tracking: User monitors the dashboard as the system updates the ticket from "Pending" to "Assigned".
Resolution: The Agent completes the work, marks the ticket as "Resolved," and the User views the successful outcome.

3.2 Solution Requirement

Functional Requirements: Secure user registration/authentication, CRUD (Create, Read, Update, Delete) operations for complaints, Admin assignment controls, and Agent status toggle functionalities.
Non-Functional Requirements: High usability with a modern "Glassmorphism" UI, strict data security via password hashing, system reliability, and high availability (24/7 access).

3.3 Data Flow Diagram

The system's data flow begins with the User inputting complaint data into the React frontend. This data is transmitted as a JSON payload via RESTful APIs to the Node.js/Express backend, which validates and stores it in the MongoDB database. The Admin queries this database to fetch pending records and updates the document with an Agent's ID. Finally, the Agent queries their specific tasks and pushes an update back to the database upon completion.

3.4 Technology Stack

Frontend: React.js, React Router DOM, Bootstrap, Custom CSS.
Backend: Node.js, Express.js.
Database: MongoDB, Mongoose ORM.
Tools: Postman (API Testing), VS Code, GitHub.

4. PROJECT DESIGN

4.1 Problem Solution Fit

Aligning with the Problem-Solution Fit Canvas, our primary Customer Segment is service users restricted by time constraints and poor communication channels. The core Job-to-be-done is filing a complaint and receiving a guaranteed resolution timeline. Because the Root Cause of current frustration is a lack of centralized tracking, our Solution perfectly fits by acting as a digital bridge. By shifting the Channel of Behavior to an online dashboard, we transition the user's emotion from anxious to confident.

4.2 Proposed Solution

The proposed system is an interactive MERN stack web application. It introduces novelty through its distinct, segregated dashboards that prevent data overlap. The UI leverages a modern, professional command-center aesthetic, ensuring the application is visually engaging while remaining highly functional for administrative tasks.

4.3 Solution Architecture

The project follows a strict Model-View-Controller (MVC) architectural pattern.
Model: Mongoose schemas defining the structure of Users and Complaints.
View: React components rendering the dynamic user interface.
Controller: Express routes handling business logic, API endpoints, and database interactions.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The project was executed using Agile methodologies, divided into distinct sprints:
Sprint 1: Requirement gathering, UI/UX wireframing, and environment setup.
Sprint 2: Database schema design and backend API development (Node/Express).
Sprint 3: Frontend development (React) and integration of the role-based dashboards.
Sprint 4: End-to-end integration, bug fixing, and user acceptance testing.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

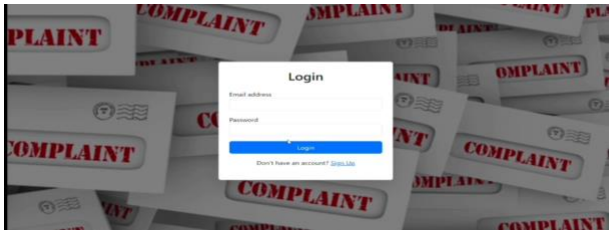
Testing was conducted continuously throughout the development lifecycle:
API Testing: Endpoints were tested using Postman to ensure rapid response times (under

200ms) and correct HTTP status codes (200 OK, 201 Created, 400 Bad Request).
Frontend Rendering: React component load times were optimized to ensure the dashboards render smoothly without lag, even when fetching multiple complaint documents.
UAT (User Acceptance Testing): Manual testing verified that state changes (e.g., an Admin clicking "Assign") instantly reflected accurately in both the database and the respective Agent's dashboard.

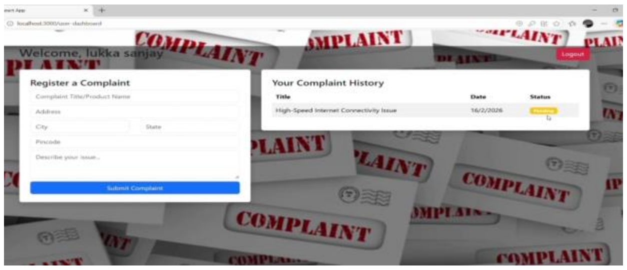
7. RESULTS

7.1 Output Screenshots

(Note: Insert the actual screenshots of your project below these captions)



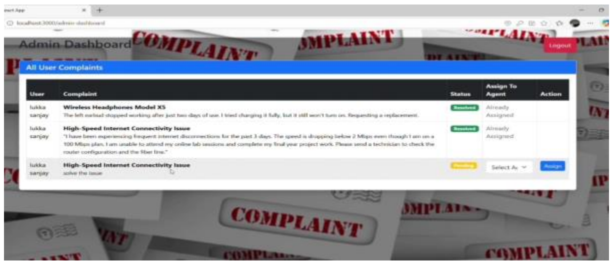
Screenshot 1: User Registration and Login interface featuring the high-tech Glassmorphism design.



Your Complaint History

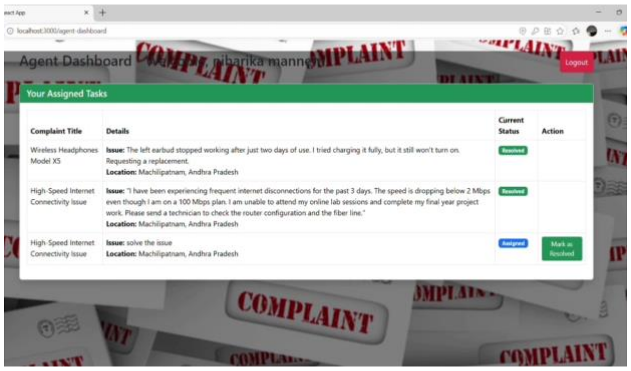
Title	Date	Status
High-Speed Internet Connectivity Issue	16/2/2026	Pending

Screenshot 2: Customer Dashboard displaying a history of raised complaints with "Pending" status badges.



User	Complaint	Status	Assign To Agent	Action
lukka sanjay	**Wireless Headphones Model X5** The left earbud stopped working after just two days of use. I tried charging it fully, but it still won't turn on. Requesting a replacement.	Resolved	Already Assigned	
lukka sanjay	**High-Speed Internet Connectivity Issue** "I have been experiencing frequent internet disconnections for the past 3 days. The speed is dropping below 2 Mbps even though I am on a 100 Mbps plan. I am unable to attend my online lab sessions and complete my final year project work. Please send a technician to check the router configuration and the fiber line."	Resolved	Already Assigned	
lukka sanjay	**High-Speed Internet Connectivity Issue** solve the issue	Pending	Select A	Assign

Screenshot 3: Administrator Dashboard showing the master list and the Agent assignment dropdown menu.



Complaint Title	Details	Current Status	Action
Wireless Headphones Model X5	**Issue:** The left earbud stopped working after just two days of use. I tried charging it fully, but it still won't turn on. Requesting a replacement. **Location:** Machilipatnam, Andhra Pradesh	Resolved	
High-Speed Internet Connectivity Issue	**Issue:** "I have been experiencing frequent internet disconnections for the past 3 days. The speed is dropping below 2 Mbps even though I am on a 100 Mbps plan. I am unable to attend my online lab sessions and complete my final year project work. Please send a technician to check the router configuration and the fiber line." **Location:** Machilipatnam, Andhra Pradesh	Resolved	
High-Speed Internet Connectivity Issue	**Issue:** solve the issue **Location:** Machilipatnam, Andhra Pradesh	Assigned	Mark as Resolved

Screenshot 4: Field Agent Dashboard displaying assigned tasks and the "Mark as Resolved" action button.

8. ADVANTAGES & DISADVANTAGES

Advantages:
Transparency & Tracking: Users are continuously aware of their complaint's status.
Operational Efficiency: Admins can delegate tasks in seconds without paperwork.
Scalability: The NoSQL nature of MongoDB allows the system to scale and handle thousands of complaints effortlessly.
Accessibility: Cloud-based architecture means the system can be accessed from any device with a browser.
Disadvantages:
Internet Dependency: Requires an active internet connection to log and track complaints.
Digital Literacy: Requires users to have basic technical knowledge to navigate web forms and dashboards.

9. CONCLUSION

The Online Complaint Registration and Management System successfully achieves its objective of providing a streamlined, digital alternative to traditional grievance redressal. By leveraging the power of the MERN stack, the development team created a highly responsive, secure, and transparent platform. The successful implementation of role-based access control ensures that Customers, Admins, and Agents can interact seamlessly within the same ecosystem, significantly reducing resolution times and improving overall service quality.

10. FUTURE SCOPE

While the current system is fully functional, future enhancements could include:
Automated Notifications: Integrating API services (like Twilio or Nodemailer) to send

instant SMS or Email alerts when a complaint status is updated.
AI Chatbot Integration: Adding an intelligent virtual assistant to help users categorize their issues accurately before submitting the form.
Data Analytics: Developing graphical reporting tools within the Admin dashboard to analyze complaint trends, average resolution times, and agent performance.
Source Code: Available upon academic request. Core modules include App.js (Frontend Routing), index.js (Server Entry), and complaints.js (API Logic).
Dataset Link: Not applicable (Data is generated dynamically by users in MongoDB).
GitHub & Project Demo Link: Repository: https://github.com/LukkaSanjay/online-complaint-registration-and-management-system
Demo Video:
https://drive.google.com/drive/folders/1IsfoLoByEv0VklPzHOBKAreumnf9p850