

ResolveNow: Your Platform for Online Complaints

Project Documentation Format

1. Introduction

- Project Title: *ResolveNow – Your Platform for Online Complaints*
 - Team Members: List the names of team members and their specific roles (e.g., Frontend Developer, Backend Developer, UI/UX Designer).
-

2. Project Overview

- Purpose: ResolveNow is designed to streamline the process of submitting, tracking, and managing complaints across various sectors, providing users with a simple and transparent grievance redressal platform.
- Features:
 - User registration and login
 - Complaint submission and status tracking
 - Admin dashboard for complaint resolution
 - Email/SMS notifications
 - Feedback and rating system

3. Architecture

- Frontend: Built using React.js with responsive design and intuitive navigation for end-users and administrators.
 - Backend: Developed with Node.js and Express.js, handling routing, business logic, and API integration.
 - Database: Utilizes MongoDB to store user profiles, complaint records, status updates, and feedback securely.
-

4. Setup Instructions

- Prerequisites:
 - Node.js
 - MongoDB
 - Git
 - Installation:
 1. Clone the repository.
 2. Run `npm install` in both the client and server directories.
 3. Set environment variables (`.env` file).
 4. Start MongoDB service locally or connect via MongoDB Atlas.
-

5. Folder Structure

- Client:
 - `/src` contains components, pages, and context for state management.
 - `/public` contains static files.
 - Server:
 - `/routes` defines API endpoints.
 - `/controllers` handles logic for each route.
 - `/models` defines Mongoose schemas.
-

6. Running the Application

- Commands:
 - Frontend: `npm start` (from the `client` directory)
 - Backend: `npm start` (from the `server` directory)
-

7. API Documentation

- Each RESTful endpoint is documented with:
 - Method (GET, POST, etc.)
 - Route URL
 - Parameters and body data
 - Sample request/response payloads

8. Authentication

- User sessions managed via JWT tokens.
 - Middleware used to protect restricted routes (e.g., Admin dashboard).
 - Login credentials are securely hashed using bcrypt.
-

9. User Interface

- Clean, accessible, and mobile-responsive interface.
 - Pages include:
 - Homepage
 - Complaint form
 - User dashboard
 - Admin panel
 - Feedback submission screen
-

10. Testing

To ensure the platform performs reliably and meets user expectations, the following testing strategies and tools were implemented:

- **Automated Testing:** Unit tests and integration tests were developed using tools like Jest and Mocha.
- **Manual Testing:** Functional and usability tests were conducted to verify complaint submission, tracking, and resolution workflows.
- **Performance Testing:** Tools such as Apache JMeter were used to simulate user loads and test system responsiveness.

11. Screenshots or Demo

To help users and stakeholders understand the platform features, visual aids and demonstrations are provided:

- Screenshots: Key UI components and workflow steps are captured to showcase the complaint submission process, dashboard overview, and resolution status.
- Live Demo: A link to a functional demo is available [Insert Demo Link] to explore real-time platform usage.

12. Known Issues

While the platform is fully operational, the following issues have been identified and are under review:

- Some users experience intermittent login session expiration.
- Limited browser support for legacy Internet Explorer versions.
- Occasional delays in email notifications for complaint status updates.

13. Future Enhancements

To improve user experience and expand functionality, the following enhancements are planned:

- **Mobile App Integration:** Develop dedicated mobile apps for Android and iOS.
- **Multilingual Support:** Enable complaint registration and resolution in multiple languages.
- **AI-Powered Complaint Categorization:** Automatically classify complaints for faster routing and resolution.
- **Real-Time Chat Support:** Integrate chatbots and live agent support for instant help.