

Full Stack Development with MERN

Title: Online Complaint Registration and Management System (OCRMS)

Team Members:

1. Kishore R
2. Prasanna Kumar S
3. Mohamed Sheik M
4. Nishanth A

Project Overview

Purpose:

OCRMS provides users with a platform to lodge and track complaints across various sectors. It enables organizations to manage, resolve, and respond to complaints efficiently, ensuring transparency and user satisfaction.

Features:

1. **User-Friendly Complaint Registration:**
Lodge complaints with an intuitive form interface. Include complaint details, category, and preferred resolution type.
2. **Personalized Profiles:**
 - Users can view complaint history, update personal details, and track resolutions.
 - Admins and employees can manage user complaints.
3. **Role-Based Dashboards:**
 - **Users:** Access complaint statuses and feedback history.
 - **Admins:** Manage complaints, monitor employee performance, and generate reports.
 - **Employees:** Resolve assigned complaints and communicate progress.
4. **Notification System:**
 - Instant notifications for updates, responses, and resolutions.
 - SMS or email integration for communication.
5. **Analytics & Insights:**
Track resolution times, complaint trends, and user satisfaction metrics.

Architecture

1. Frontend:

Built using **React.js**, providing a responsive and dynamic interface. Key modules include:

- **User Module:** Complaint registration, history, and notifications.
- **Admin Module:** Dashboard for managing complaints and users.
- **Employee Module:** Interface for resolving complaints.

2. Backend:

Developed using **Node.js** and **Express.js**, ensuring efficient server-side operations with RESTful APIs. Key features include:

- **Role-Based Access Control:** Secure access for users, admins, and employees.
- **Middleware:** Authentication (JWT), error handling, and request validation.
- **Real-Time Updates:** Integrating WebSockets for instant notifications.

3. Database:

MongoDB for scalable and flexible data management.

- **User Collection:** Stores user profiles and access roles.
- **Complaint Collection:** Tracks complaints, statuses, and updates.
- **Activity Logs:** Maintains complaint and communication histories.

Setup Instructions

Prerequisites:

- **Node.js:** v14+
- **MongoDB:** v4.4+
- **React:** v18+
- **npm:** v7+

Steps to Set Up:

Clone the Repository:

bash

Copy code

```
git clone https://github.com/username/OCRMS.git
```

```
cd OCRMS
```

Install Dependencies:

bash

Copy code

```
cd Frontend
```

```
npm install
```

```
cd ../Backend
```

```
npm install
```

Set Up Environment Variables:

Create a `.env` file in the server directory with the following details:

env

Copy code

```
PORT=5000
```

```
MONGO_URI=mongodb+srv://<username>:<password>@cluster.mongodb.net/OC
```

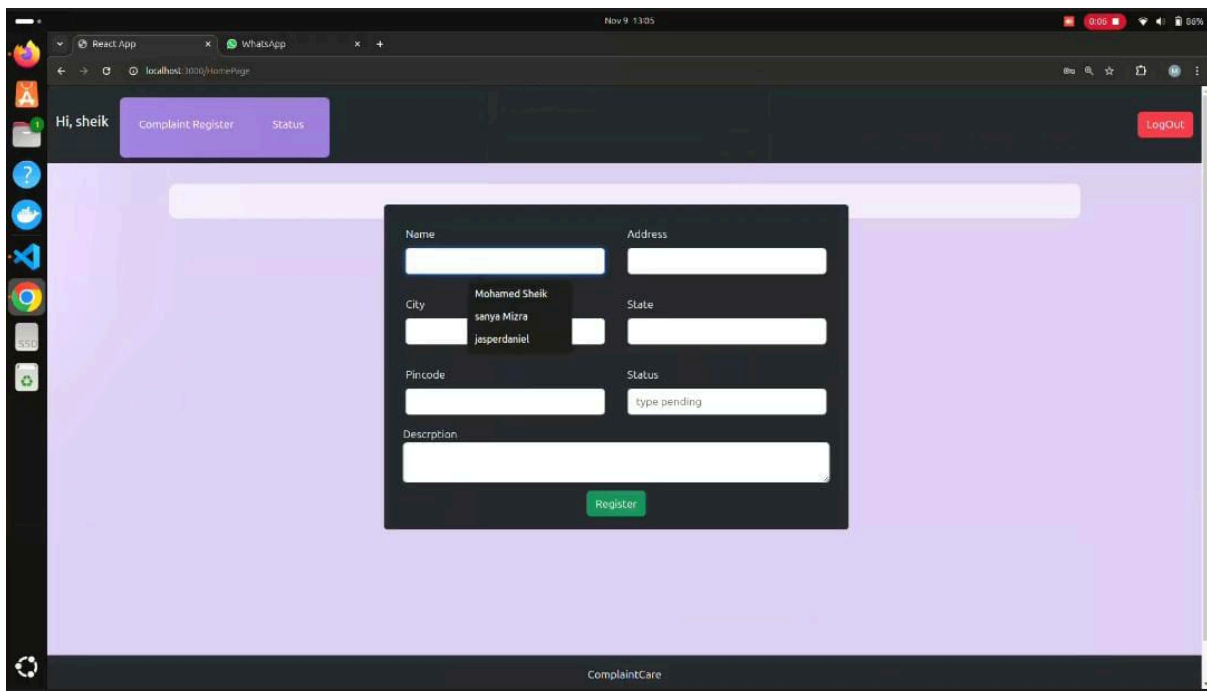
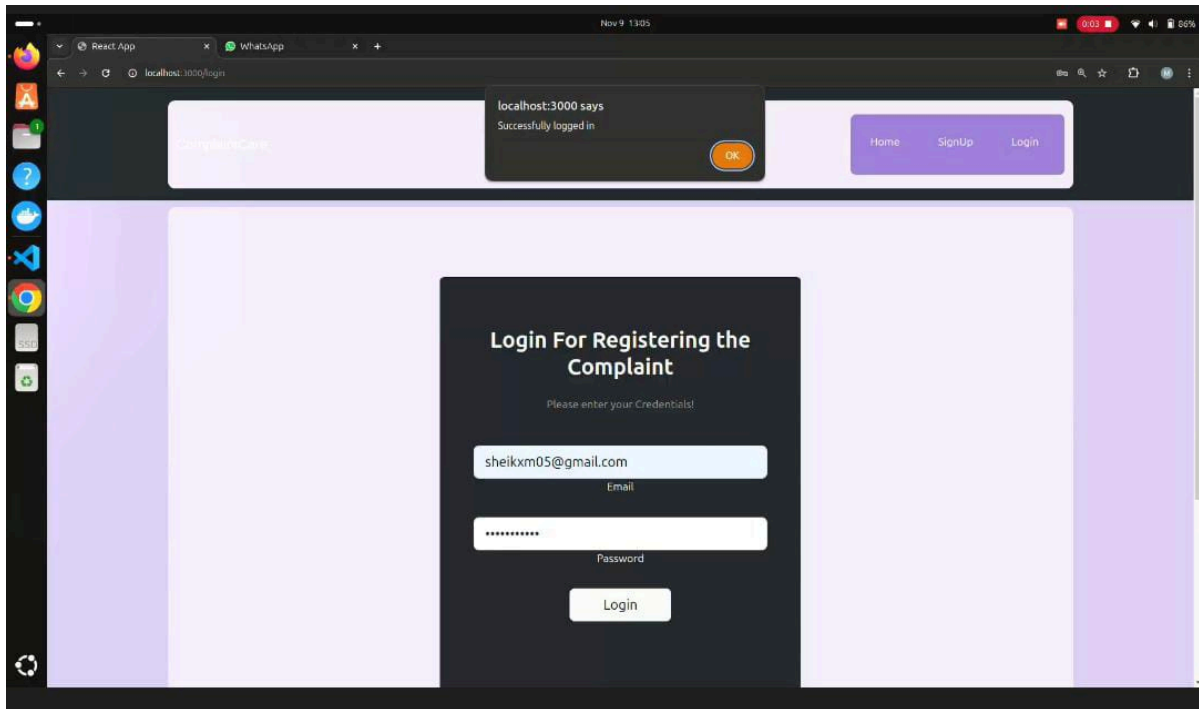
```
RMS
```

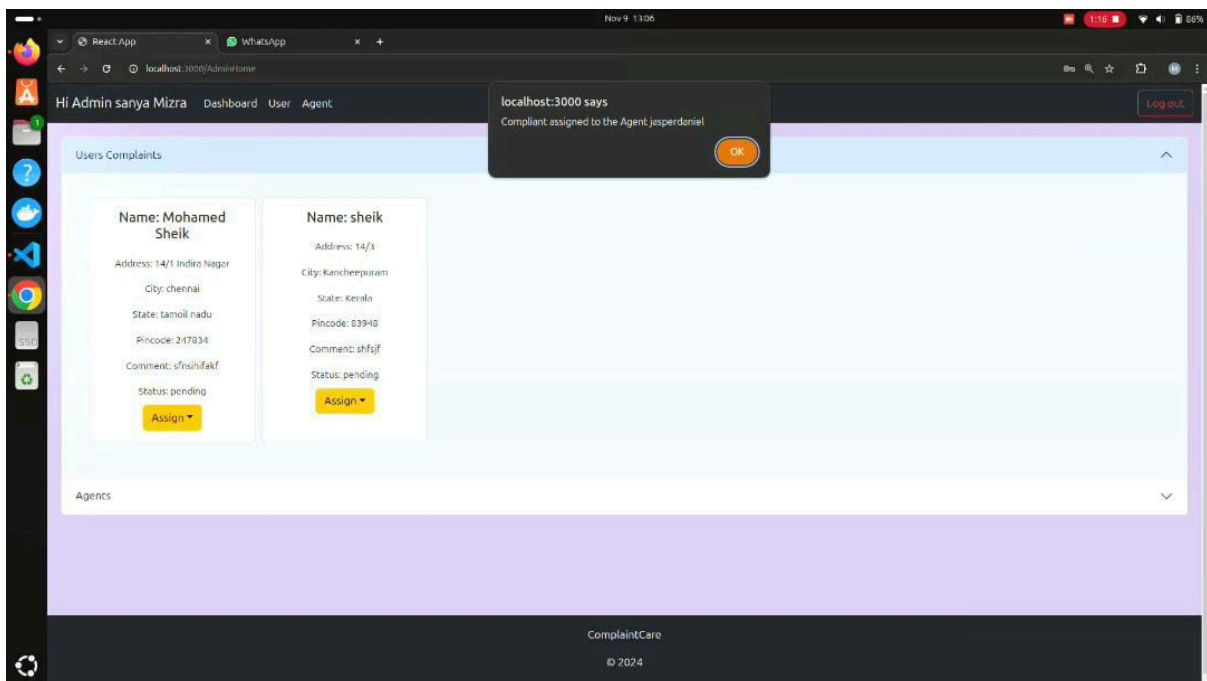
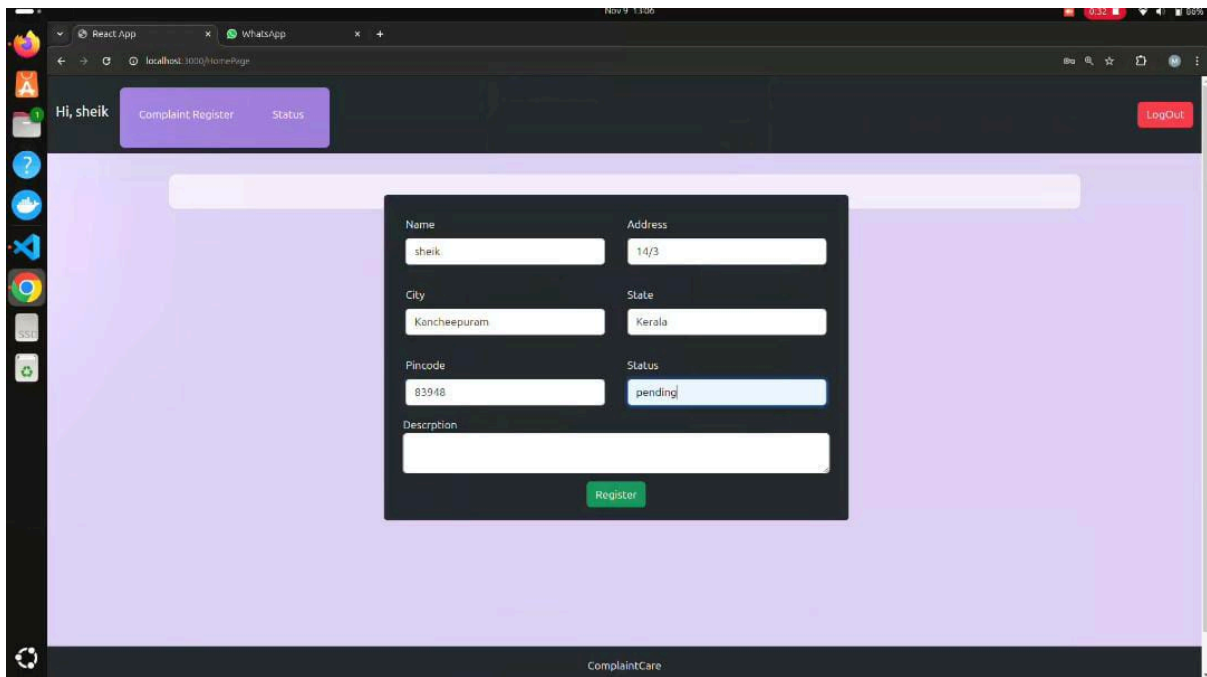
```
JWT_SECRET=your_secret_key
```

Testing Strategy

1. **Unit Testing:**
Test individual modules like complaint forms, APIs, and database queries.
2. **Integration Testing:**
Ensure smooth communication between the frontend, backend, and database.
3. **End-to-End Testing:**
Simulate user actions like lodging and tracking complaints.
4. **Performance Testing:**
Assess response times under different complaint volumes.
5. **Security Testing:**
Validate data encryption, secure login, and role-based access control.

SCREENSHOTS





DEMO VIDEO LINK :

https://drive.google.com/file/d/18a8ujrhHN2P1eGLK1ZKndF5_w8nw_pOj/view?usp=sharing

Future Enhancements

1. **AI-Powered Resolution Suggestions:**
Suggest resolutions based on complaint history and trends.
2. **Multilingual Support:**
Expand accessibility with language options.
3. **Mobile Application:**
Develop Android and iOS versions for user convenience.
4. **Advanced Analytics:**
Detailed performance and satisfaction reports for organizations.