

Full Stack Development with MERN

Project Documentation format

Title:online complaint registration and management system

Team Members:

M.Keerthana

S.Thenmozhi

R.Harishma

R.Nithya

S.Hemalatha

Project description:

An online complaint registration and management system is a software application or platform that allows individuals or organizations to submit and track complaints or issues they have encountered. It can help optimize the complaint handling process and empower organizations to develop a safety management system to efficiently resolve customer complaints, while staying in line with industry guidelines and regulatory compliance obligations. It provides a centralized platform for managing complaints, streamlining the complaint resolution process, and improving customer satisfaction.

key features:

- **User registration:** Users can create accounts to submit complaints and track their progress.

- **Complaint submission:** Users can enter details of their complaints, including relevant information such name, description of the issue, address etc.
- **Tracking and notifications:** Users can track the progress of their complaints, view updates, and receive notifications via email or SMS when there are any changes or resolutions.
- User can interact with the agent who has assigned the complaint.
- Assigning and routing complaints: The system assigns complaints to the appropriate department or personnel responsible for handling them. It may use intelligent routing algorithms to ensure efficient allocation of resources.
- Security and confidentiality: The system ensures the security and confidentiality of user data and complaint information through measures such as user authentication, data encryption, access controls, and compliance with relevant data protection regulations.

DESCRIPTION:

The Online Complaint Registration and Management System is a user-friendly software solution designed to streamline the process of submitting, tracking, and resolving complaints or issues encountered by individuals or organizations. It provides a centralized platform for efficient complaint management, allowing users to securely register complaints, track their progress in real-time, and interact with assigned agents for issue resolution. With features such as automatic notifications, intelligent complaint routing, and robust security measures, this system ensures timely and effective handling of complaints while prioritizing user Details.

SCENARIO:

Scenario: John, a customer, recently encountered a problem with a product he purchased online. He notices a defect in the item and decides to file a complaint using the Online Complaint Registration and Management System.

- **User Registration and Login:**

- John visits the complaint management system's website and clicks on the "Sign Up" button to create a new account.
- He fills out the registration form, providing his full name, email address, and a secure password.
- After submitting the form, John receives a verification email and confirms his account.
- He then logs into the system using his email and password.

- **Complaint Submission:**

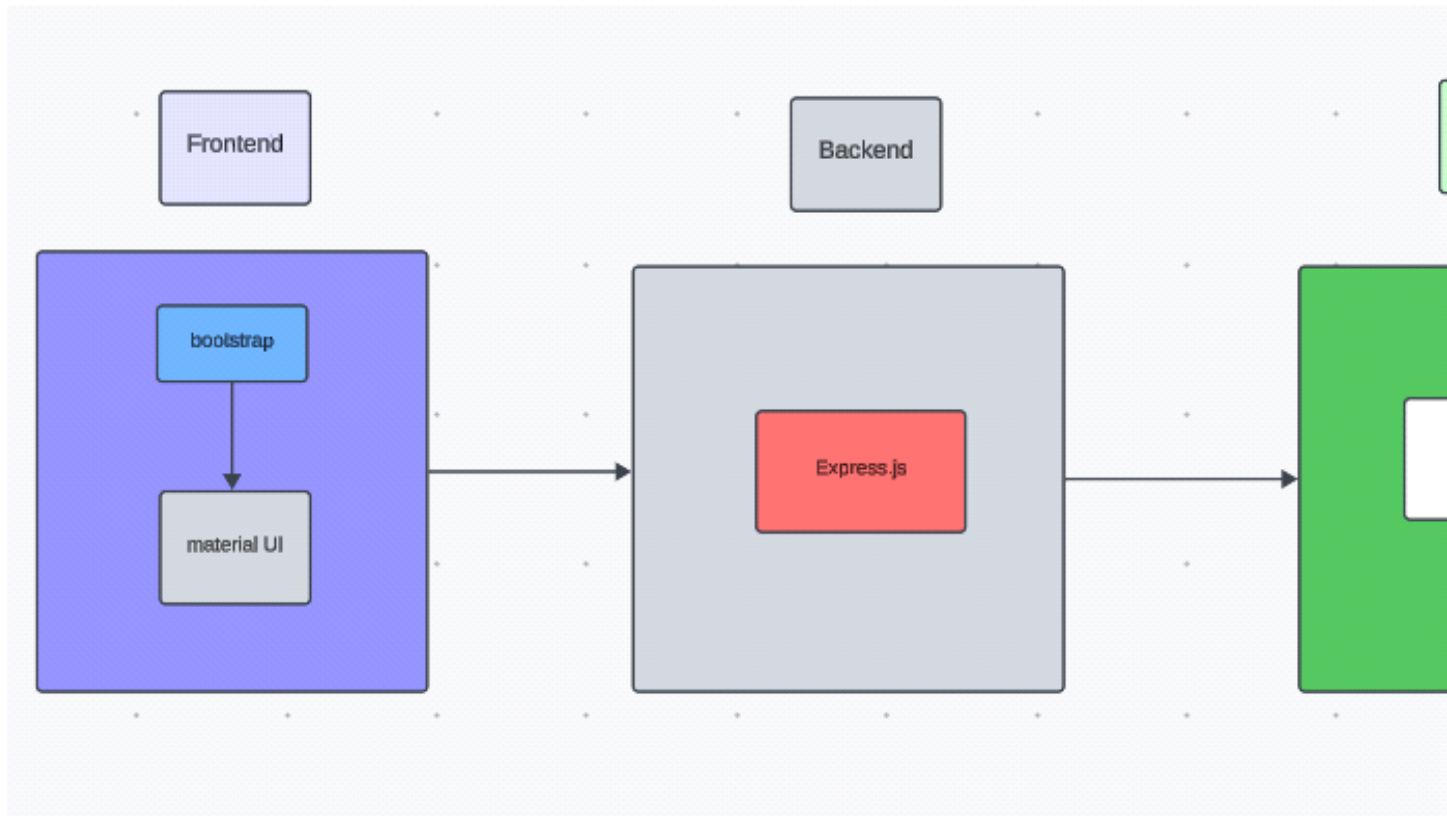
- Upon logging in, John is redirected to the dashboard where he sees options to register a new complaint.
- He clicks on the "Submit Complaint" button and fills out the complaint form.
- John describes the issue in detail, attaches relevant documents or images showcasing the defect, and provides additional information such as his contact details and the product's purchase date.
- After reviewing the information, John submits the complaint.

- **Tracking and Notifications:**

- After submitting the complaint, John receives a confirmation message indicating that his complaint has been successfully registered.
- He navigates to the "My Complaints" section of the dashboard, where he can track the status of his complaint in real-time.
- John receives email notifications whenever there is an update on his complaint, such as it being assigned to an agent or its resolution status.
- **Interaction with Agent:**
 - A customer service agent, Sarah, is assigned to handle John's complaint.
 - Sarah reviews the details provided by John and contacts him through the system's built-in messaging feature.
 - John receives a notification about Sarah's message and accesses the chat window to communicate with her.
 - They discuss the issue further, and Sarah assures John that the company will investigate and resolve the problem promptly.
- **Resolution and Feedback:**
 - After investigating the complaint, the company identifies the defect in the product and offers John a replacement or refund.
 - John receives a notification informing him of the resolution, along with instructions on how to proceed.

- He provides feedback on his experience with the complaint handling process, expressing his satisfaction with the prompt resolution and courteous service provided by Sarah.
- **Admin Management:**
 - Meanwhile, the system administrator monitors all complaints registered on the platform.
 - The admin assigns complaints to agents based on their workload and expertise.
 - They oversee the overall operation of the complaint management system, ensuring compliance with platform policies and regulations.

TECHNICAL ARCHITECTURE:



The technical architecture of our online complaint registration and management app follows a client-server model, where the frontend serves as the client and the backend acts as the server. The frontend encompasses not only the user interface and presentation but also incorporates the axios library to connect with backend easily by using RESTful Apis.

The frontend utilizes the bootstrap and material UI library to establish real-time and better UI experience for any user whether it is agent, admin or ordinary user working on it.

On the backend side, we employ Express.js frameworks to handle the server-side logic and communication.

For data storage and retrieval, our backend relies on MongoDB. MongoDB allows for efficient and scalable storage of user data, including user profiles, for complaints registration, etc. It ensures reliable and quick access to the necessary information during registration of user or any complaints.

Together, the frontend and backend components, along with socket.io, Express.js, WebRTC API, and MongoDB, form a comprehensive technical architecture for our video conference app. This architecture enables real-time communication, efficient data exchange, and seamless integration, ensuring a smooth and immersive video conferencing experience for all users.

Backend Development:

- **Implement Data Models:**

- Define Mongoose schemas for the different data entities like Bank, users, transactions, deposits and loans.
- Create corresponding Mongoose models to interact with the MongoDB database.
- Implement CRUD operations (Create, Read, Update, Delete) for each model to perform database operations.

- **Frontend Development:**

Setup React Application:

Bringing Customer Care Registry to life involves a three-step development process. First, a solid foundation is built using React.js. This includes creating the initial application structure, installing necessary libraries, and organizing the

project files for efficient development. Next, the user interface (UI) comes to life. To start the development process for the frontend, follow the below steps.

- Install required libraries.
- Create the structure directories

Design UI components:

Reusable components will be created for all the interactive elements you'll see on screen, from stock listings and charts to buttons and user profiles. Next, we'll implement a layout and styling scheme to define the overall look and feel of the application. This ensures a visually-appealing and intuitive interface. Finally, a navigation system will be integrated, allowing you to effortlessly explore different sections of Customer Care Registry, like making specific complaints or managing your Product complaints.

Implement frontend logic:

In the final leg of the frontend development, we'll bridge the gap between the visual interface and the underlying data. It involves the below stages.

- Integration with API endpoints.
- Implement data binding.

github link:

implementation link:

https://github.com/Thenmozhi05236/CM_REG/blob/main/Project%20Implementation.docx

Backend link:

https://github.com/Thenmozhi05236/CM_REG/blob/main/backend.zip

Frontend link:

https://github.com/Thenmozhi05236/CM_REG/blob/main/frontend.zip

Demo video link:-

GITHUB:

[https://github.com/Thenmozhi05236/CM_REG/blob/main/demo%20video\(CM_REG\).mp4](https://github.com/Thenmozhi05236/CM_REG/blob/main/demo%20video(CM_REG).mp4)

Drive: <https://drive.google.com/file/d/16ApoxmnLfp4naHEzoASAJKu0-rzNUtpe/view?usp=sharing>