**PROJECT ARCHITECTURE AND STRUCTURE**

**🧩 ResolveNow System Architecture**

**🔄 High-Level Flow**

User (Browser)

|



| [HTTP Requests]

v

React Frontend ----------------------

|



| Axios API Calls Static Asset Rendering



v

Node.js + Express Backend <--------> MongoDB

|

| Business Logic + Authentication + Email Notifications

|



v

Nodemailer (SMTP)

**⚙️ Components Breakdown**

**🖼️ Frontend (React + MUI)**

* **React.js**: Single Page Application (SPA) framework for routing and rendering.
* **Axios**: HTTP client to send requests to backend APIs.
* **Material UI**: Component library for a modern, responsive interface.
* **Pages**: Login, Register, Complaint Form, Dashboard, Admin Panel.
* **Components**: ComplaintCard, ChatBox, ComplaintStatus, etc.

**🔙 Backend (Node.js + Express)**

* **Express.js**: Web framework handling routing, middleware, and API endpoints.
* **Controllers**: Contain logic for each route (e.g., login, create complaint).
* **Middleware**: JWT authentication, input validation, error handling.
* **Nodemailer**: Sends emails on complaint status updates.
* **Environment Variables** (.env):
  + MONGO\_URI – DB connection string
  + JWT\_SECRET – for secure token handling
  + EMAIL\_USER & EMAIL\_PASS – for email sending

**🗃️ Database (MongoDB with Mongoose)**

* **User Schema**:
  + name, email, password, role (user/agent/admin)
* **Complaint Schema**:
  + userId, category, description, status, assignedTo, timestamps
* **Message Schema**:
  + complaintId, senderId, text, timestamp

MongoDB is non-relational, allowing for flexible schema and nested structures.

**🔐 Authentication**

* **JWT-based** – Tokens are issued at login and required for protected routes.
* **bcryptjs** – Password hashing for security.
* Tokens stored on client-side in localStorage.

**✉️ Notification System**

* **Nodemailer**:
  + Sends email notifications to users upon:
    - Complaint creation
    - Complaint assignment
    - Status updates (e.g., resolved)
  + Configured via .env SMTP credentials.

**📱 Role-Based Behavior**

| **Role** | **Capabilities** |
| --- | --- |
| User | Register, login, file complaint, view status, chat with agent |
| Agent | Login, view assigned complaints, chat, mark as resolved |
| Admin | Manage all complaints/users/agents, assign complaints |

**🧠 ERD Overview (Entity Relationship)**

User ──────┬────────► Complaint ◄───────┬────── Agent

│ │

└──────► Message ◄──────────┘

**🗂 Project Structure Overview**

Resolve\_online\_complaints/

│

├── backend/ # Backend - Node.js + Express + MongoDB

│ ├── controllers/ # Route handling logic (e.g., complaintController.js)

│ ├── models/ # Mongoose schemas (User.js, Complaint.js, Message.js)

│ ├── routes/ # Express route files (authRoutes.js, complaintRoutes.js)

│ ├── middleware/ # (Optional) Auth middlewares (verifyToken.js)

│ ├── config/ # (Optional) DB config or constants

│ ├── .env # Environment variables (JWT, Mongo URI, Email creds)

│ ├── server.js # Entry point - initializes server & connects DB

│ └── package.json # Backend dependencies & scripts

│

├── frontend/ # Frontend - React + MUI + Axios

│ ├── public/ # Static files and index.html

│ ├── src/

│ │ ├── components/ # Reusable UI components (Navbar, ComplaintCard, ChatBox)

│ │ ├── pages/ # Pages (Login.js, Register.js, Dashboard.js, AdminPanel.js)

│ │ ├── services/ # API calls using Axios (authService.js, complaintService.js)

│ │ ├── App.js # Main app component with routing

│ │ ├── index.js # React entry point

│ │ └── App.css # Custom styles

│ └── package.json # Frontend dependencies & scripts

│

├── README.md # Project overview and usage

└── .gitignore # Node modules, env files, etc.

**💡 Suggested Enhancements (Optional Folders)**

You can further modularize with:

* backend/utils/ – common helper functions (email sending, token generation)
* frontend/hooks/ – custom React hooks (e.g., useAuth, useFetch)
* frontend/context/ – for managing global state (auth, theme, etc.)
* backend/validators/ – for request validation using Joi or express-validator