

Full Stack Development using MERN

ResolveNow:Your Platform for Online Complaints

1.Introduction

Project Title: ResolveNow: Your Platform for Online Complaints

Team Members:

Team Leader : G Roselin

Team member : Divvela Jagadeesh

Team member : Dharmavarapu Jahnvi

Team member : Divi Vamsi

2.Project Overview

Purpose:The purpose of this project is to provide a centralized, user-friendly digital platform for efficiently registering, managing, and resolving complaints. It is designed to streamline communication between users, customer service agents, and administrators to ensure timely and transparent issue resolution.

Features:

User Registration and Authentication

Complaint Submission

Real-Time Complaint Tracking

Notifications System

Agent Interaction

Complaint Assignment and Routing

Admin Dashboard

3.Architecture

Frontend: Built with React.js, the frontend uses component-based architecture and integrates Axios for RESTful API communication. Bootstrap and Material UI enhance responsiveness and user experience. It includes dynamic routing and real-time updates for complaint tracking and messaging.

Backend: Developed using Node.js and Express.js, the backend handles API routing, authentication, and business logic. Middleware manages requests, sessions, and error handling. It enable real-time communication between users and agents.

Database: MongoDB stores user data, complaints, messages, and roles in a flexible document-based schema. Mongoose ORM manages schema definitions and CRUD operations. Relationships between users, complaints, and agents are maintained via ObjectId references.

4.Setup Instructions

Prerequisites:

Node.js and npm

MongoDB, Express

ReactJs, Vscode

Installation:

1. Clone the Repository

```
git clone https://github.com/awdhesh-student/complaint-registry.git
cd complaint-registry
```

2. Install Frontend Dependencies

```
cd frontend
```

```
npm install
```

3. Install Backend Dependencies

```
cd ../backend
```

```
npm install
```

4. Set up Environment Variables

Create a .env file inside backend/ and add:

PORT=5000

MONGODB_URL=mongodb://localhost:27017/

DB_NAME=ResolveNowDB

JWT_SECRET=your_secret_key

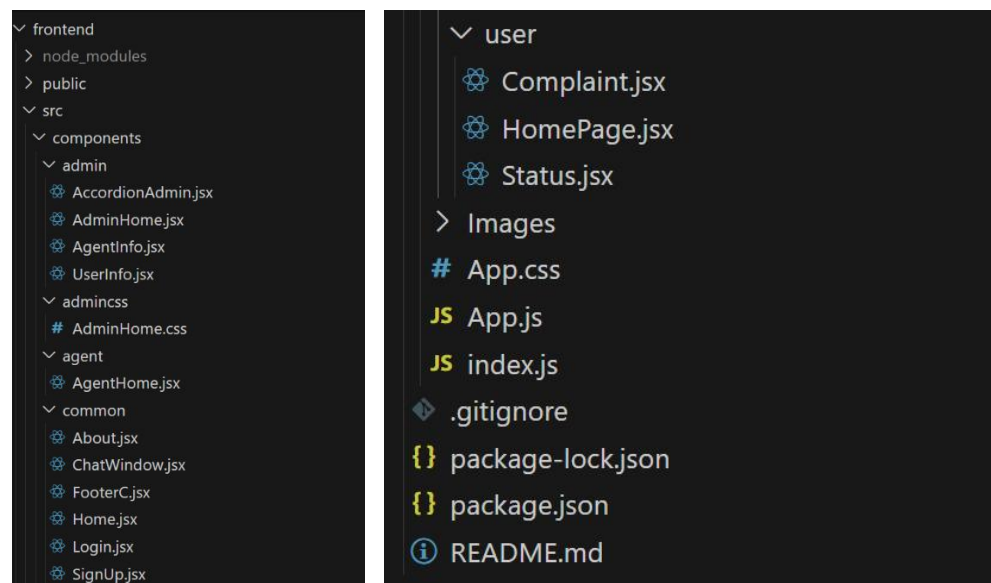
5. Start Servers

```
npm start # For backend
```

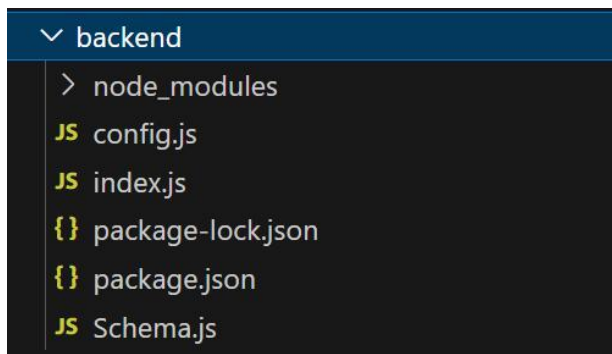
```
cd ../frontend && npm start # For frontend
```

5. Folder Structure

Frontend:



Backend:



6. Running the Application

Frontend:

```
cd frontend
```

```
npm start
```

Backend:

```
cd backend
```

```
npm start
```

Access the app at: <http://localhost:3000>

7. API Documentation

Endpoint	Method	Description
/api/auth/register	POST	Register a new user
/api/auth/login	POST	Login user and return token
/api/complaints	POST	Submit a new complaint
/api/complaints/:id	GET	Get specific complaint details
/api/admin/all	GET	Admin fetches all complaints

(Add more routes as needed, with example responses.)

8. Authentication

- JWT (JSON Web Tokens): Used for secure user session handling.
- Role-based Access: Admin, Agent, and User roles handled via token payload and middleware.
- Password Encryption: Implemented using bcryptjs.

9. User Interface

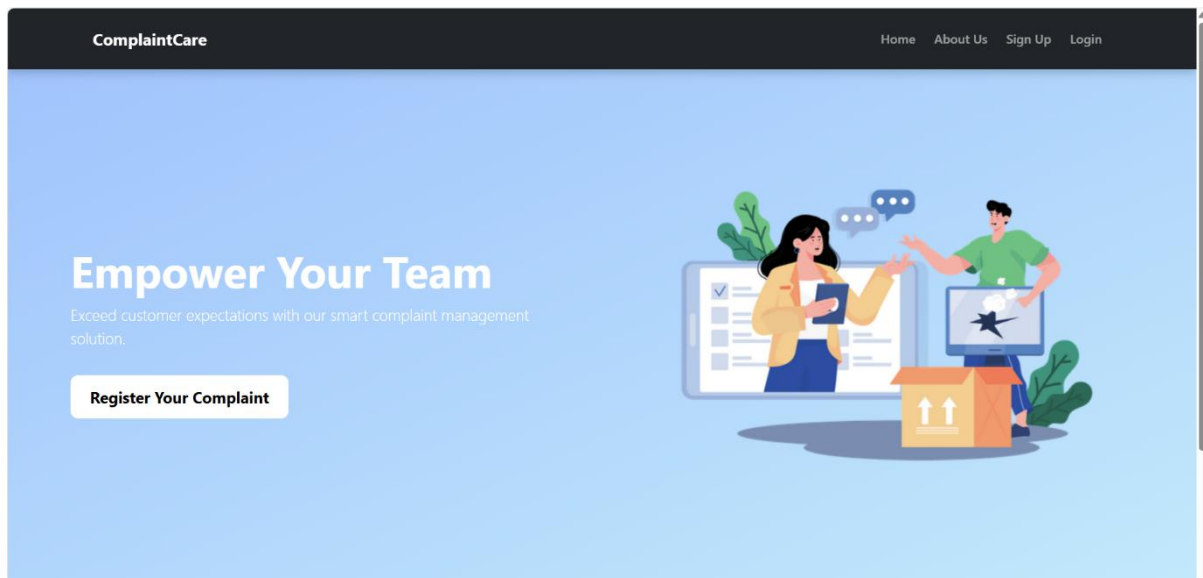
- **User Dashboard: Submit and track complaints**
- **Agent Panel: View assigned complaints and respond**
- **Admin Dashboard: Assign and monitor complaints**
(Include screenshots or demo links here)

10. Testing

- Manual and component-level testing using browser tools.
- Optionally integrate Jest or Mocha for unit and integration testing.
- Test cases cover complaint submission, login, and role-based access.

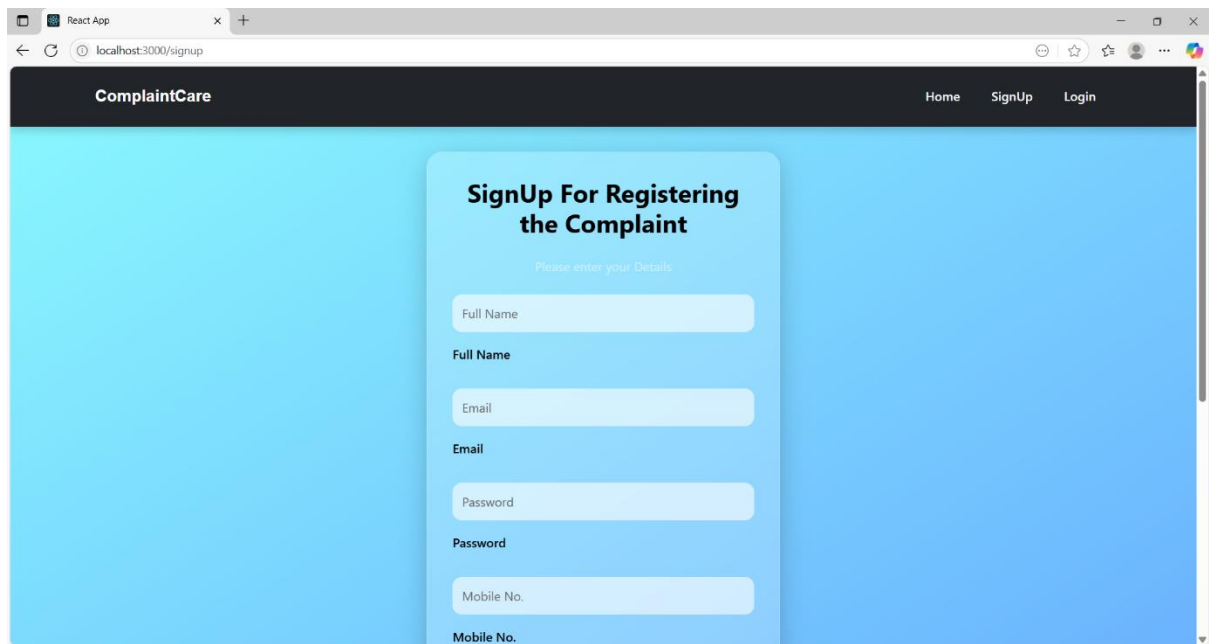
11. Screenshots or Demo

Home:



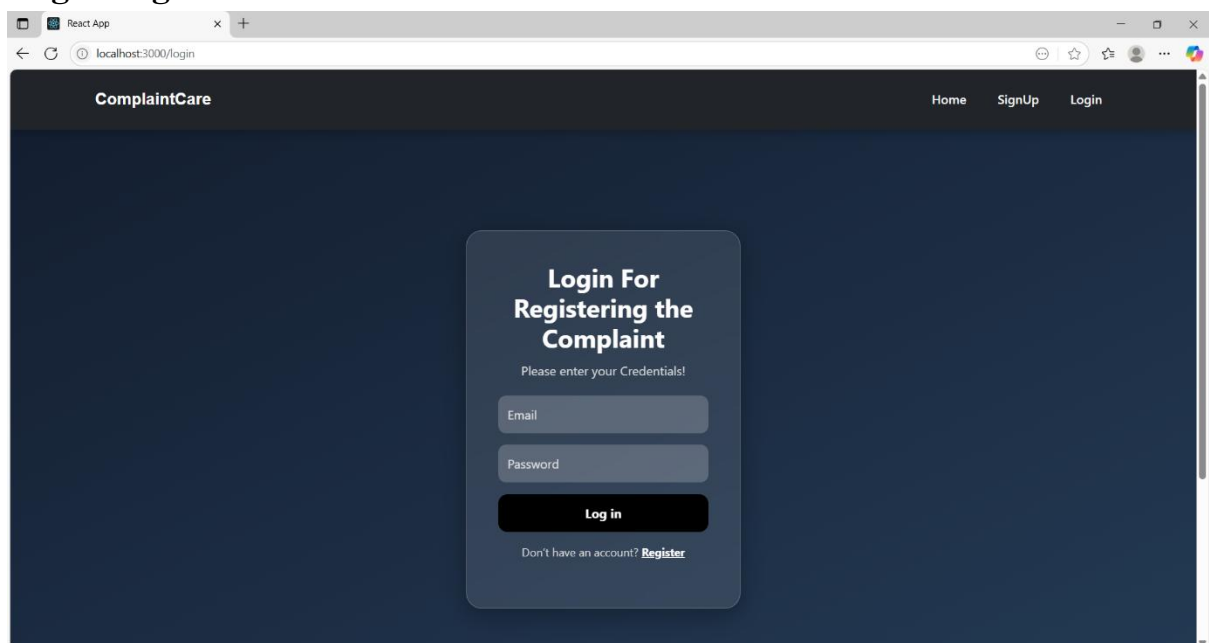
In this page user may signup, login, and register a complaint

SignUp Page:

A screenshot of a web browser showing the 'ComplaintCare' application. The page is titled 'SignUp For Registering the Complaint' and prompts the user to 'Please enter your Details'. The form includes input fields for 'Full Name', 'Email', 'Password', and 'Mobile No.', each with a corresponding label above it. The background is a light blue gradient. The browser's address bar shows 'localhost:3000/signup'.

In this page the user will register as admin or agent or ordinary. Admin may assign complaints to agent. Then agent interacts with the ordinary who raise a complaint

Login Page:

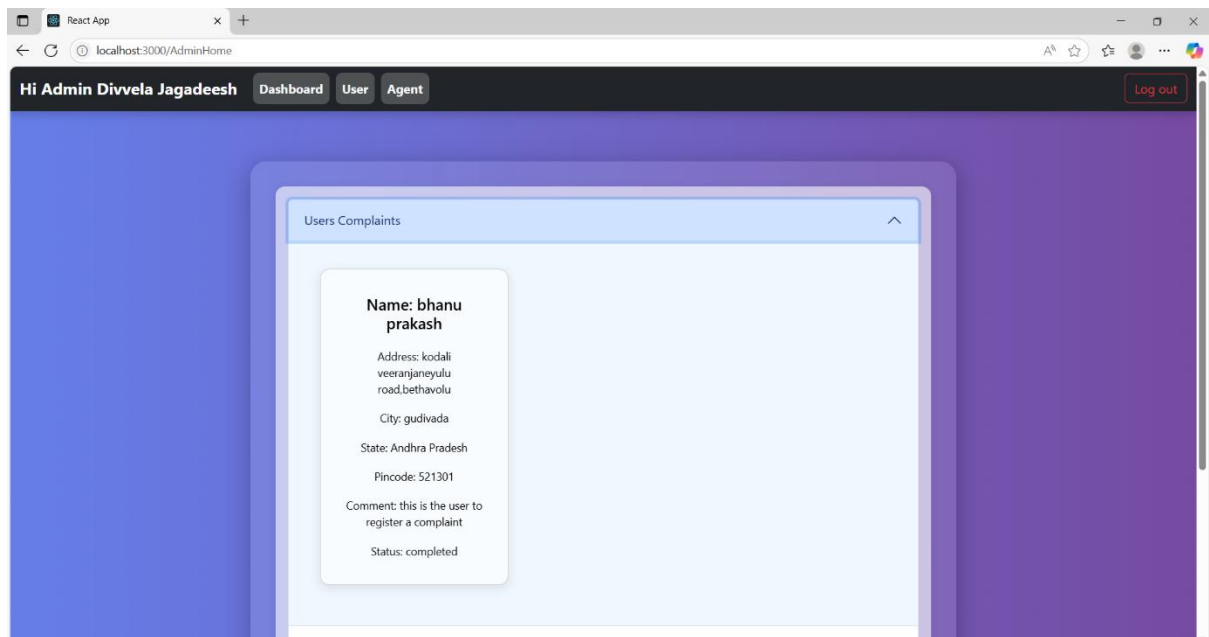
A screenshot of a web browser showing the 'ComplaintCare' application. The page is titled 'Login For Registering the Complaint' and prompts the user to 'Please enter your Credentials!'. The form includes input fields for 'Email' and 'Password', and a 'Log in' button. Below the button, there is a link that says 'Don't have an account? Register'. The background is a dark blue gradient. The browser's address bar shows 'localhost:3000/login'.

In this page every one will be login and do they work

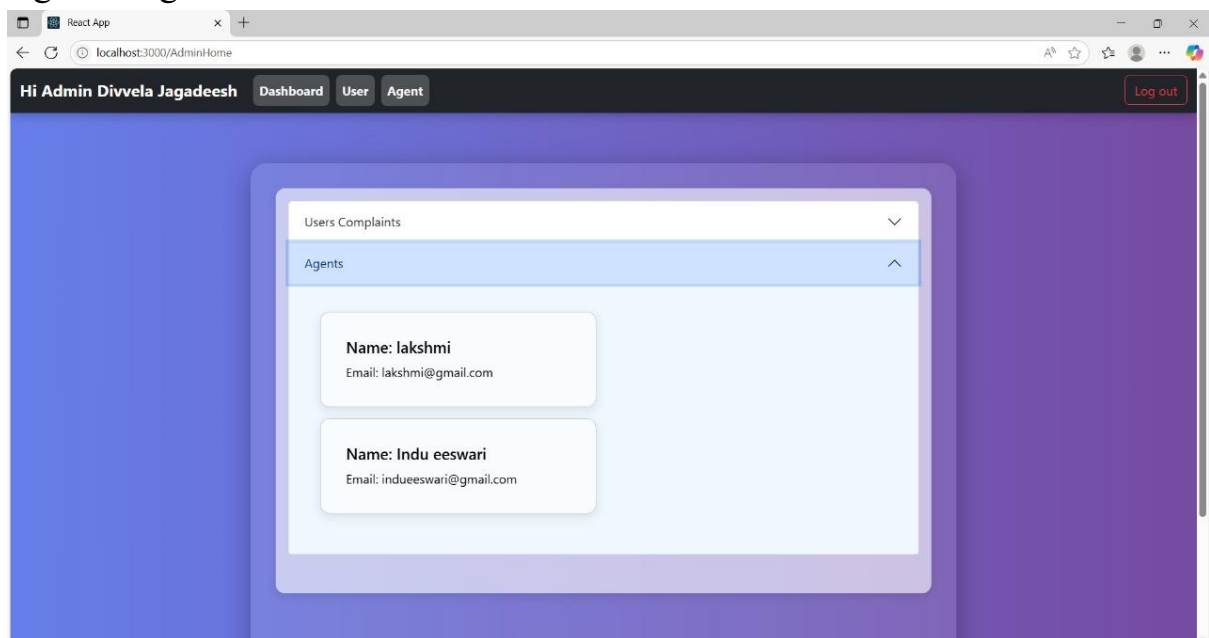
Admins Page:

In admin there is two services one is view agents and another one is user complaints

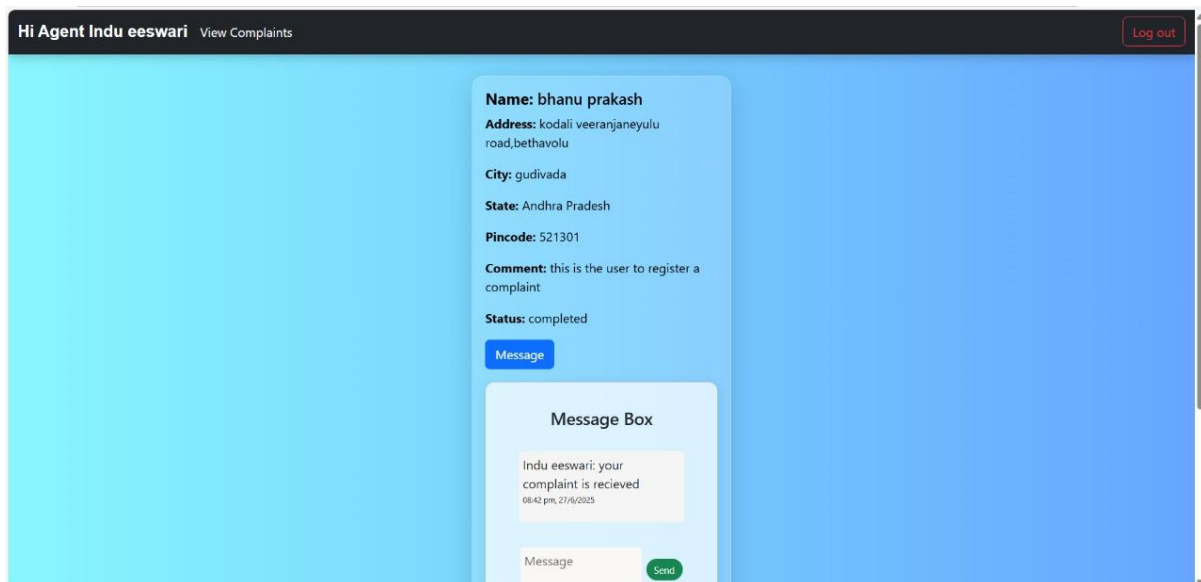
UserComplaints page:



Agents Page:

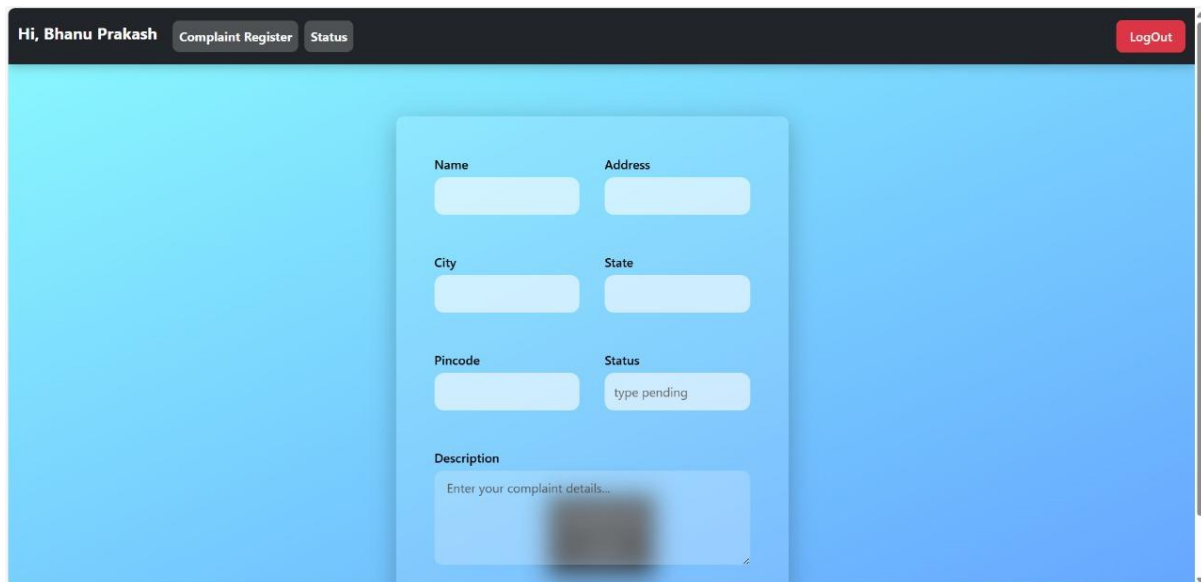


Agents page:



In the agent page the agent will interact with the ordinary to solve the complaint through the chat box

Ordinary users Page:



In this page user sent a complaint through the form and also click on left top status button they see the complaint status completed or pending

12. Known Issues

- WebRTC video chat may not work on all browsers.
- Email notifications may require SMTP setup for production.

13. Future Enhancements

- Integrate live video support with WebRTC
- Admin analytics dashboard with charts
- Push notifications via Firebase
- Mobile app version using React Native

