Online Complaint Management System using MERN Stack

# 1. Introduction

Project Title: Online Complaint Management System using mern stack

**TEAM MEMBERS:**

1. Surekha.S
2. Rebecasuji.A
3. Abirami
4. Vaishanavi M.R

# 2. Project Overview

Purpose:

The Complaint Management System is designed to streamline the process of lodging, tracking, and managing customer complaints. It ensures that issues are addressed in a timely and efficient manner, improving overall customer satisfaction.

Features:

- User-friendly interface for submitting complaints  
- Role-based access for admins and users  
- Real-time status updates on complaint resolution  
- Secure authentication and authorization  
- Dashboard for admin analytics and reporting

# 3. Architecture

Frontend:

The frontend is built using React.js, offering a responsive and intuitive user interface.

Backend:

The backend is powered by Node.js and Express.js, handling business logic, API endpoints, and user authentication.

Database:

MongoDB is used for data storage, providing a flexible schema to handle various complaint details.

# 4. Setup Instructions

Prerequisites:

- Node.js  
- MongoDB  
- npm or yarn

Installation:

1. Clone the repository.  
2. Navigate to the project directory.  
3. Run `npm install` in both `client` and `server` folders.  
4. Set up environment variables for MongoDB URI, JWT Secret, etc.  
5. Start the application using `npm start` for both client and server.

# 5. Folder Structure

Client:

- src/components  
- src/pages  
- src/context  
- src/services

Server:

- routes  
- controllers  
- models  
- middleware  
- config

# 6. Running the Application

Client:

Navigate to the `client` directory and run `npm start`.

Server:

Navigate to the `server` directory and run `node server js’

# 7. API Documentation

GET /api/complaints - Retrieve all complaints  
POST /api/complaints - Submit a new complaint  
GET /api/complaints/:id - Get complaint by ID  
PUT /api/complaints/:id - Update complaint status  
DELETE /api/complaints/:id - Delete complaint

# 8. Authentication

The system uses JWT (JSON Web Tokens) for secure authentication. Admin and User roles are defined, enabling access control.

# 9. User Interface

Screenshots or demo links will showcase the complaint submission form, admin dashboard, and analytics view.

# 10. Testing

Testing Strategy:

- Unit testing with Jest  
- API testing using Postman  
- Integration testing to ensure system modules work together

# 11. Screenshots or Demo

Provide screenshots of key functionalities or a demo video link.

# 12. Known Issues

No known issues at this point. Continuous testing and debugging are ongoing.

# 13. Future Enhancements

- Integration with third-party CRM systems  
- AI-based sentiment analysis for complaint categorization  
- Mobile app version for Android and iOS  
- Enhanced analytics using data visualization tools