

Date	24 February 2026
Team ID	LTVIP2026TMIDS90304
Project Name	ONLINE COMPLAINT REGISTRATION AND MANAGEMENT SYSTEM

1. Introduction

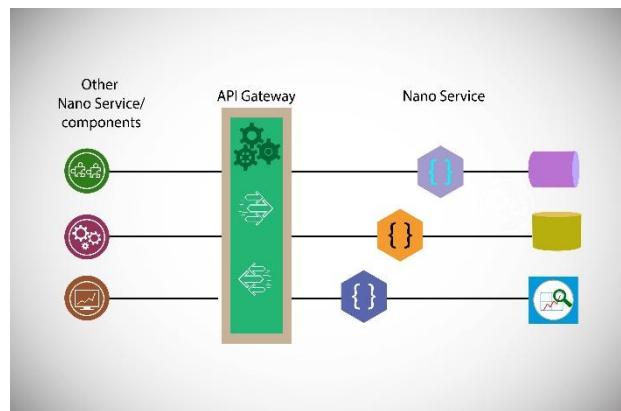
- **Project Title:** Online Complaint Registration and Management System.
- **Team Members:** Chandan (Full Stack Developer).

2. Project Overview

- **Purpose:** To provide a centralized platform for managing complaints, streamlining the resolution process, and improving customer satisfaction.
- **Features:**
 - **User Registration:** Accounts for submitting and tracking complaints.
 - **Complaint Submission:** Entry of issue details including description and address.
 - **Real-time Interaction:** Built-in messaging feature to chat with assigned agents.
 - **Admin Management:** Assignment of complaints to agents based on workload.

3. Architecture

- **Frontend:** Built with **React.js** using **Axios** for RESTful API communication and **Bootstrap/Material UI** for the interface.
- **Backend:** Powered by **Node.js** and **Express.js** to handle server-side logic and routing.
- **Database:** **MongoDB** stores user profiles, complaint details, and chat messages via **Mongoose**.



Shutterstock

4. Setup Instructions

- **Prerequisites:** Node.js, npm, and MongoDB.
- **Installation:**
 1. **Clone the Repository:** `git clone https://github.com/awdhesh-student/complaint-registry.git`.
 2. **Install Frontend Dependencies:** `cd frontend && npm install`.
 3. **Install Backend Dependencies:** `cd ..//backend && npm install`.

5. Folder Structure

- **Frontend:** Organized into components/ (admin, agent, user, common), hooks/, api/, and

Images/.

- **Backend:** Consists of config/ (database connection), controllers/, models/ (schemas), and routes/.

6. Running the Application

- **Backend:** Run npm start in the backend directory (starts on port 8000).
- **Frontend:** Run npm start in the frontend directory (accessible at <http://localhost:3000>).

7. API Documentation

- **Auth:** POST /api/auth/register, POST /api/auth/login.
- **Complaints:** POST /api/users/complaint, GET /api/users/complaints.
- **Admin:** POST /api/admin/assign.
- **Messages:** GET /api/messages/:complaintId, POST /api/messages.

8. Authentication

- **Method:** Handled via **Bcrypt** for password hashing and **Express-session** or **JWT** for maintaining secure user sessions.

9. User Interface

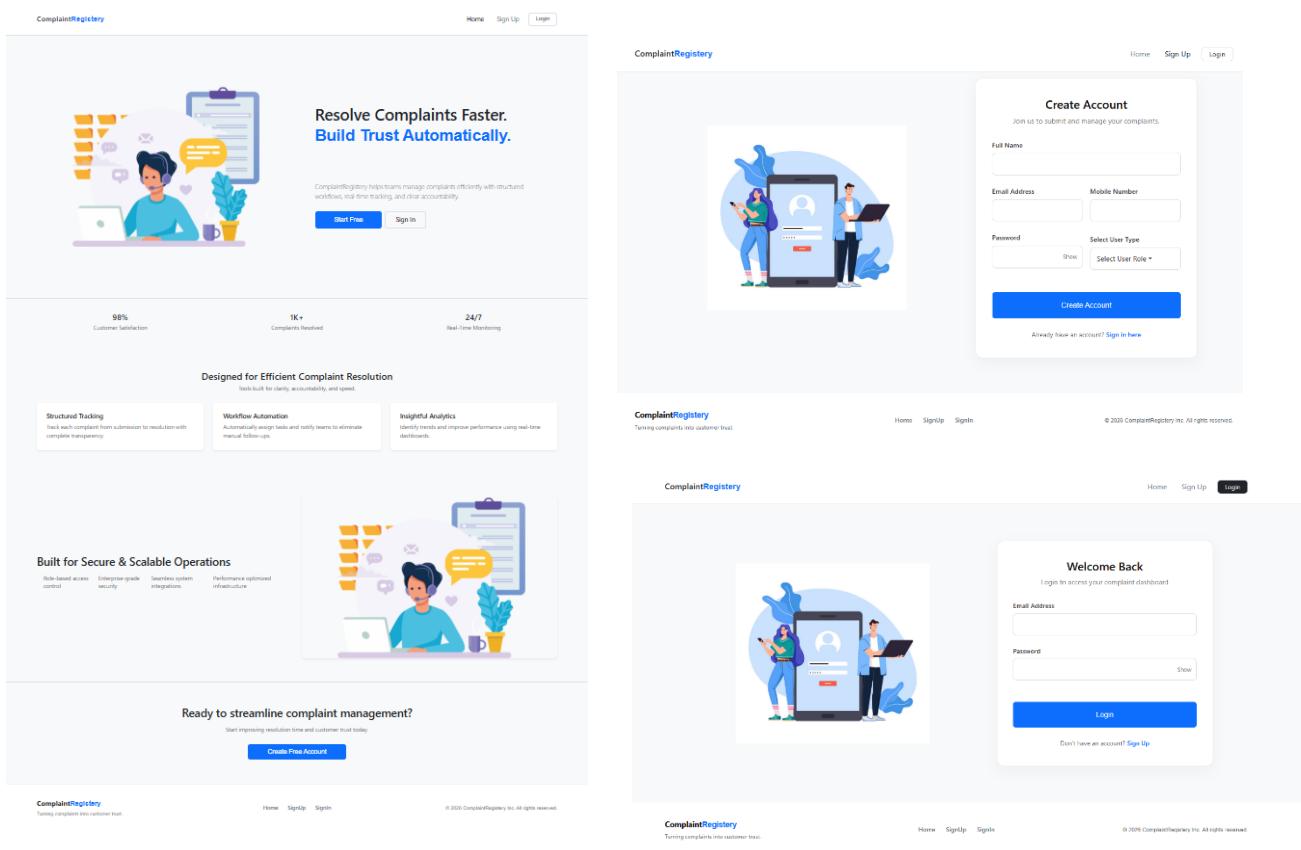
- **Dashboards:** Unique views for Users (Register/Status), Agents (View Assigned), and Admins (Full oversight).

10. Testing

- **Strategy:** Functional testing of CRUD operations and performance testing of API response times.

11. Screenshots or Demo

- **Visuals:** Includes Landing Page, Registration, and specific User/Agent/Admin Dashboards



SYSTEM OVERVIEW

Admin Dashboard

TOTAL 2 **AGENTS** 1 **PENDING** 0 **RATE** 50%

Search customers or issues... All Statuses

CUSTOMER	STATUS	ISSUE DETAILS	ACTION
Chandan Dakka Kodak AP	Assigned	Working...	Assign
Chandan Dakka Kodak AP	Waiting	Test	Assign

COMPLAINTREGISTRY | New Complaint | Status

OFFICIAL COMPLAINT (ID: 123-001-000)

SEARCH

First Name of Recipient: Chandan Dakka
Last Name: AP

DETAILED DESCRIPTION:
To file official complaints here.

MAILING ADDRESS:
Street and House Number: 123 Main Street
City: Anytown
State: CA
ZIP: 90210

FILE COMPLAINT

AgentPortal | View Complaints

Assigned Complaints

Manage and resolve tickets in your queue

ID	Customer	Status	Action
REF: 12345	Chandan Dakka Kodak AP	Assigned	Chat
REF: 54321	Chandan Dakka Kodak AP	Assigned	Chat

Done **Chat**

COMPLAINTREGISTRY | New Complaint | Status

TRACKING ARCHIVES

History of ticket archives at our organization

ID	Customer	Status	Action
REF: 12345	Chandan Dakka Kodak AP	Assigned	Open Correspondence
REF: 54321	Chandan Dakka Kodak AP	Assigned	Open Correspondence

12. Future Enhancements

- Intelligent Routing:** Implementation of AI/ML algorithms to automatically categorize and route complaints.
- SMS Integration:** Adding direct SMS notifications for status updates.