

# Full Stack Documentation with MERN

## Resolve Now: Your Platform for Online Complaints

---

### 1. Introduction

- **ProjectTitle:** Resolve Now – Your Platform for Online Complaints
- **Team Members:**
  1. Sri Varshini – Frontend & Backend Development

### 2. Project Overview

- **Purpose:**

Resolve Now is designed to streamline the process of submitting, tracking, and resolving user complaints through a centralized digital platform, enhancing transparency, communication, and customer satisfaction.
- **Features:**
  1. Secure user registration and login
  2. Complaint submission with multimedia support
  3. Real-time complaint tracking and notifications
  4. Built-in chat between users and agents
  5. Admin panel for complaint and agent management
  6. Intelligent routing of complaints
  7. Strong security and data confidentiality
  8. Feedback system and status updates

### 3. Architecture

- **Frontend:**

Built with React.js, styled using Bootstrap and Material UI for responsive and dynamic user interfaces. Integrated with backend using Axios and supports chat functionality using Socket.io.
- **Backend:**

Developed using Node.js and Express.js to handle REST APIs, routing, authentication, complaint processing, and chat support.
- **Database:**

MongoDB used for storing user profiles, complaints, chat messages, and complaint assignments. Schemas include:

  1. User Schema
  2. Complaint Schema
  3. Assigned Complaint Schema
  4. Chat Window Schema

### 4. Setup Instructions

- **Prerequisites:**

-> Node.js

-> MongoDB

-> npm

-> Git

- **Installation:**

1. Clone the repository
2. Run "npm install" in both "client" and "server" directories
3. Configure ".env" for database and JWT secrets
4. Start frontend and backend servers

## 5. Folder Structure

- Client (Frontend):

/client

├─ public/

├─ src/

| └─ components/

| └─ pages/

| └─ services/

| └─ App.js

- Server (Backend):

/server

├─ config/

├─ controllers/

├─ models/

├─ routes/

└─ server.js

## 6. Running the Application

- **Frontend:**

cd client

npm start

- **Backend:**

cd server

npm start

## 7. API Documentation

- Register a new User

A screenshot of a web browser displaying a registration form titled "SignUp For Registering the Complaint". The form is centered on a dark background. It includes input fields for "Full Name" (containing "sri"), "Email" (containing "srivarshinigeddada3@mail.com"), "Password" (masked with "\*\*\*\*\*"), and "Mobile No." (containing "01234567890"). Below these fields is a dropdown menu labeled "Select User Type" with "Ordinary" selected. The browser's address bar shows "localhost:3000/SignUp".

- Login Page

A screenshot of a web browser displaying a login form titled "Login For Registering the Complaint". The form is centered on a dark background. It includes input fields for "Email" (containing "srivarshinigeddada3@mail.com") and "Password" (masked with "\*\*\*\*\*"). Below these fields is a "Login" button. At the bottom, there is a link that says "Don't have an account? [SignUp](#)". The browser's address bar shows "localhost:3000/login".

- Register Complaint

The screenshot shows a web browser window with the URL `localhost:3000/HomePage`. The page has a dark header with the text "Hi, sri" and navigation links "Complaint Register" and "Status". A "LogOut" button is in the top right. The main content area features a dark-themed form with the following fields:

- Name: `sri`
- Address: `lkvm`
- City: `Lakkavaram`
- State: `Andhra Pradesh`
- Pincode: `533254`
- Status: `pending`
- Description: `hello`

A green "Register" button is located below the description field. The footer of the page displays "ComplaintCare".

- Get Complaint details and Assign to Agent

The screenshot shows a web browser window with the URL `localhost:3000/AdminHome`. The page has a dark header with the text "Hi Admin adminv" and navigation links "Dashboard", "User", and "Agent". A "Log out" button is in the top right. The main content area is titled "Users Complaints" and displays a grid of complaint cards. Each card contains the following information:

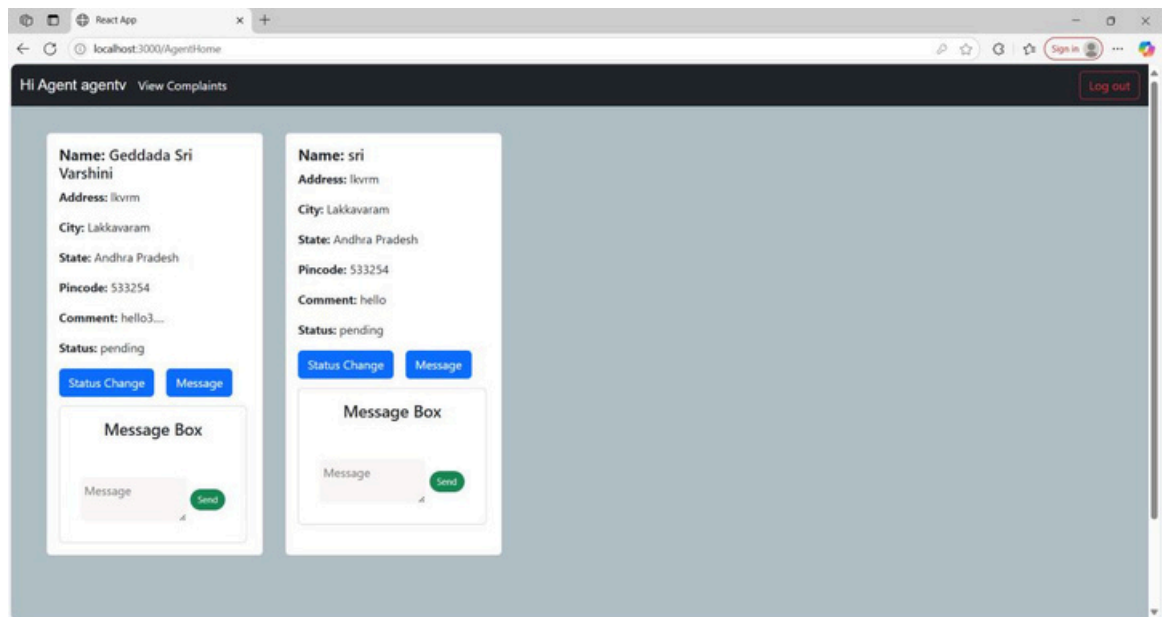
- Name: Geddada Sri Varshini
- Address: lkvm
- City: Lakkavaram
- State: Andhra Pradesh
- Pincode: 533254
- Comment: hello2
- Status: completed

Below the grid, there is a section for "Name: sri" with the following details:

- Address: lkvm
- City: Lakkavaram
- State: Andhra Pradesh
- Pincode: 533254
- Comment: hello

Each card has an "Assign" button at the bottom.

- **Send Response**



## 8. Authentication

- **Method: JSON Web Tokens (JWT)**
  - > Tokens are generated at login and used for authorization in protected routes.
  - > Middleware is used to verify tokens and determine user roles (admin, agent, customer).

## 9. User Interface

- **CustomerDashboard:** Submit complaints, track status, chat with agents
- **Agent Dashboard:** View assigned complaints, respond to users, update status
- **Admin Panel:** Monitor all activity, assign agents, manage users

## 10. Testing

- Manual testing through Postman for API validation
- Frontend testing through UI interactions
- Unit testing setup can be extended using Jest or Mocha

## 11. Screenshots or Demo

-Here is the VIDEO DEMO Link:

[https://drive.google.com/file/d/1sYFgYx5FWyvRVqjwKXv5UU\\_NZp5jhAqI/view?usp=sharing](https://drive.google.com/file/d/1sYFgYx5FWyvRVqjwKXv5UU_NZp5jhAqI/view?usp=sharing)

## 12. Known Issues

- Video chat support may not work on all browsers (WebRTC compatibility)

- Email delivery may delay under heavy server load
- UI responsiveness under low network conditions may lag

### **13. Future Enhancements**

- Implement advanced analytics dashboard for admins
- Add complaint categories and SLA monitoring
- Integrate WhatsApp/SMS support for notifications
- Role-based access control with granular permissions
- Support for multilingual UI