Full Stack Development with MERN

Project Documentation

Online Complaint Registration and Management Platform

DHAANISH AHMED COLLEGE OF ENGINEERING

Department of Computer Science And Engineering

Team Members:

1. H.Mohammed salman	UI/UX Developer
2. Cibi Raj	Frontend Developer
3. Sheik Meera mohaideen	Backend Developer
4. Syed Waseemuddin	Database management

Submitted to Naan Mudhalavan Team

ANNA UNIVERTSITY.

UNDER THE GUIDANCE OF

PROF Dr.M.Sree Rajeshwari

1.INTRODUCTION

PROJECT TITLE : Online Complaint Registration and Management Platform

ComplaintConnect is a modern online complaint registration platform designed to address the increasing need for efficient grievance redressal systems in the digital age. With a focus on user convenience, secure communication, and scalability, ComplaintConnect offers businesses and organizations a reliable and innovative solution to engage with their customers and stakeholders, ensuring transparency and accountability globally.

2.PROJECT OVERVIEW

PURPOSE

ComplaintConnect is an online complaint registration platform designed to streamline grievance resolution by offering a seamless user experience combined with robust administrative tools. Built with the MERN stack, the platform caters to organizations of all sizes, enabling them to manage complaints efficiently, track resolutions, and interact with users in real-time.

The key objectives of **ComplaintConnect** are:
• To create a secure and scalable platform for complaint registration and management.
• To provide a dynamic and user-friendly interface for customers.
• To offer organizations tools for tracking and resolving complaints efficiently.
• To integrate modern technologies for enhanced transparency and accountability.

FEATURES

Dynamic Complaint Categories

- Displays complaint types with real-time updates and customizable filters.
- Supports multiple categories and subcategories for streamlined issue identification.

Complaint Tracking System

- Allows users to submit, update, and review the status of their complaints.
- Provides real-time notifications and updates on resolution progress.

Admin Dashboard

- Enables complaint management with analytics, reporting, and performance metrics.
- Offers tools for prioritizing, assigning, and tracking complaint resolutions.

Mobile-Friendly Design

- Optimized for seamless navigation and complaint submission across devices.

3. ARCHITECTURE

Frontend:

- The frontend leverages React.js with a component-based architecture to ensure reusability and maintainability.
- React Router is used for efficient navigation, and Redux manages the application's state.

Backend:

- Built with Node.js and Express.js, the backend provides a RESTful API for handling user requests, processing business logic, and managing data interactions.

Database:

- MongoDB serves as the database, storing user details, complaint records, and resolution history.
- Mongoose is utilized for schema design, data validation, and seamless database interactions.

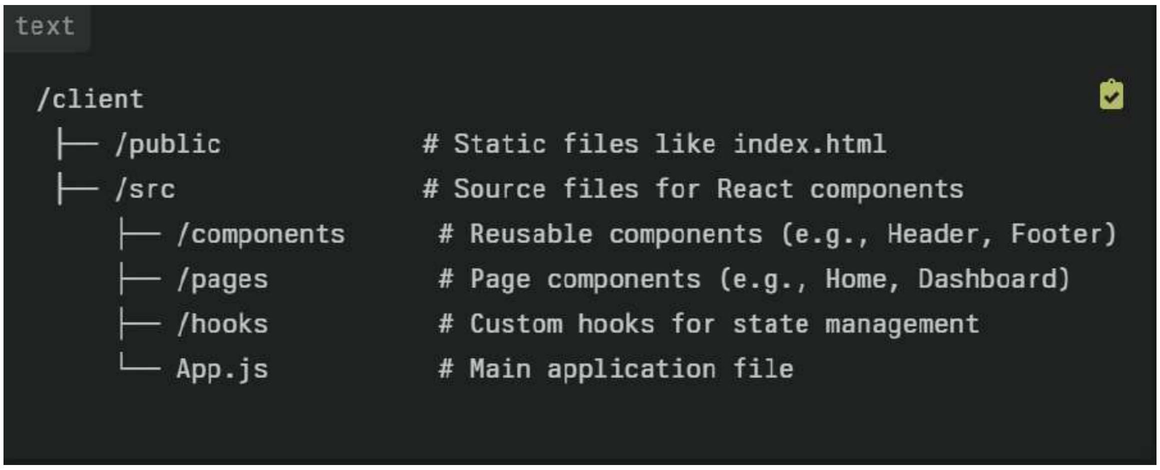
4. SETUP INSTRUCTIONS

- **Prerequisites:**
  - Node.js
  - MongoDB
- **Clone the repository:**
- Gitclone https://github.com/Mohammedsalman30291/Complaint-register-NM.git
- **Navigate to the client directory:**
  - cd frontend/client
- **Install frontend dependencies:**
  - npm install
- **Set up environment variables (create a .env file):**
  - MONGODB_URI=your_mongodb_uri
  - JWT_SECRET=your_jwt_secret

- **Navigate to the server directory:**
    - cd ../backend
- **Install backend dependencies:**
    - npm install

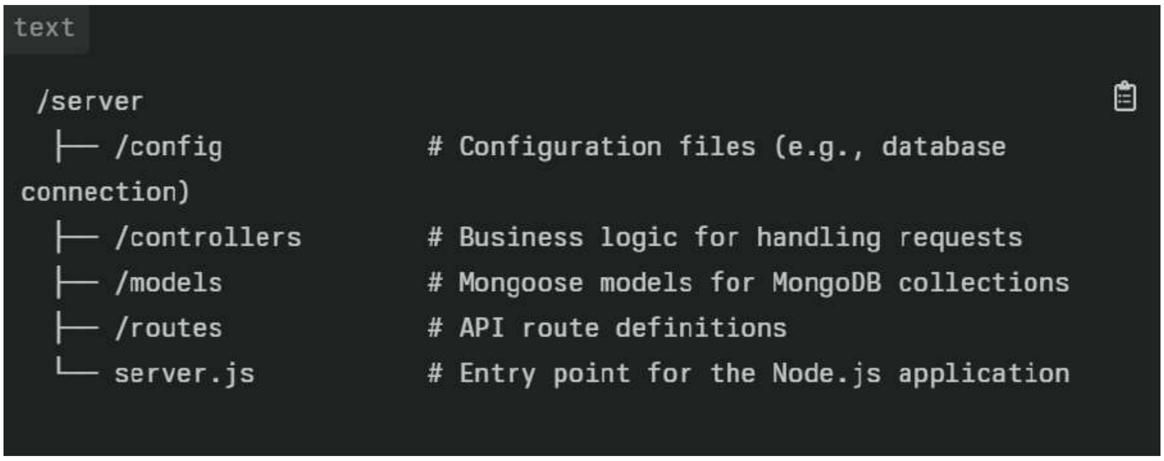
5. FOLDER STRUCTURE

- **Client Directory Structure**:



```text
/client
  ├── /public              # Static files like index.html
  ├── /src                 # Source files for React components
      ├── /components       # Reusable components (e.g., Header, Footer)
      ├── /pages            # Page components (e.g., Home, Dashboard)
      ├── /hooks            # Custom hooks for state management
      └── App.js            # Main application file
```

- **Server Directory Structure:**



```text
/server
  ├── /config              # Configuration files (e.g., database
connection)
  ├── /controllers         # Business logic for handling requests
  ├── /models              # Mongoose models for MongoDB collections
  ├── /routes              # API route definitions
  └── server.js            # Entry point for the Node.js application
```

6. RUNNING THE APPLICATION

- Provide commands to start the frontend and backend servers locally.
    - **Frontend:** **npm start** in the client directory.

    - **Backend:** **npm start** in the server directory.

7. API DOCUMENTATION

- **GET /api/tasks**: Retrieve all tasks for the authenticated user.
- **POST /api/users/register**: Register a new user.
- **POST /api/users/login**: Authenticate a user.
- **POST /api/tasks**: Create a new task.
- **PUT /api/tasks/:id**: Update an existing task.
- **DELETE /api/tasks/:id**: Delete a task.

8. AUTHENTICATION

- Authentication is handled using JWT (JSON Web Tokens) for secure user login and session management, where tokens are stored in HTTP-only cookies or local storage. Authorization is managed by verifying user roles and permissions based on the JWT claims to control access to protected routes and resources.

9. USER INTERFACE

Responsive Design:

- Demonstrates how the platform adapts seamlessly across devices, including desktop, tablet, and mobile views.

Intuitive Navigation:

- Highlights the user-friendly navigation bar and menu structure for quick access to key sections.

Complaint Categories Display:

- Showcases a clean and organized interface displaying various complaint categories with real-time updates.

Filtering and Sorting Options:

- Demonstrates customizable filters and sorting capabilities to help users refine complaints by type, date, or priority.

Complaint Detail Page:

- Displays detailed information about a specific complaint, including status, assigned personnel, and resolution timeline.

Complaint Tracking Interface:

- Showcases the user dashboard for adding, updating, and monitoring complaint progress in real-time.

Submission Process:

- Walkthrough of the complaint submission process, including form validation, confirmation messages, and reference ID generation.

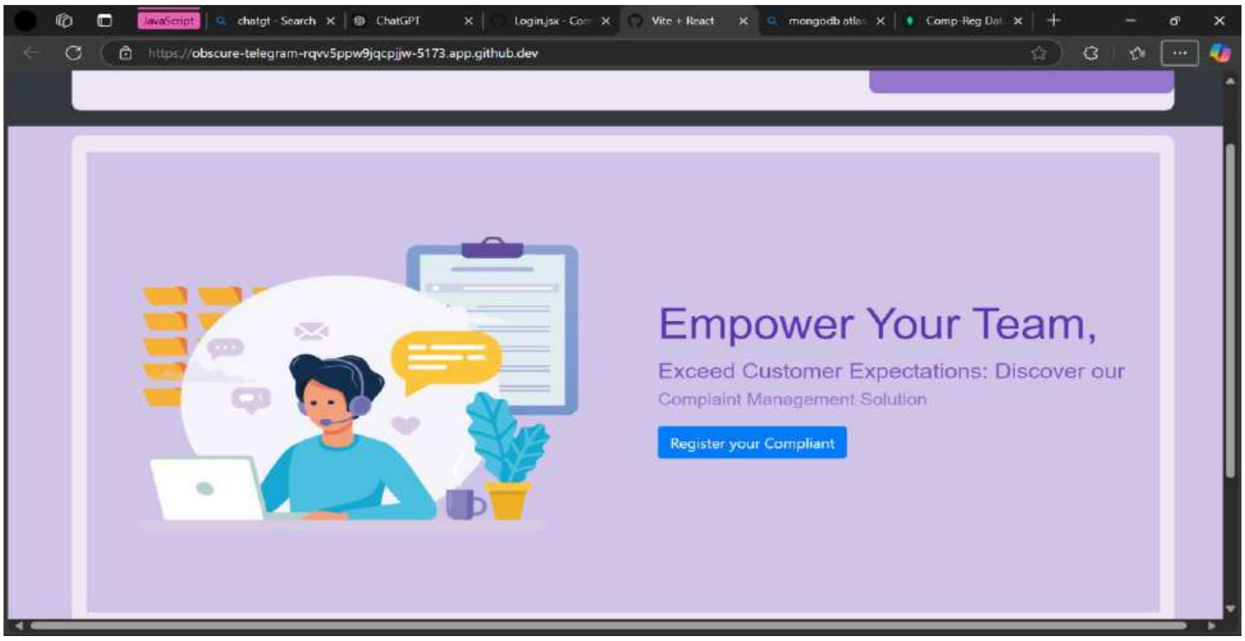
10. TESTING:

- **Exploratory Testing**
  - Conduct unscripted testing sessions to discover unexpected issues and assess user experience.
- **User Acceptance Testing (UAT)**
  - Engage end-users to validate that the application meets their expectations and requirements before launch**.**
- **Cross-Browser Compatibility Testing**
  - Verify that the site functions correctly across different web browsers (e.g., Chrome, Firefox, Safari) to ensure a consistent user experience**.**
- **Device and Responsive Testing**
  - Test the application on various devices (desktops, tablets, smartphones) to ensure it is fully responsive and functional**.**

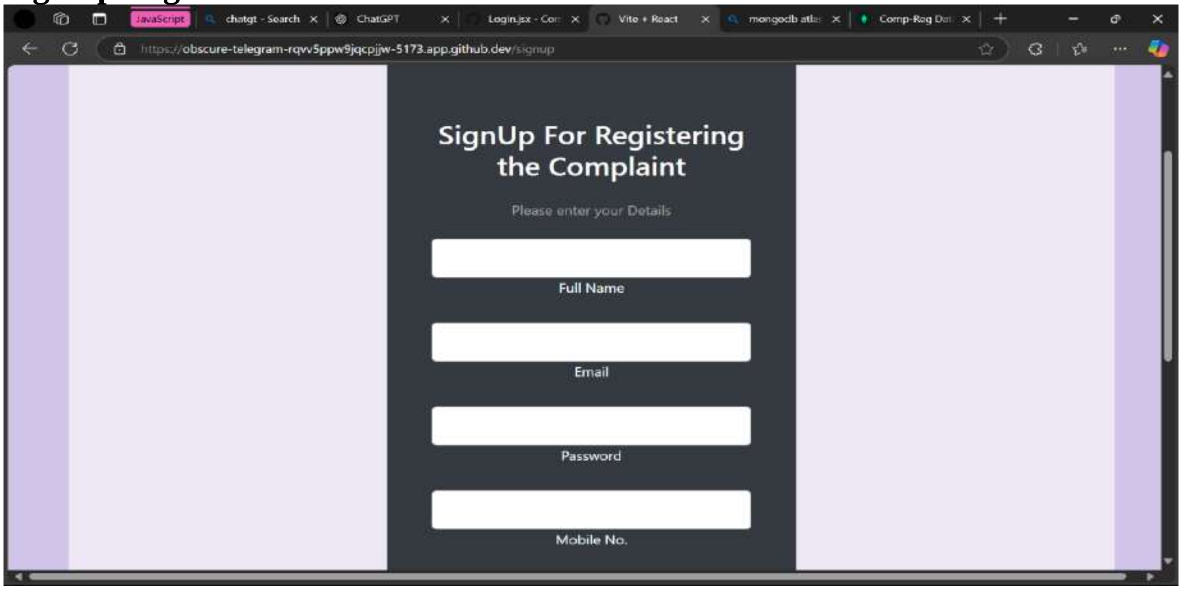
- **Performance Testing**
  - Assess the application's performance under various conditions, including load times and responsiveness during peak usage scenarios.

11. SCREENSHOTS OR DEMO

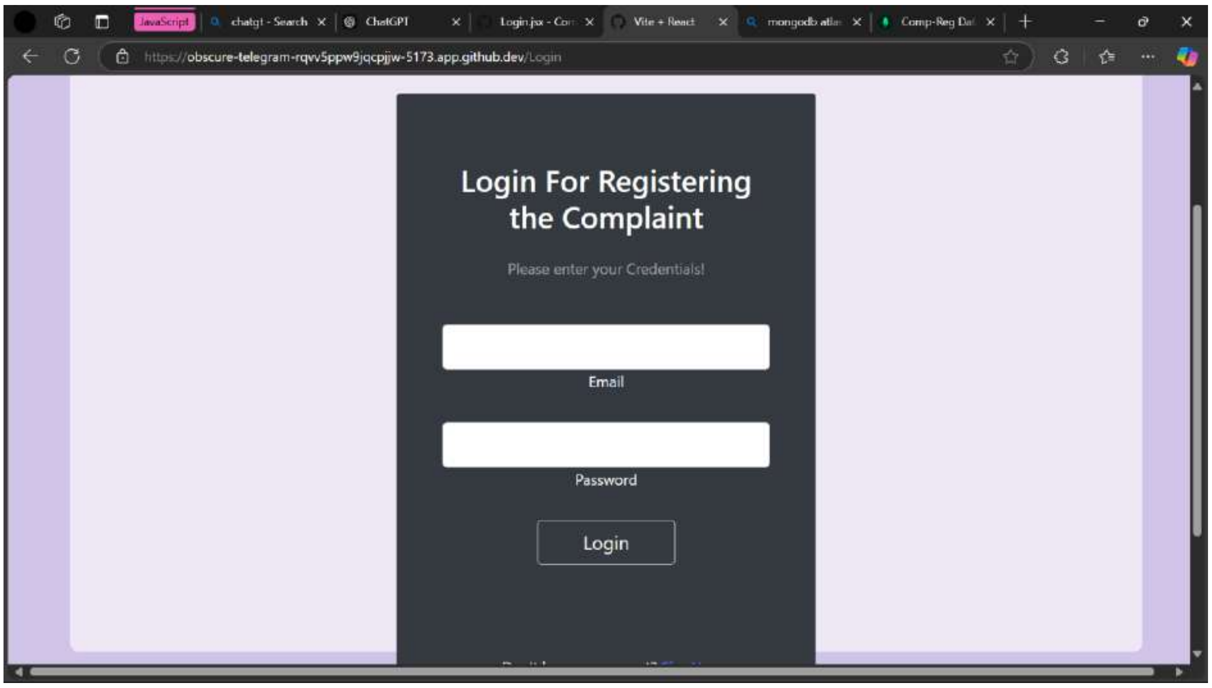
Home Page:


https://obscure-telegram-rqvv5ppw9jqcpjjw-5173.app.github.dev
Empower Your Team,
Exceed Customer Expectations: Discover our Complaint Management Solution
Register your Compliant

Sign up Page:

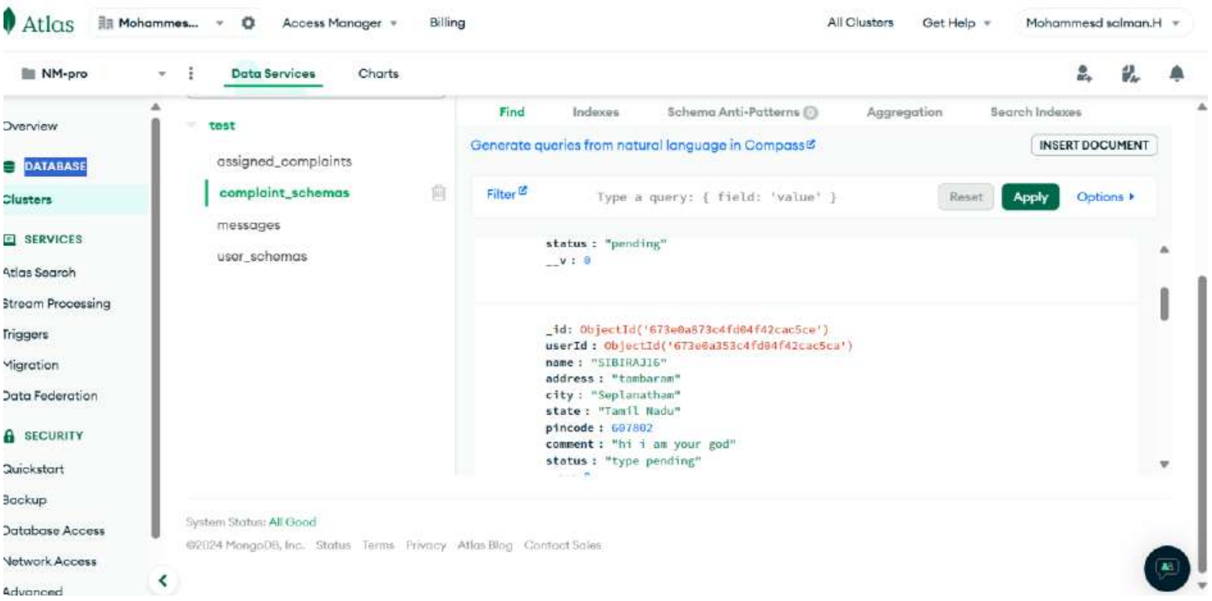

https://obscure-telegram-rqvv5ppw9jqcpjjw-5173.app.github.dev/signup
SignUp For Registering the Complaint
Please enter your Details
Full Name
Email
Password
Mobile No.

Sign in:



https://obscure-telegram-rqvv5ppw9jqcpjjw-5173.app.github.dev/Login
Login For Registering the Complaint
Please enter your Credentials!
Email
Password
Login

Database:



```
status : "pending"
__v : 0

_id: ObjectId('673e0a873c4fd04f42cac5ce')
userId : ObjectId('673e0a353c4fd04f42cac5ca')
name : "SIBIRAJ16"
address : "tambaram"
city : "Seplanatham"
state : "Tamil Nadu"
pincode : 607802
comment : "hi i am your god"
status : "type pending"
```

12. KNOWN ISSUES

- In the context of software development, particularly for applications built using the MERN stack, the inability to explore the application through automated testing can present significant challenge for us**.**

13. FUTURE ENHANCEMENTS

Develop voice-activated complaint registration capabilities through virtual assistants (e.g., Amazon Alexa, Google Assistant) to enable hands-free submission, tracking, and updates of complaints.

GITHUB REFERENCE LINK :
https://github.com/Mohammedsalman30291/Complaint-register-NM.git

DEMO REFERENCE VIDEO
https://drive.google.com/file/d/1l8VtYxX0dyJkVLNDJnFkOecE1W5Rz4Nf/view?usp=drivesdk