

Online Complaints Management System - Documentation

1. Introduction

Project Title: Online Complaints Management System

Team Members:

- Veera Kumari - Frontend Developer

2. Project Overview

Purpose:

To develop a platform where users can file complaints online and track their status while administrators manage and resolve them efficiently.

Features:

- User and Admin login
- Complaint filing and listing
- Status updates
- Secure authentication using JWT
- Role-based access control
- REST API communication

3. Architecture

Frontend:

Built using React.js with React Router and Axios. Styled with Bootstrap/CSS.

Backend:

Developed with Node.js and Express.js handling authentication, complaint management APIs.

Database:

MongoDB used via Mongoose to store user and complaint data.

Online Complaints Management System - Documentation

4. Setup Instructions

Prerequisites:

- Node.js
- MongoDB
- Git
- Code Editor

Installation:

1. Clone the repo
2. Install backend & frontend dependencies
3. Setup .env for backend
4. Run npm start for both client and server

5. Folder Structure

Client:

client/

src/

 components/

 pages/

 services/

 App.js

Server:

server/

 controllers/

 models/

 routes/

 middleware/

 server.js

Online Complaints Management System - Documentation

6. Running the Application

Frontend:

```
$ cd client
```

```
$ npm start
```

Backend:

```
$ cd server
```

```
$ npm start
```

7. API Documentation

POST /api/auth/register - Register user

POST /api/auth/login - Login user

POST /api/complaints - Create complaint

GET /api/complaints/my - User's complaints

GET /api/admin/complaints - All complaints (Admin)

PUT /api/complaints/:id - Update status

DELETE /api/complaints/:id - Delete complaint

8. Authentication

JWT is used for login token generation.

Passwords are hashed using bcryptjs.

Middleware restricts routes based on user roles.

9. User Interface

Includes:

- Login/Register Pages

- User Dashboard

- Admin Panel

(Screenshots to be added manually)

Online Complaints Management System - Documentation

10. Testing

Manual testing using Postman for APIs.

Frontend tested via browser with edge case validation.

11. Screenshots or Demo

(Insert screenshots or link to live demo)

12. Known Issues

- No email notifications yet
- No file upload support

13. Future Enhancements

- Add email/SMS notifications
- Document uploads
- Complaint categories and filters
- Admin analytics
- Mobile app with React Native