**Full Stack Development with MERN**

**Team ID**: LTVIP2025TMID57433  
**Project Documentation**

**1. Introduction**

* **Project Title**: ComplaintCare: Online Complaint Registration and Management System
* **Team Members**:
  + **Manam Hari Krishna** – *Full Stack Developer*: Oversees complete system development, authentication, and deployment.
  + **Lochana Yerra** – *Frontend Developer*: Designs and builds the UI using React.js, ensuring user-friendly complaint submission and status tracking.
  + **Kurra Sai Durga Rajeswari** – *Backend Developer*: Develops RESTful APIs using Node.js/Express.js and integrates business logic for complaint handling.
  + **Kota Asha** – *Database Administrator*: Manages MongoDB schemas for users, complaints, assignments, and messages.

**2. Project Overview**

* **Purpose**: ComplaintCare streamlines the process of registering and managing public complaints by enabling digital submission, tracking, and resolution.
* **Features**:
  + User and Agent authentication
  + Complaint registration and status tracking
  + Admin dashboard for agent assignment
  + Agent dashboard for complaint updates and resolution
  + Real-time chat between users and agents
  + Responsive interface for all user roles

**3. Architecture**

* **Frontend**:
  + Built with **React.js** using functional components and hooks
  + Routing handled via **React Router**
  + Bootstrap and custom CSS for responsive design
* **Backend**:
  + Developed using **Node.js** and **Express.js**
  + RESTful APIs for complaints, users, and chat messaging
  + Authentication and role-based access control
* **Database**:
  + **MongoDB** with Mongoose
  + Schemas: User, Complaint, AssignedComplaint, Message

**4. Setup Instructions**

* **Prerequisites**: Node.js, MongoDB, Git

**5. Folder Structure**

* **Frontend**:
  + components/, pages/, context/, App.jsx, index.js
* **Backend**:
  + routes/, models/, controllers/, middleware/, config/

**6. Running the Application**

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

**7. API Documentation**

* POST /register – New user registration
* POST /login – User or Agent login
* POST /complaint – Submit a complaint
* GET /complaints/:userId – Fetch user complaints
* PUT /complaint/status/:id – Update complaint status
* POST /assign – Admin assigns complaint to agent
* POST /message – Send message in chat
* GET /messages/:complaintId – Fetch chat messages

**8. Authentication**

* **JWT-based authentication**
* Tokens stored in **localStorage**
* Role-based access control for user, agent, and admin routes

**9. User Interface**

* Built using **React + Bootstrap**
* Responsive for all roles: User, Agent, and Admin
* UI Screens:
  + Login/Signup
  + User Dashboard
  + Complaint Registration
  + Status Tracker
  + Admin Panel
  + Agent Dashboard with Chat Window

**10. Testing**

* **Manual Testing**:
  + Browser testing across different roles
  + Validation for form fields and routing
* **API Testing**:
  + Using **Postman** for endpoints
* **Future Testing**:
  + Implement **Jest** and **Supertest** for automated backend tests

**11. Screenshots or Demo**

* **Demo Video**: *<https://youtu.be/7Z5Xxpf0csQ?si=Y4XfENaMlsBEqYA->*

**12. Known Issues**

* Real-time chat may require refresh for new messages
* No admin email notification system implemented yet
* Mobile view for chat window may require additional CSS tuning

**13. Future Enhancements**

* Add email or SMS notifications for updates
* Real-time complaint status push notifications
* File attachments in complaint submission
* Role-based analytics dashboard for admin
* Progressive Web App (PWA) version for mobile users