# ONLINE COMPLAINT REGISTRATION AND MANAGEMENT SYSTEM

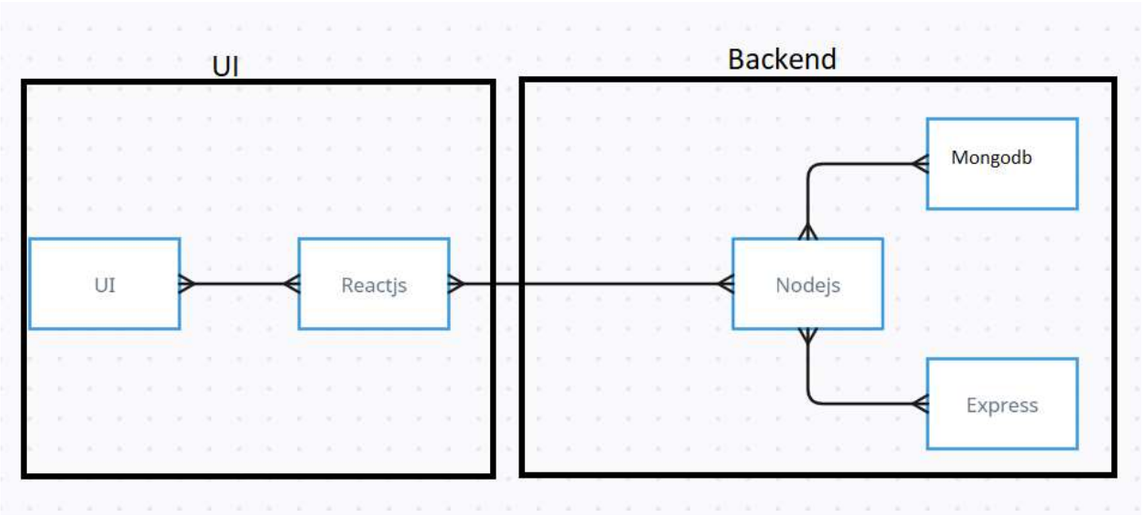
## INTRODUCTION :-

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.

It consists of some key features which include:

1. User registration: Users can create accounts to submit complaints and track their progress.
2. Complaint submission: Users can enter details of their complaints, including relevant information such name, description of the issue, address etc.
3. 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.
4. User can interact with the agent who has assigned the complaint.
5. 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.
6. 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.

## 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.

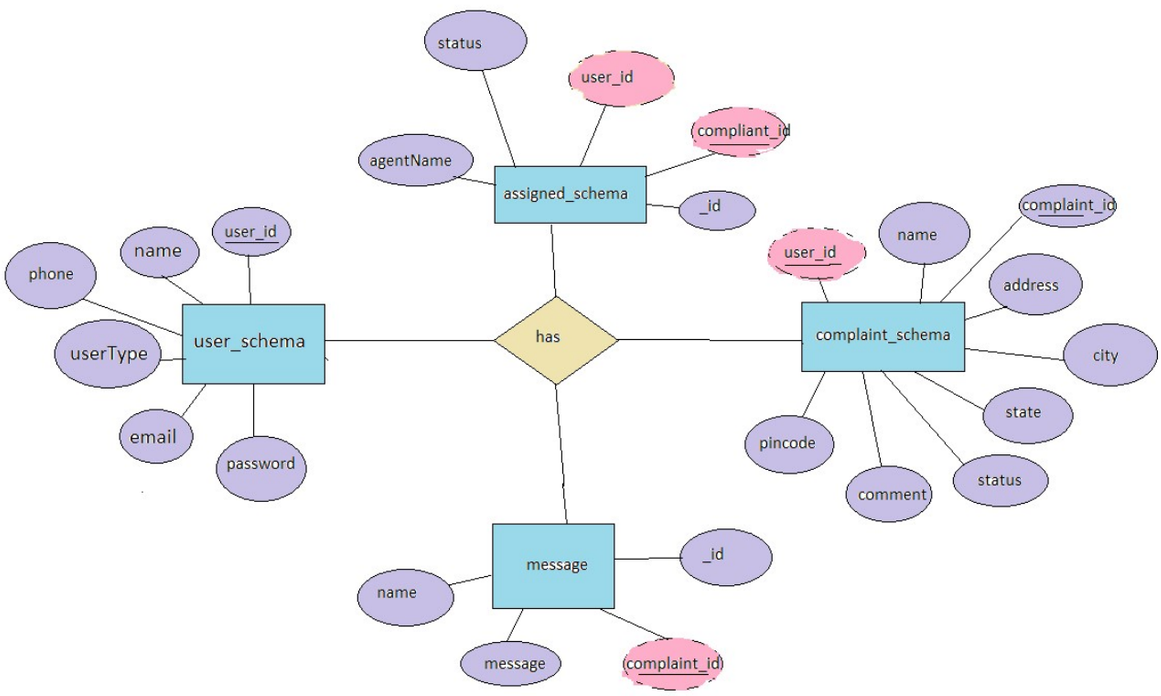
The frontend utilizes the bootstrap and material UI library to establish real-time and better UI experience for any user whether it is agent, admin or ordinary user working on it.

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.

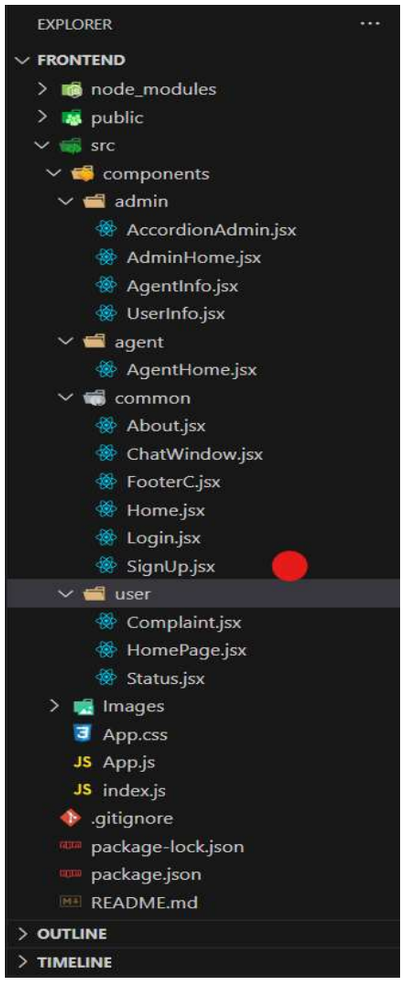
Together, the frontend and backend components, along with socket.io, Express.js, WebRTC API, and MongoDB, form a comprehensive technical architecture for our video conference app. This architecture enables real-time communication, efficient data exchange, and seamless integration, ensuring a smooth and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

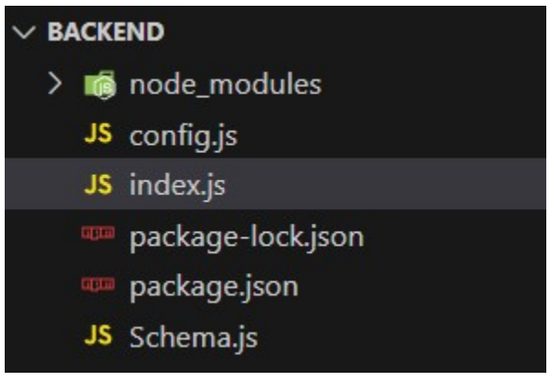
## ER DIAGRAM :-




This is the er diagram of the project which shows the relationship between user and agent
It shows how user which have required fields can raise a complaint by fillings required fields.
It illustrates how these entities relate to each other, helping us understand the underlying database structure and the flow of information within the app. He / She can also communicate with the agent with chat window which follows the message schema which uses userId and complaintId from other schemas.

## PROJECT STRUCTURE:




The first image is of frontend part which is showing all the files and folders that have been used in UI development

The second image is of Backend part which is showing all the files and folders that have been used in backend development

## PRE-REQUISITES : -

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, React.js:

✔ **Node.js and npm:**

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server-side. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.
Download: https://nodejs.org/en/download/
Installation instructions: https://nodejs.org/en/download/package-manager/

✔**Express.js:**

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture.
Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.
Installation: Open your command prompt or terminal and run the following command:

```
npm install express
```

✔**MongoDB:**

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

Set up a MongoDB database to store your application's data.
Download: https://www.mongodb.com/try/download/community
Installation instructions: https://docs.mongodb.com/manual/installation/

✔**React.js:**

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.
Install React.js, a JavaScript library for building user interfaces.
Follow the installation guide: https://reactjs.org/docs/create-a-new-react-app.html

✔ **HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

✔ **Database Connectivity**: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations. To Connect the Database with Node JS go through the below provided link:
https://www.section.io/engineering-education/nodejs- mongoosejs-mongodb/

✔ **Front-end Framework**: Utilize Reactjs to build the user-facing part of the application, including entering complaints, status of the complaints, and user interfaces for the admin dashboard.
For making better UI we have also used some libraries like material UI and boostrap.

✔**Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.
Git: Download and installation instructions can be found at: https://git-scm.com/downloads

✔ **Development Environment**: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

• Visual Studio Code: Download from https://code.visualstudio.com/download

To run the existing Video Conference App project downloaded from GitHub:
Follow below steps:
Clone the Repository:

- Open your terminal or command prompt.
- Navigate to the directory where you want to store the e-commerce app.
- Execute the following command to clone the repository:

**git clone**: https://github.com/awdhesh-student/complaint-registery.git

Install Dependencies:

• Navigate into the cloned repository directory:
cd complaint-registery
• Install the required dependencies by running the following commands:
cd frontend
npm install
cd ../backend
npm install
Start the Development Server:
• To start the development server, execute the following command:
npm start
• The online complaint registration and management app will be accessible at http://localhost:3000

You have successfully installed and set up the online complaint registration and management app on your local machine. You can now proceed with further customization, development, and testing as needed.

## Roles and Responsibilities:

The project has two types of users – Agents and Customer and third is Admin which takes care to all the user whether it is Agent or simple user. The roles and responsibilities of these two types of users can be inferred from the API endpoints defined in the code. Here is a summary:

### Customer/Ordinary:

1. Create an account and log in to the system using their email and password.
2. Browse and fill the form of your complaint or any issues for the agent to solve.
3. After filling your complaint, he/she can view the status of complaint in the status section.
4. He/She can connect to the agent directly by sending message and talk more about the complaints by using chat window.
5. Manage their profile information, including personal details and shipping addresses.

### Agent:

1. Create an account and log in to the system using their email and password.
2. Manage all the complaints assigned by the Admin.
3. He/She can connect directly to the user of the complaint by sending the message through chat window.
4. If complaints are resolved, he can change the status by clicking the button a
5. Interact with customers by responding to inquiries, resolving issues, and addressing feedback.

### Admin:

1. Manage and monitor the overall operation of the complaint registering platform.
2. Monitor and moderate all the complaints that are coming from the user
3. Easily assigned the complaints to the desired agent.
4. Manage user as well as agents accounts.
5. Implement and enforce platform policies, terms of service, and privacy regulations.
6. Continuously improve the platform's functionality, user experience, and security measures.

**Project Flow:**

Before starting to work on this project, let's see the demo.
**Project demo:**
**https://drive.google.com/file/d/1YwXaHRBZJL_V7dcEK8SOmtPWZasAxccm/view?usp=drive_link**
Use the code in: https://github.com/awdhesh-student/complaint-registery.git
or follow the videos below for better understanding.

**Milestone 1: Project setup and configuration.**

✔**Folder setup:**

1. Create frontend and
2. Backend folders

✔**Installation of required tools:**

1. Open the frontend folder to install necessary tools

For frontend, we use:

- React JS
- Bootstrap
- Material UI
- Axios

2. Open the backend folder to install necessary tools

For backend, we use:

- cors
- express
- mongoose

**Milestone 2: Backend Development**

✔**Setup express server**

1. Create index.js file in the server (backend folder).
2. define port number to access it.
3. Configure the server by adding cors, body-parser.

✔**Configure MongoDB**

1. Import mongoose.
2. Add Database URL to the config.js file.
3. Connect the database to the server.
4. Create a ‘schema’ file in the server to store all the DB models.

✔ **Add authentication**

1. Create the “User Schema” model for the MongoDB.
2. Define registration & login activities.
3. Using Axios library, make request from the frontend and vice-versa.

**Milestone 3: Web Applications development**

✔**Add complaints and the status of the complaints**

1. Create the “complaint_schema” model for the MongoDB to register complaint
2. Defined well RESTful API for better fetching the records

✔ **Sending message to agent through chat window**

1. Create the “message” model for the MongoDB to sending more description about the complaint easily.
2. It can keep records by using various attributes present in rest of the collections.
3. The chat window can adjust himself with the latest message arrived at the chat window.

✔**CRUD operation are done by admin**

1. Admin has the right to perform deletion, updating or add user in the database.
2. He can assign the issues or complaints by himself to the right agents present in collection.