

# ResolveNow: Online Complaint Registration and Management System

Final Project Report

## 1. Introduction

- **Project Title:** ResolveNow – Online Complaint Registration and Management System
- **Team Members:**
  - **Team ID:** LTVIP2025TMID59341
  - **Team Leader:** Konduri Pavithra
  - **Team Member:** Katuru V Venkata Sairama Anirudh
  - **Team Member:** Katari Ashok
  - **Team Member:** Katikala Kiran Mai

## 2. Project Overview

- **Purpose:**

ResolveNow is a role-based complaint management system that enables Users, Admins, and Engineers to register, track, and resolve complaints in a structured and secure environment. It improves accountability and efficiency in grievance redressal workflows.
- **Features**
  - User registration and authentication
  - Role-based dashboards for Admins and Engineers
  - Complaint submission and assignment
  - Real-time status tracking
  - In-app messaging system
  - Notification via Email
  - Clean and responsive UI with Bootstrap

### 3. Architecture

- **Frontend:**  
Built with React.js, using React Router for page navigation, Context API for global state management, and styled using Bootstrap 5. Axios is used for communicating with the backend API.
- **Backend:**  
Developed using Node.js and Express.js, which expose RESTful API endpoints. Authentication is handled with JWT, password security with bcryptjs, and environment management via dotenv.
- **Database:**  
Uses MongoDB, with schemas for Users, Complaints, and Messages. Mongoose ODM is used to define and interact with the database collections.

### 4. Setup Instructions

- **Prerequisites:**
  - Node.js
  - MongoDB
  - Git
  - npm (Node Package Manager)
- **Installation Steps:**
  - # Clone the repository
  - git clone  
<https://github.com/thesocalledhonouredone/ResolveNow-Complaint-Management-System.git>
  - # Navigate to ResolveNow-Complaint-Management-System
  - Npm install

- **Environment Setup:**

Create a `.env` file in the backend folder and define:

`MONGO_URI = your_mongo_db_connection_string`

`JWT_SECRET = your_jwt_secret`

`PORT = 5000`

## 5. Folder Structure

- **Client**

(Frontend):

`frontend/`

|— `public/`

|— `src/`

| |— `components/`

| |— `pages/`

| |— `context/`

| |— `App.js`

| |— `index.js`

- **Server (Backend):**

`backend/`

|— `controllers/`

|— `models/`

|— `routes/`

|— `middleware/`

|— `config/`

|— `server.js`

## 6. Running the Application

- **Frontend:**  
`cd frontend`  
  
`npm start`
- **Backend:**  
`cd backend`  
  
`npm start`

Application runs on:

- **Frontend:** `http://localhost:3000`
- **Backend API:** `http://localhost:5000`

## 7. API Documentation

### User Routes

- POST `/api/register` → Create new user
- POST `/api/login` → Login and receive JWT token
- GET `/api/user/profile` → Get logged-in user profile

### Complaint Routes

- POST `/api/complaints` → Submit complaint
- GET `/api/complaints` → View user complaints
- PATCH `/api/complaints/:id` → Update complaint status

### **Admin Routes**

- GET /api/admin/complaints → View all complaints
- PATCH /api/admin/assign/:id → Assign complaint to engineer

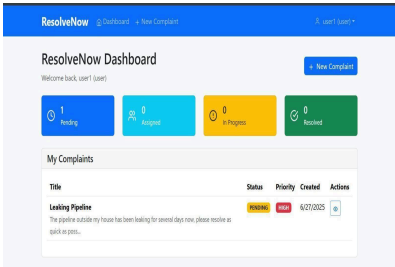
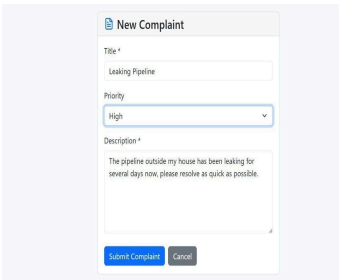
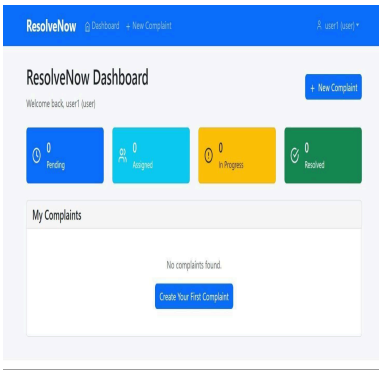
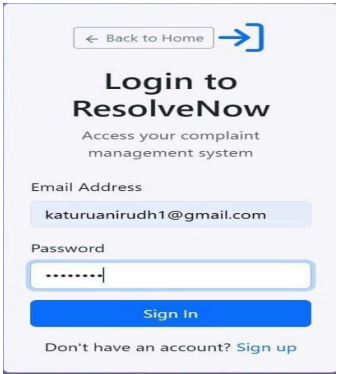
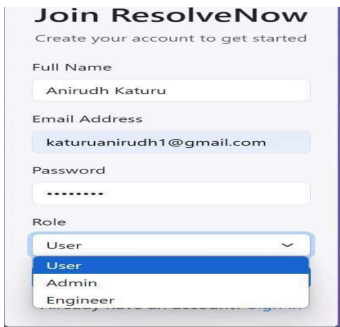
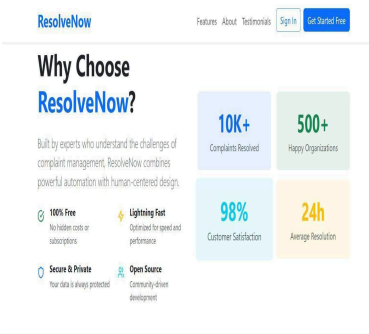
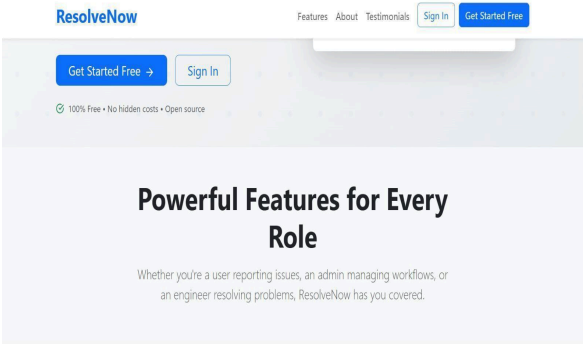
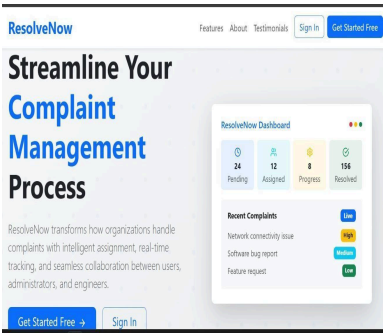
## **8. Authentication**

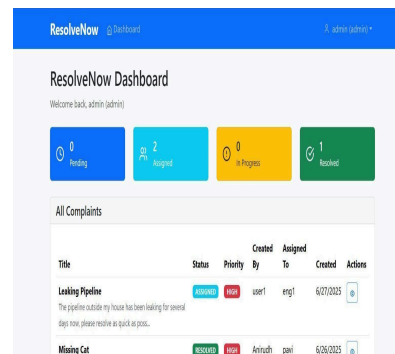
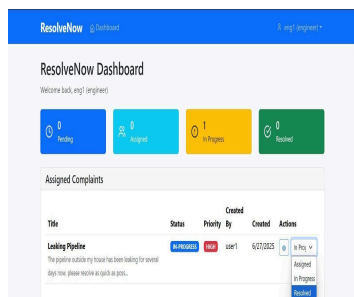
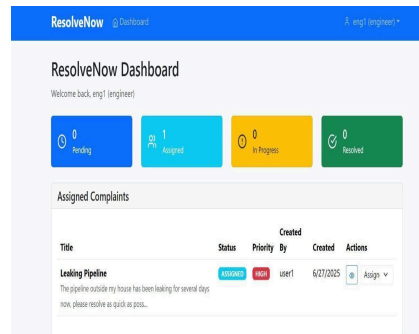
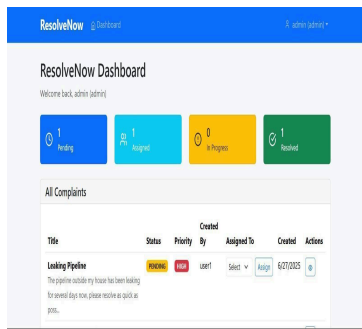
- **Method Used:**  
JSON Web Tokens (JWT)
- **How it Works:**
  - User logs in with email/password
  - Server generates a token using JWT
  - Token is stored in localStorage (frontend)
  - Protected routes are verified using middleware on the backend

## **9. User Interface**

- **UI Technology:**  
React + Bootstrap 5
- **Features:**
  - Responsive Dashboard
  - Complaint Submission Form
  - Admin Panel to assign complaints
  - Chat window for engineer-user interaction

Output:





## 10. Testing

- **Tools Used:**

- Postman / Thunder Client (API Testing)
- Chrome DevTools (UI testing)
- Manual Testing for User/Engineer/Admin roles

- **Scenarios Tested:**

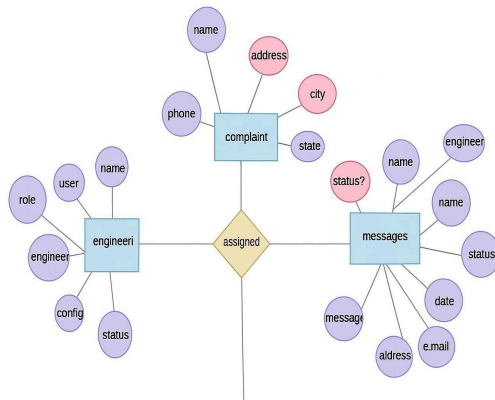
- Multiple users submitting complaints
- Role-based access restrictions
- Chat system functionality
- Data integrity on MongoDB updates



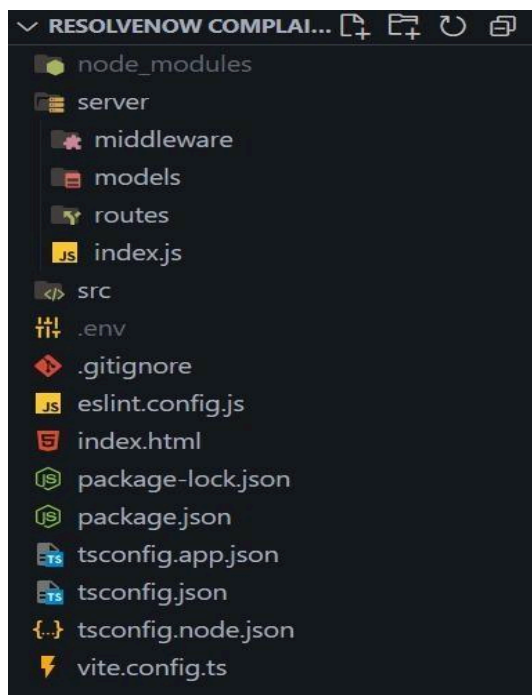
## 11. Screenshots or Demo

- screenshots  
:

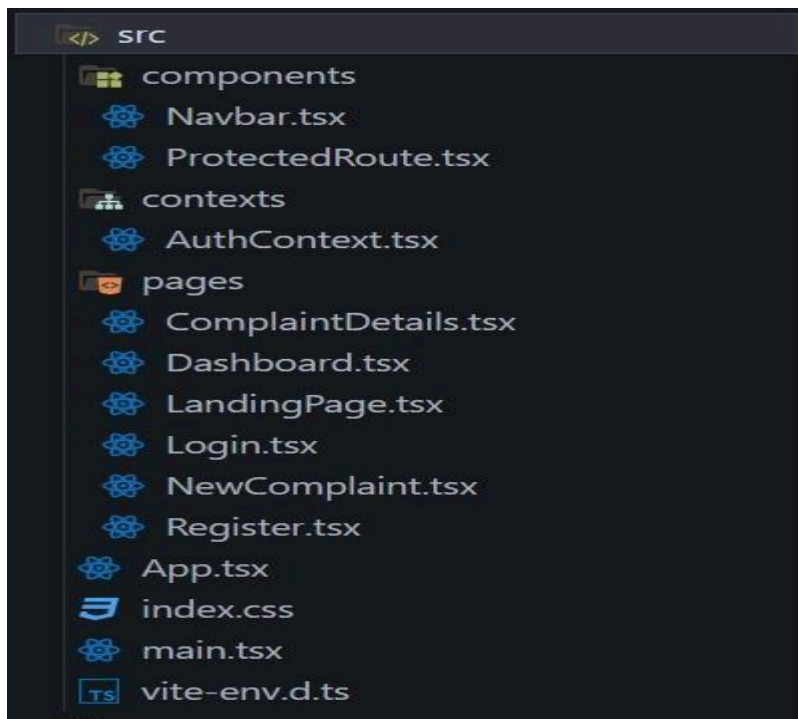
### ER Diagram



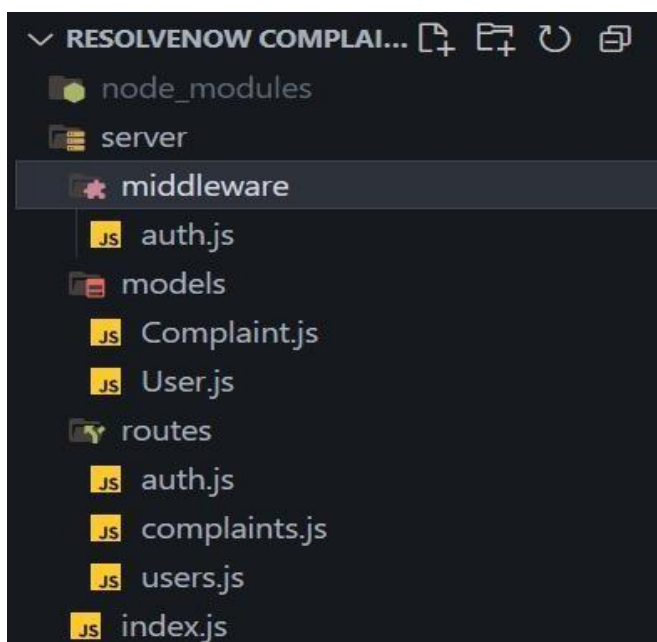
### 1. Overall Project Structure



## 2. Frontend



## 3. Backend



- GitHub Repo: [ResolveNow GitHub](#)
- Localhost Demo: <http://localhost:5173>

## **12. Known Issues**

- No mobile app version yet
- Chat feature doesn't support file/image attachments
- No dark mode theme included

## **13. Future Enhancements**

- Native mobile app support
- AI chatbot for FAQs and issue categorization
- Real-time push notifications using WebSockets
- Multi-language interface support
- Admin Analytics Dashboard (Complaint trends, resolution time)

