Online Complaint Registry - Project Report

1. Title and Abstract

Title: Online Complaint Registry

Abstract:

The Online Complaint Registry is a web-based system designed to simplify the complaint registration and management process. The application allows users to submit complaints, track their statuses, and communicate with agents, while providing admin users the ability to manage complaints and users efficiently. Built using the MERN stack (MongoDB, Express.js, React.js, and Node.js), the application includes features such as user authentication, complaint submission, real-time tracking, and user-agent interaction.

2. Introduction

Objective:

The objective of this project is to develop a platform where users can submit complaints, track their resolution status, and communicate with agents. The goal is to streamline complaint management by providing both users and admins with an intuitive and efficient system.

3. Literature Review

Existing Solutions:

Similar complaint management systems are widely used by businesses and public services, such as the Indian government's grievance redressal systems. These systems often provide users with the ability to submit complaints online and track their status. However, many lack features like real-time communication or efficient backend management. This project incorporates some of the best features from existing platforms while focusing on a user-friendly interface and efficient backend design.

4. System Requirements

Hardware Requirements:

- Processor: Intel i3 or above

- RAM: 4GB minimum

- Storage: 500 MB for app data

Software Requirements:

- Frontend: React.js

- Backend: Node.js, Express.js

- Database: MongoDB

- Tools: NPM, GitHub, Postman (for API testing)

5. Methodology

Modules:

1. **User Module**: Allows users to register, log in, submit complaints, and track status.

2. **Admin Module**: Provides admin users with the ability to view, manage, and resolve complaints.

3. **Complaint Submission**: Allows users to submit complaints with necessary details.

4. **Complaint Status Tracking**: Users and agents can track the current status of complaints.

5. **Real-Time Communication**: Enables users to chat with agents for clarification or updates on complaints.

Tools and Technologies Used:

- **Frontend**: React with React Router, Axios for API calls

- **Backend**: Node.js, Express.js, JWT for authentication

- **Database**: MongoDB for storing complaints, users, and related data

- **Real-Time Communication**: Socket.io for real-time chat functionality

6. System Design

Architecture:

The system follows a client-server architecture. The frontend is built with React and communicates with the backend via REST APIs created using Node.js and Express.js. MongoDB is used for storing users, complaints, and related data.

Database Schema:

- **Users**: `_id`, `username`, `email`, `password`, `role` (user/admin)

- **Complaints**: `_id`, `userId`, `title`, `description`, `status`, `createdAt`

- **Messages**: `_id`, `complaintId`, `senderId`, `content`, `timestamp`

Flowchart:

Illustrates the flow: User Registration → Login → Submit Complaint → Track Status → Admin manages complaints → Real-time chat.

7. Implementation

Frontend Implementation:

The frontend is built with React.js and includes components for user registration, login, complaint submission, status tracking, and real-time chat. React Router manages the routing, and Axios is used for making API calls to the backend.

Backend Implementation:

The backend is implemented with Node.js and Express.js, handling RESTful APIs for user authentication, complaint submission, and management. MongoDB and Mongoose are used to manage data storage and validation.

Admin Panel Implementation:

Admin users can view, update, and resolve complaints. They also have the ability to track the status of complaints and communicate with users through the real-time chat system.

8. Key Functionalities

User Features:

- **Registration & Login**: Allows users to create an account and log in.

- **Complaint Submission**: Users can submit complaints, providing necessary details.

- **Complaint Status Tracking**: Track the status of complaints submitted by the user.

- **Real-Time Chat**: Communicate with agents about complaints and updates.

Admin Features:

- **Complaint Management**: Admins can view, update, and resolve complaints.

- **Real-Time Communication**: Admins can chat with users to provide updates on complaints.

9. Testing

Testing Types:

- **Unit Testing**: Testing individual components and API endpoints.

- **Integration Testing**: Testing interactions between the frontend, backend, and database.

- **User Acceptance Testing (UAT)**: Ensures the system meets user requirements and expectations.

Testing Tools:

- Jest for frontend testing

- Postman for backend API testing

- Manual testing for UI validation

10. Results

Performance Metrics:

The application performs efficiently with an average API response time of under 200ms for most requests. The real-time chat functionality operates smoothly, even with multiple users online.

User Feedback:

Test users found the complaint submission process easy to use, and they appreciated the ability to track complaint status and communicate with agents in real-time.

11. Challenges and Solutions

- **Challenge**: Ensuring real-time communication between users and agents.

  - **Solution**: Implemented Socket.io for real-time chat functionality, ensuring efficient communication.

- **Challenge**: Handling user authentication securely.

  - **Solution**: Used JWT for secure authentication and authorization, ensuring that users only access their own complaints.

12. Future Enhancements

- **Admin Dashboard**: Add advanced features like filtering complaints based on status or type.

- **Notifications**: Implement push notifications to alert users when their complaint status changes.

- **Mobile Support**: Develop mobile applications for iOS and Android for on-the-go access.

13. Conclusion

The **Online Complaint Registry** project successfully demonstrates the use of modern web technologies to build a scalable and efficient complaint management system. It provides a user-friendly interface, real-time communication, and effective complaint tracking for users, with robust management features for admins.

---

14. Project Setup for GitHub

Project Setup Instructions

Frontend

1. **Navigate to the frontend directory**:

   ```bash
   cd frontend
   ```

2. **Install dependencies**:

   ```bash
   npm install
   ```

3. **Start the frontend server**:

   ```bash
   npm start
   ```

```
```

Backend

1. **Navigate to the backend directory**:

   ```bash
   cd backend
   ```

2. **Install dependencies**:

   ```bash
   npm install
   ```

3. **Start the backend server**:

   ```bash
   npm start
   ```

Admin Panel

1. **Navigate to the admin panel directory**:

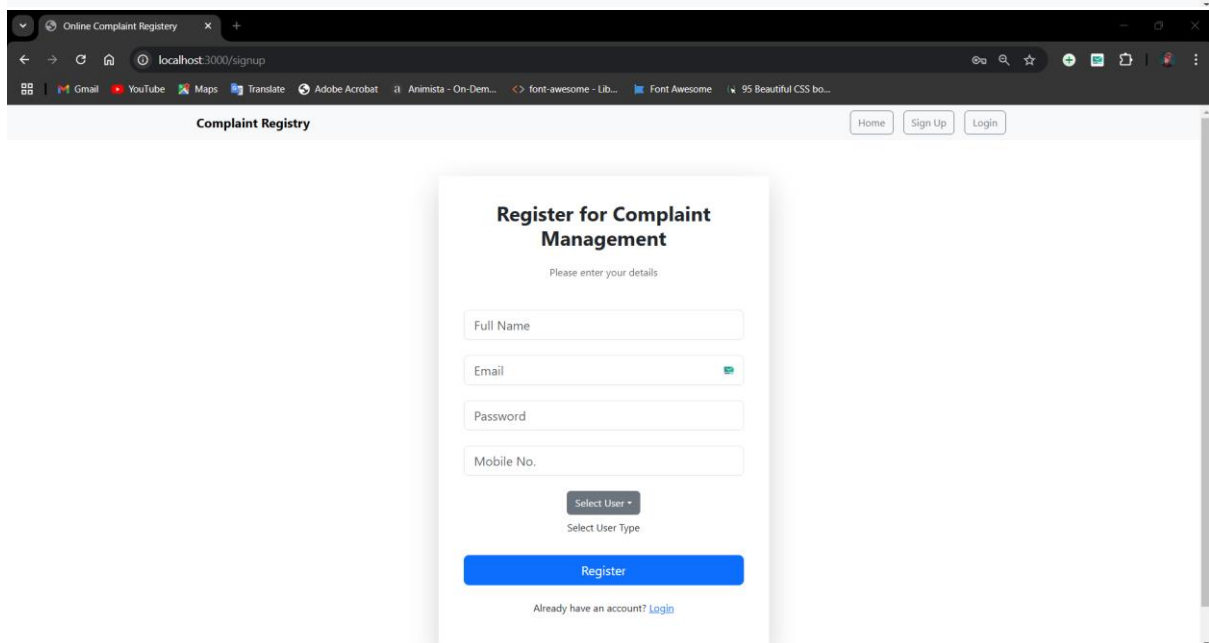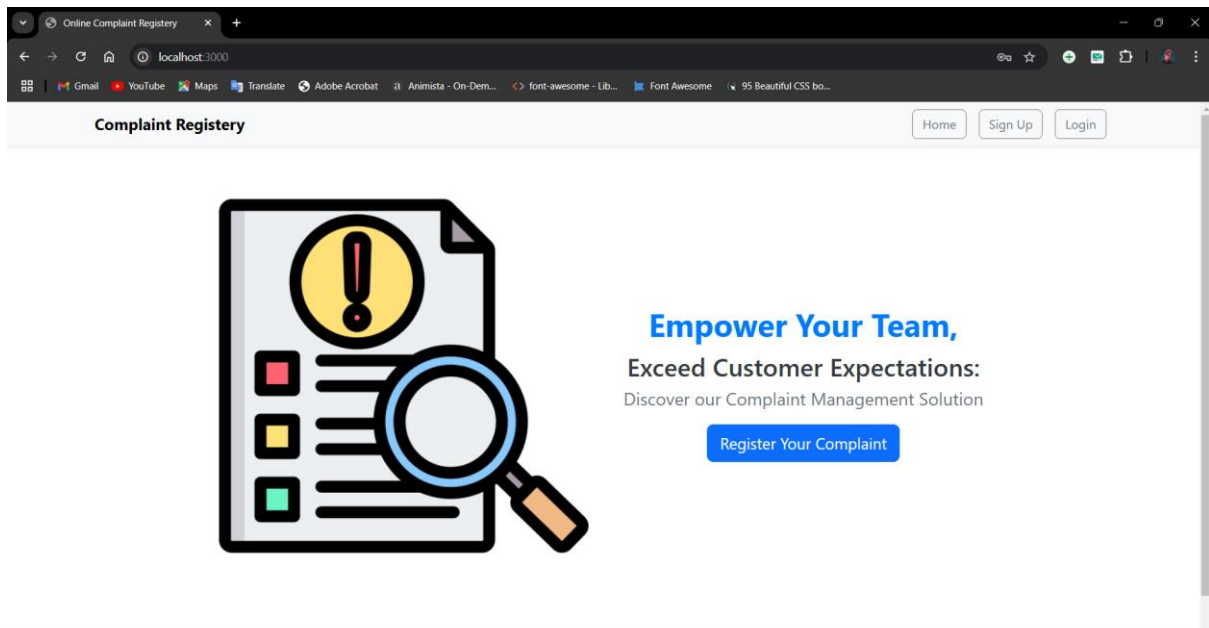   ```bash
   cd admin
   ```

2. **Install dependencies**:

   ```bash
   npm install
   ```

3. **Start the admin panel**:

   ```bash
```

```
npm run dev
```
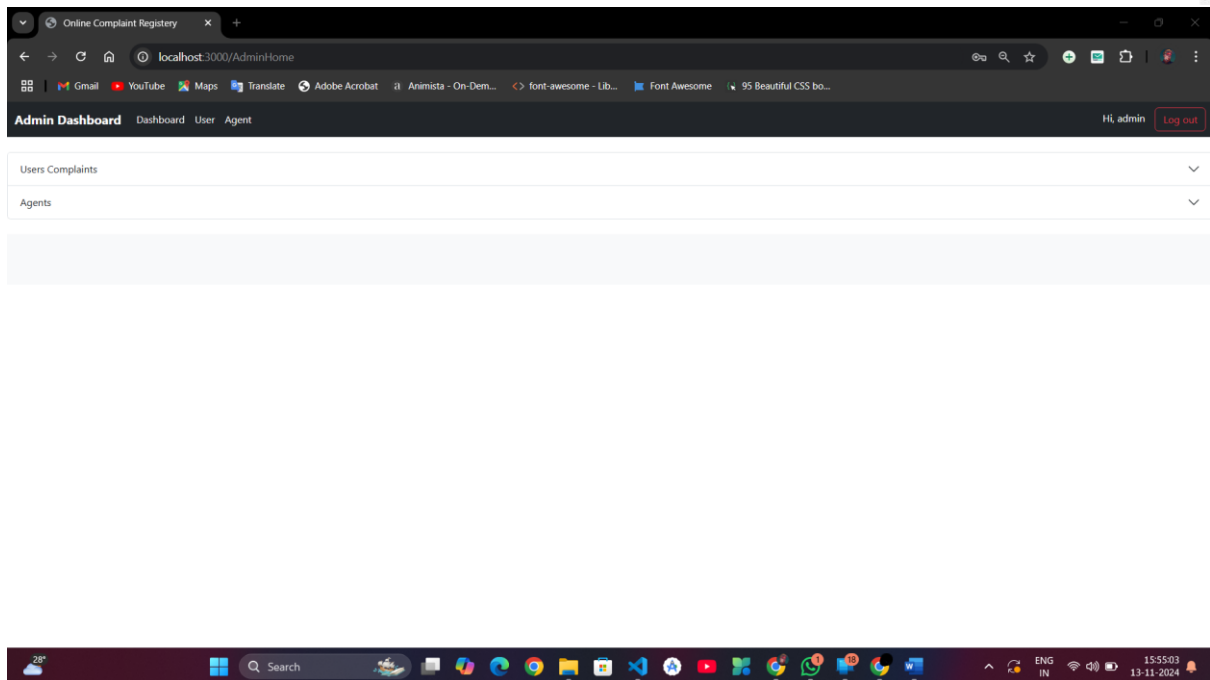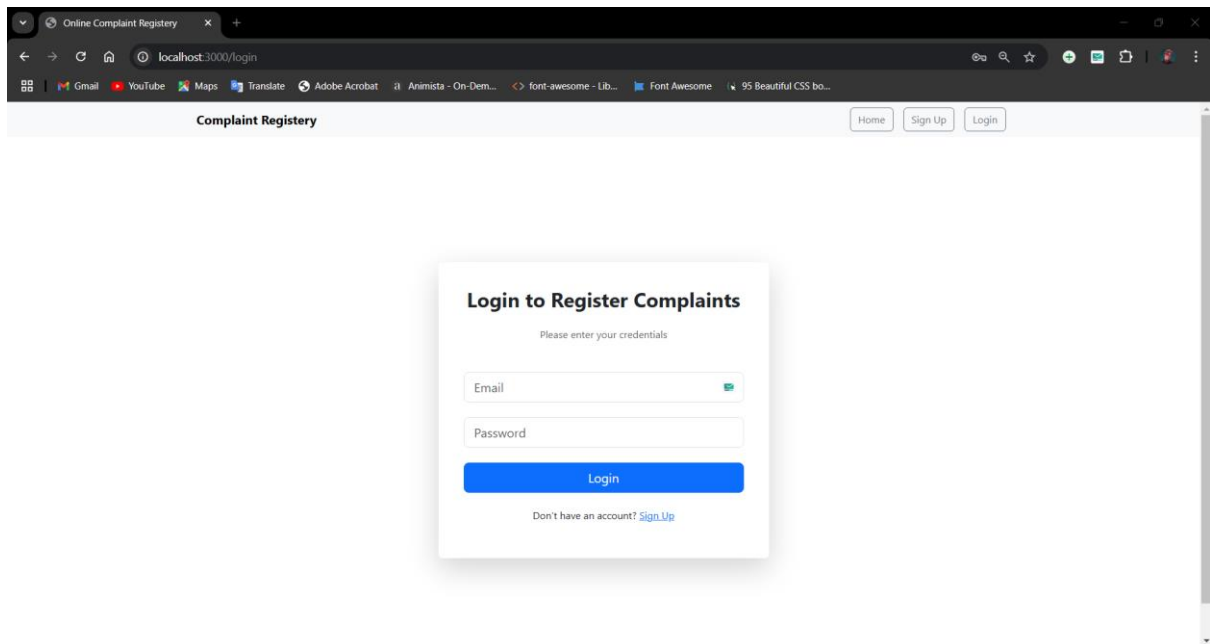
Admin Dashboard Dashboard User Agent

Hi, admin   Log out

Users Complaints

asdfg
Address: sdkjcbsdjcn
City: fouiuwhsef
State: dcihweuif
Pincode: 604403
Comment: yjtdkugfkg
Status: ygyug
Assign Agent ▾

user
Address: 23,sdkfuh
City: chennai
State: tn
Pincode: 604403
Comment: issue
Status: pending
Assign Agent ▾

qwer
Address: fytfd
City: uykfyu
State: jhgf
Pincode: 61421454
Comment: ktfkfgh
Status: pending
Assign Agent ▾

Agents

asdfg
Email: asdfg@gmail.xom

Hi, asdfg   Complaint Register Status   Log Out

Register Your Complaint

Name

Address

City

State

Pincode

Status
Type 'Pending'

Description

Submit Complaint

Hi, asdfg | Complaint Register | Status | Log Out

user
Address: 23,sdkfuh
City: chennai
State: tn
Pincode: 604403
Comment: issue
Status: pending
View Messages

qwer
Address: fytfd
City: uykfyu
State: jhgf
Pincode: 61421454
Comment: ktfkfgh
Status: pending
View Messages