Full Stack Development

1. Introduction

- **Project Title:** RightVoice – Your Platform for Online Complaints
- **Team Members:** Balusu Samhitha
  Chandramalla Manoj Kumar
  Chilamkurthi Navi Sai Ram
  Digumarthi Manasa

2. Project Overview

Purpose: RightVoice is a comprehensive complaint registration and resolution platform. It enables users to securely lodge complaints, track them in real time, and communicate directly with support agents. The platform empowers organizations to streamline issue resolution, meet regulatory compliance, and enhance customer satisfaction.

Goals:

- Simplify the complaint submission process
- Enable efficient complaint assignment to agents
- Track status updates and improve transparency
- Provide a secure and centralized system for all stakeholders

Features:

User Registration & Login:
Users create secure accounts with email verification.

Complaint Submission:
Users submit detailed complaints with text, documents, and images.

Complaint Dashboard:

- Users: View and track their complaints
- Admins: View all complaints and assign to agents
- Agents: View assigned complaints and update progress

Complaint Assignment:
Admins allocate complaints to agents based on expertise and availability.

User-Agent Communication:
Real-time chat feature for direct user-agent interaction.

Notifications:
Email or SMS alerts on complaint status updates.

Security & Compliance:
JWT-based auth, access control, data encryption, GDPR-compliant handling

3. Architecture

- **Frontend:** Developed with React.js using Bootstrap and Material UI for responsive, interactive web pages. Axios is utilized to make HTTP requests for backend API communication.
- **Backend:** Built with Node.js and the Express.js framework. RESTful APIs handle frontend-backend data exchange. Socket.IO and WebRTC for real-time communication in chat between user and agent.
- **Database:** Data persistence with MongoDB to store users, complaints, chat history, and admin data. Schema-based interaction provided by Mongoose ORM.

4. Setup Instructions

➢ Step 1: Install Prerequisites

Make sure the following are installed on your machine:

- **Node.js** – Download
- **MongoDB** (local or Atlas) – Download MongoDB Community Server
- **Git** – Download Git

➢ Step 2: Clone the Project Repository

➢ Step 3: Setup Backend

- Go to the server folder
- Install dependencies
- Create a .env file in the server directory
- Start the backend server

➢ Step 4: Setup Frontend

- Go to the client folder
- Install frontend dependencies
- Start the frontend server

➢ Step 5: Use the Application

- Open your browser and visit: http://localhost:3000
- Register as a **User**, **Agent**, or **Admin**
- Start submitting complaints, assigning them, and tracking their progress

5. Folder Structure

COMPLAIN-REGISTER-NM-MERN-Stack/

├── backend/

│

├── frontend/

│ ├── node_modules/

│ ├── public/

```
│   ├── src/
│   │   ├── components
│   │   │   ├── admin/
│   │   │   │   ├── AccordionAdmin.jsx
│   │   │   │   ├── AdminHome.jsx
│   │   │   │   └── UserInfo.jsx
│   │   │   ├── agent/
│   │   │   │   └── AgentInfo.jsx
│   │   │   ├── common
│   │   │   │   ├── ChatWindow.jsx
│   │   │   │   ├── FooterC.jsx
│   │   │   │   └── Home.jsx
│   │   │   ├── user/
│   │   │   │   ├── Complaint.jsx
│   │   │   │   ├── HomePage.jsx
│   │   │   │   └── Status.jsx
│   │   ├── Images/
│   │   │   └── Image1.png
│   │   ├── app.css
│   │   ├── App.js
│   │   └── index.js
│   │
│   ├── .gitignore
```

6. Running the Application

🔹 1. Start the Backend Server

🔑 Navigate to the backend folder: cd backend

📦 Install backend dependencies: npm install

⚙️ Create a .env file in the backend folder

▶️ **Start the backend server:**

npm start
Server running on http://localhost:5000
MongoDB connected successfully

🔷 **2. Start the Frontend Server**

🔑 **Navigate to the frontend folder: cd ../frontend**

📦 **Install frontend dependencies:**npm install

▶️ **Start the frontend app:**npm start

You should see the app automatically open in your browser at: http://localhost:3000

🧪 **3. Test the Application**

- Visit http://localhost:3000
- Register as:
    - 👦 User → Submit and track complaints
    - 🛠️ Admin → Assign agents and monitor complaints
    - 👩‍💼 Agent → View and resolve assigned complaints
- Use chat and status updates to interact across roles

7. API Documentation

🔐 Authentication APIs

- Register API

Allows a new user, admin, or agent to sign up by providing name, email, password, and role.

- Login API

Authenticates a user and returns a session token (JWT) along with user details.

📝 Complaint Management APIs

- Submit Complaint
  Used by users to file a new complaint by entering details like title, description, and address.
- Get User Complaints
  Retrieves a list of complaints submitted by a specific user.
- Get All Complaints (Admin)
  Admins can view all complaints registered in the system, including their current status and assigned agents.
- Assign Complaint to Agent (Admin)
  Admin can assign a specific complaint to an available agent for handling and resolution.
- Get Assigned Complaints (Agent)
  Agents can view a list of complaints that have been assigned to them.

💬 Messaging / Chat APIs

- Send Message (User or Agent)
  Allows the user or the assigned agent to send a message related to a complaint, enabling communication between them.
- Get Complaint Messages
  Retrieves the chat history or conversation related to a specific complaint.

🛡️ Security and Access Control

- All APIs use JWT-based authentication.
- Role-based authorization is implemented to protect sensitive routes:
  - User: Can submit and view their own complaints.
  - Agent: Can view and respond to assigned complaints.
  - Admin: Can view all complaints and assign them.

8. Authentication

1. User Registration

- **Purpose:**
  Allows a new user (or admin/agent) to create an account by providing their details.
- **Required Data:**
  - Full Name

- Email Address
- Password
- Role (user, admin, or agent)

- **Access:**
  Public (no token required)
- **Response:**
  Returns a success message and a JWT token upon successful registration.

📌 2. User Login

- **Purpose:**
  Authenticates an existing user (or admin/agent) using their email and password.
- **Required Data:**
  - Email Address
  - Password
- **Access:**
  Public (no token required)
- **Response:**
  Returns a success message, user role, and a JWT token for further authenticated requests.

9.User Interface

👩‍💻 User Interface(For Users)

- **Sign Up / Login Page:**
  Users can create an account or log in securely with email and password.
- **User Dashboard:**
  After logging in, users land on a personalized dashboard with the following options:
  - **Submit Complaint**: A form to enter title, description, address, and upload images.
  - **My Complaints**: View a list of complaints the user has submitted.
  - **Status Tracking**: Each complaint shows its current status (e.g., pending, in progress, resolved).
  - **Chat with Agent**: Opens a chat window to communicate directly with the assigned agent.
  - **Logout Button**: Ends session securely.

🛠️ Admin Interface

- **Admin Dashboard:**
  Designed for system administrators with a panel that includes:
  - **View All Complaints**: A table of all registered complaints.
  - **Assign Agents**: Select and assign agents to specific complaints.
  - **View Users and Agents**: Access details of registered users and agents.
  - **Search/Filter Options**: Quickly filter complaints based on status or user.
  - **Logout Button**: Ends session securely.

👩‍💼 Agent Interface

- **Agent Dashboard:**
  Tailored for agents to efficiently manage complaints assigned to them:
  - **Assigned Complaints List**: Displays all complaints the agent is responsible for.
  - **Complaint Details View**: See user-submitted data and attachments.
  - **Chat Panel**: Allows real-time communication with the user.
  - **Status Update Option**: Update complaint status (e.g., in-progress, resolved).
  - **Logout Button**: Ends session securely.

10.Testing :

RightVoice was tested using both manual and automated methods. Manual testing covered core workflows like user registration, complaint submission, agent assignment, and chat functionality. API endpoints were tested using Postman to ensure proper request handling and role-based access. For backend logic, Jest was used to automate tests for authentication, complaint management, and middleware. The system was also checked for responsiveness and usability on multiple devices.

11.Screenshots or Demo

Youtube Link : https://youtu.be/hqyw5ibwM1o?si=0fDA8w9a4Iyj44xx

12.Known Issues :

- 🔄 Real-time notifications via SMS are not yet implemented (only email supported).
- 📱 Mobile responsiveness may be inconsistent on smaller screens.
- 🧩 Chat history does not auto-refresh without manual interaction.
- ⏱️ No loading indicators on some pages, which may cause confusion during API delays.
- 🌐 Multi-language support is not available yet.

13. Future Enhancements

- Mobile application integration (React Native)
- Auto-response based on AI for frequent grievances
- Real-time admin analytics dashboard