Overview

ResolveNow: Your Platform for Online Complaints, is a powerful full-stack web application built with React.js, Node.js, Express.js, MongoDB, Socket.io, and WebRTC. It streamlines the entire complaint lifecycle—from submission to resolution—including real-time chat, automated routing, and admin control.

🔍 Key Features

- ## User Registration & Secure Authentication

Users register with email/password and verify via email. Authentication is managed through JSON Web Tokens (JWT), ensuring secure access—an approach similar to best practices shown in the E-Complaints project

- ## Complaint Submission Form

Users can submit detailed complaints with descriptions

- ## Automated Assignment & Routing

Complaints are automatically classified based on category, then routed to appropriate agents—mirroring the admin-routing logic evident in user-Complaints

- ## Real-Time Status Tracking

Users can monitor complaint progress on dashboards with real-time updates, driven by WebSockets (Socket.io), similar to the dynamic dashboards in Complaint Management System React apps .

- ## Live Chat Feature

Users and agents can communicate via in-app chat powered by Socket.io. Messages are linked to complaintId and userId, matching common practices in other helpdesk solutions.

- Admin Dashboard

Admins have comprehensive oversight: they can review complaints, assign agents, monitor workloads, and enforce policies—features similar to admin interfaces in several open-source solutions .

- Agent Interface

Agents can view complaints assigned to them, update status, chat with users, and see resolution feedback.

- Security & Compliance

Built with secure authentication (JWT), HTTPS support, encrypted data transport, role-based access control, and audit logs—aligned with industry-standard architectures .

Technology Stack

Layer	Tools & Frameworks
Frontend	React.js, Material-UI, Bootstrap, Axios, Socket.io-client
Backend	Node.js, Express.js, Mongoose, Socket.io, WebRTC
Database	MongoDB

📇 ER Diagram & Data Models

- User: Stores personal info, hashed password, role, contact details.
- Agent: Similar to User but with agent-specific metadata and workload stats.

- Complaint: Tracks submission details, status history, attachments.

- Message: Maps user-agent chat messages to complaintId, timestamps.

- Admin: Handles complaint assignment and system-wide logs/metric
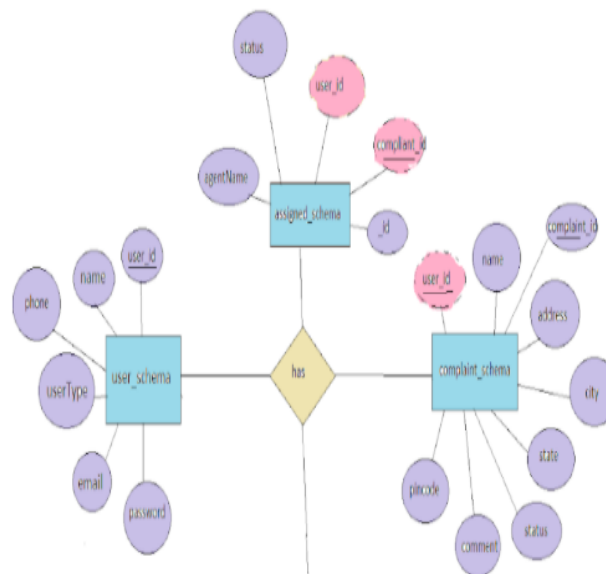
⚒ API & Routing

Auth Routes: /signup, /login, /verify, protected routes using JWT.

Complaints Routes: CRUD operations, file uploads, attachments processing.

Agent/Admin Routes: /assign, /updateStatus, /allComplaints, workload tracking.

Messages Route: Real-time messaging with persistent storage.

Notifications: Endpoints or background service to trigger communication

▦ Frontend Architecture

- Structured into components (Login, Dashboard, ComplaintForm, Chat, Profile).

- Utilizes client-side routing (React Router) for seamless user navigation.

- Manages state via React hooks/context, maintaining clarity and modularity.

- Communicates via Axios, handles authentication via tokens.

- Implements Socket.io for live updates and chat.

🚀 Setup & Deployment

1. Clone Project

git clone https://github.com/Gattu-Nandini/Online-Complaint-Registration-andManagement-System

cd complaint-registery

NOTE: Install all the required applications before execution . In your terminal run the below commands in different terminals for backend and frontend

2. Backend Setup

cd backend

npm install

npm start

3. Frontend Setup

cd frontend

npm install

 npm start

4. Run Application

Access at http://localhost:3000, with backend typically at http://localhost:5000.

Once you accessed, you will be able to see the below web page