

Online Complaint Registration And Management System – using MERN



OBJECTIVE

The Online Complaint Registration and Management System offers a streamlined, efficient platform for users to easily submit, track, and resolve complaints. It enhances response times, ensures transparency, and improves accountability. By centralizing the complaint management process, it optimizes resource allocation, boosts customer satisfaction, and enables real-time updates. The system transforms traditional complaint handling into a structured, user-friendly experience, meeting the modern demand for quick, efficient, and transparent issue resolution.

INTRODUCTION



In today's digital world, customer satisfaction is crucial, and managing complaints effectively is key. Traditional methods can be slow and error-prone, leading to frustration. An Online Complaint Registration and Management System (OCRMS) offers a centralized, automated solution to handle complaints efficiently and improve customer experience

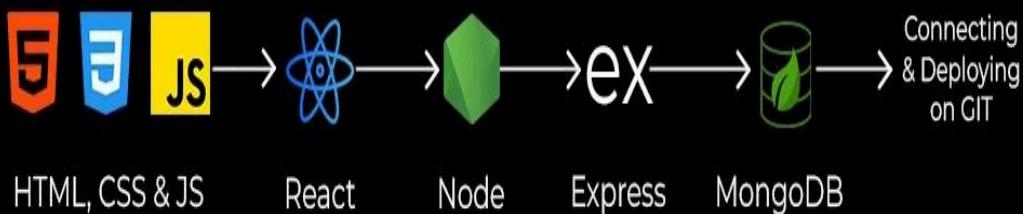
PROBLEM STATEMENT:

The problem with traditional complaint management systems is that they are often manual, inefficient, and lack transparency, leading to delays, user frustration, and unresolved complaints. In many organizations and institutions, complaints are typically filed through paper forms, phone calls, or email, which makes it difficult to track the status and progress of individual cases. This manual approach often results in prolonged response times, miscommunication, and a lack of accountability, as there is no centralized system to monitor complaint resolution effectively.

SOLUTION:

The aim of an online complaint registration and management system is to address these issues by providing a digital platform that allows users to easily submit complaints, track their progress, and receive timely updates

Roadmap to MERN



M - MongoDB (*NoSQL database*)

E - Express.js (*backend framework*)

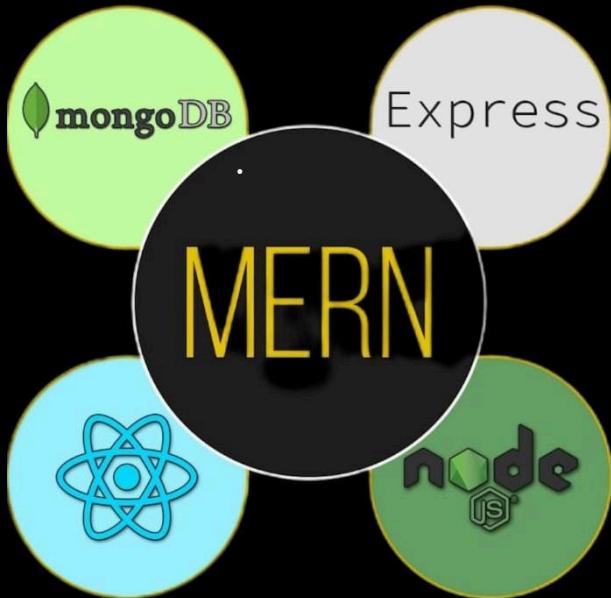
R - React.js (*frontend library*)

N - Node.js (*JavaScript runtime environment*)

The MERN stack allows developers to create scalable, efficient, and dynamic web applications using JavaScript across the entire stack.

TOOLS USED

Using the MERN stack—MongoDB, Express.js, React, and Node.js—creates a robust complaint management system. MongoDB stores complaint and user data, while Express.js and Node.js handle backend operations and API routes for smooth data flow. React provides a dynamic, user-friendly interface, enabling users to submit and track complaints seamlessly.



MONGO DB

MongoDB is a popular, open-source NoSQL database that stores data in a flexible, JSON-like format called *BSON* (Binary JSON). Unlike traditional relational databases, MongoDB doesn't require predefined schemas, allowing for dynamic and scalable data storage. It is highly scalable, offers high availability, and is often used for handling large volumes of unstructured or semi-structured data in real-time applications. MongoDB is widely used in web, mobile, and big data applications.

MongoDB



MongoDB is a flexible, scalable NoSQL database for dynamic data.

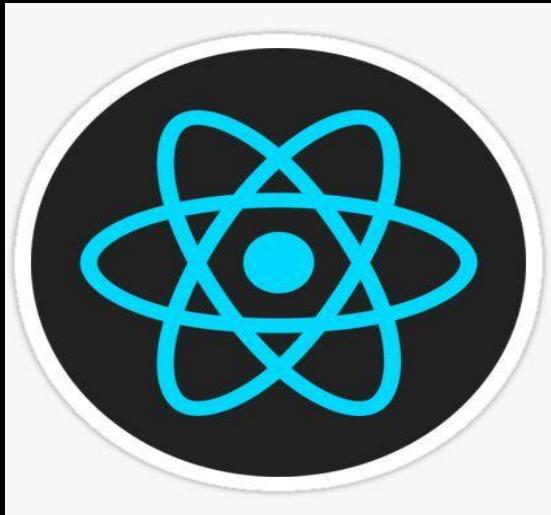
EXPRESS JS



Express is a lightweight Node.js framework for building web applications.

Express is a fast, minimal web application framework for Node.js, used to build server-side applications and APIs. It simplifies handling HTTP requests and responses, routing, and middleware, making it easier to develop robust, scalable applications. Express provides a flexible structure that allows developers to add various modules, making it ideal for building RESTful APIs and full-stack applications. It's a popular choice in the MERN (MongoDB, Express, React, Node) stack for web development.

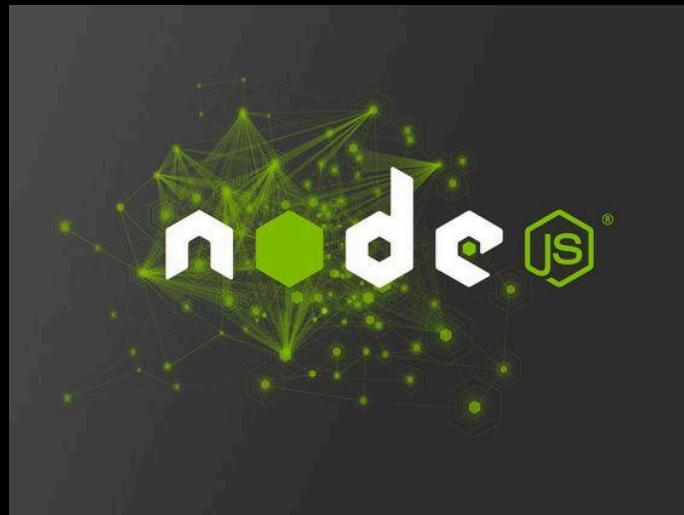
REACT JS



React is a popular, open-source JavaScript library developed by Facebook for building dynamic, interactive user interfaces, especially for single-page applications. Using a component-based architecture, React enables developers to create reusable UI components that update efficiently in response to data changes. Its virtual DOM feature optimizes rendering, making applications fast and responsive. React is widely used in modern web development, often paired with libraries and frameworks to build full-featured applications, and is a core technology in the MERN (MongoDB, Express, React, Node) stack.

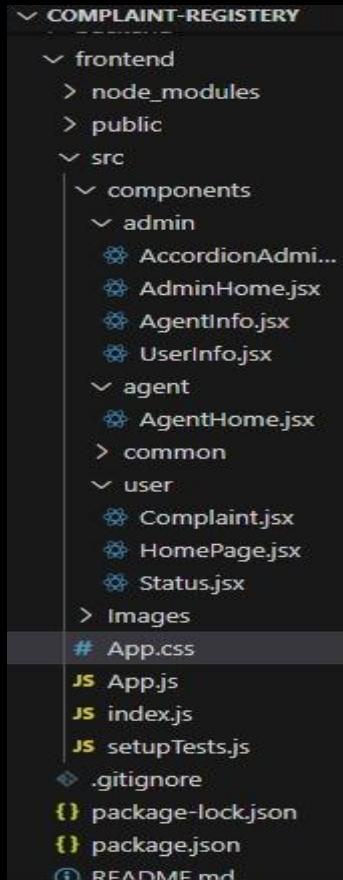
React is a powerful JavaScript library for creating fast, dynamic UIs.

NODE JS



Node.js is an open-source, server-side runtime environment that allows developers to run JavaScript code outside the browser, primarily for building scalable network applications. Built on Chrome's V8 JavaScript engine, Node.js is known for its event-driven, non-blocking I/O model, which makes it efficient and well-suited for real-time, data-intensive applications. It's commonly used for backend development, particularly in building APIs and web servers, and is a key component of the MERN (MongoDB, Express, React, Node) stack.

Node.js is a JavaScript runtime environment for building scalable, server-side applications.

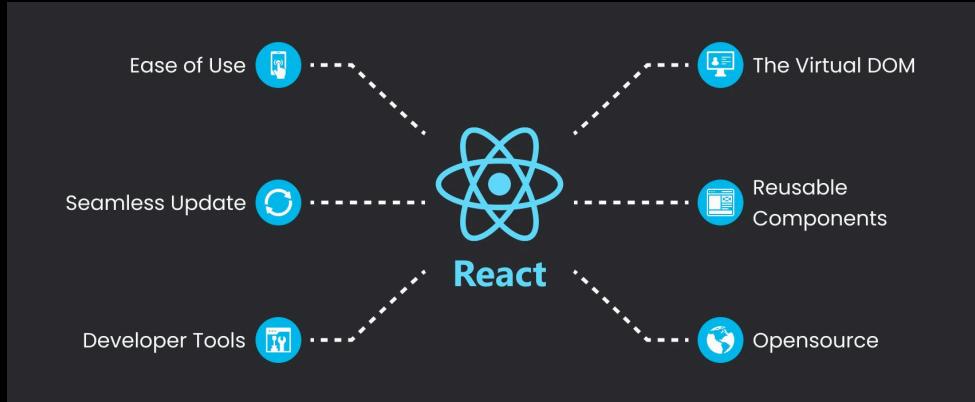


FRONT-END FUNCTIONALITY

1. User-Interface :

The login page allows users, administrators, and agents to securely access the system using credentials, with JWT for session management and live validation for a smooth user experience. The complaint registration page enables users to file complaints, capturing required details and supporting documents, with real-time validation. The user dashboard displays filed complaints, allowing users to track status and filter by various criteria. The admin and agent dashboards provide centralized complaint management, allowing for assignment, resolution, and real-time updates through tables, charts, and search functionalities.

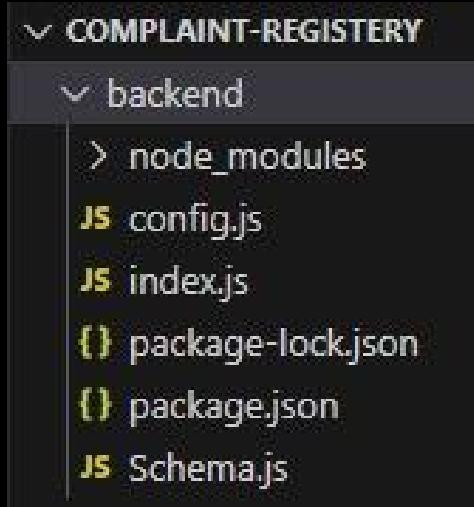
2.React:



The front-end functionality of an online complaint registration and management system plays a crucial role in providing users with an intuitive and responsive experience. Built using React, a widely popular JavaScript library for building user interfaces, the front end ensures that the application is fast, interactive, and easy to use. React's component-based architecture allows for modular development, making it easier to manage and update individual elements without affecting the entire system.

BACK-END FUNCTIONALITY

*The backend of the Online Complaint Registration and Management System, built with **Node.js** and **Express**, handles data processing, secure user authentication, and API routing.*



1. Express :

manages routes for complaint submission, status updates, and real-time notifications.

2. Node.js :

ensures a fast and scalable server. Admin and agent-specific routes enable efficient complaint management, with robust security features to protect user data and support seamless front-end communication.



DATABASE FUNCTIONALITY

MongoDB:

MongoDB is a NoSQL. In a complaint management system, the database records complaint details, tracks statuses, manages user information, logs staff-user communications, and supports real-time updates. It also generates analytics for trend analysis and reporting, and triggers notifications for status changes, enabling efficient, transparent, and accountable management of complaint resolutions.

DATABASE CONNECTIVITY

The screenshot shows the MongoDB Atlas Data Services Overview page for the project "sample-project". The top navigation bar includes "Atlas", "rithika's Org ...", "Access Manager", "Billing", "All Clusters", "Get Help", and "rithika". The left sidebar lists "Clusters", "SERVICES", "Atlas Search", "Stream Processing", "Triggers", "Migration", "Data Federation", "Programmatic Access", "SECURITY", "Quickstart", "Backup", and "Database Access". The main "Overview" section displays a large green circular progress bar labeled "RITHIKA'S ORG - 2024-09-08 > SAMPLE-PROJECT". Below it, the "Clusters" section shows "Cluster0" with "Data Size: 135.78 MB". It features three buttons: "Connect", "Edit configuration", and "Create cluster". To the right are links for "Browse collections" and "Migrate data". At the bottom is a "View monitoring" link and a "+ Add Tag" button. The right sidebar contains sections for "Toolbar", "Featured Resources" (Node.js, Aggregations in Node.js, Semantic Search with Atlas Vector Search, More Javascript Content), and "Sample Apps" (Node.js, MERN Stack, MEAN Stack, Remix Stack).

Disadvantages Before implementing online complaint care:

- 1.Limited Reach:** *Traditional channels restrict easy complaint submission.*
- 2.Slower Response Time:** *Complaints take longer to resolve*
- 3.Higher Costs:** Manual handling increases operational expenses.
- 4.Lack of Transparency:** *No visibility into complaint progress.*
- 5.Limited Insights:** *Missed data analysis for improvements.*
- 6.Increased Workload:** *Staff overwhelmed by manual processes.*
- 7.Frustration:** *Customers frustrated by slow resolutions.*

APPLICATIONS:



Customer Service:

Businesses use it to track and resolve customer issues, improving service quality and satisfaction.



Education:

Educational institutions use it to handle complaints from students, faculty, and staff, addressing issues related to facilities, teaching, or administration.



Healthcare:

Hospitals and clinics manage patient complaints regarding care quality, appointment scheduling, and billing concerns.



Transportation:

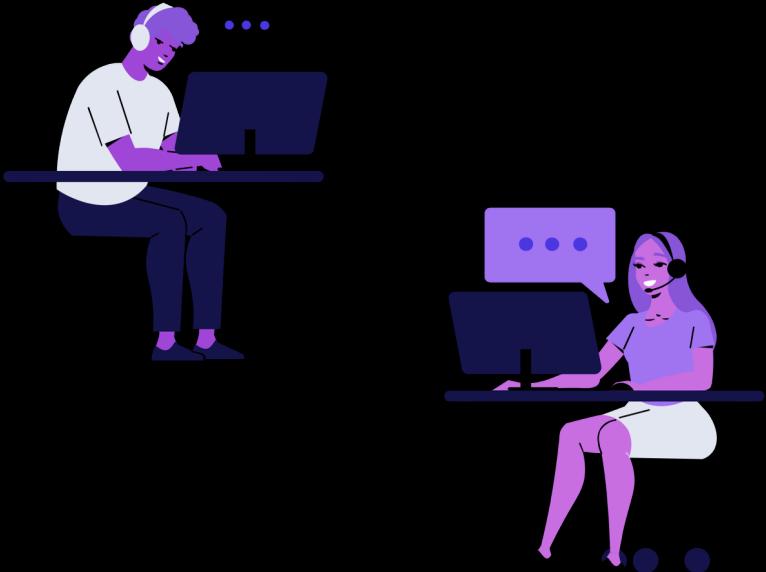
Transport services (airlines, railways, etc.) handle complaints related to delays, lost luggage, or service quality.



E-commerce:

Online retailers use it to manage customer complaints regarding orders, deliveries, and product quality.

CONCLUSION



The Online Complaint Registration and Management System was designed to create a seamless experience for handling customer concerns. By simplifying complaint submission and tracking, it enables organizations to respond swiftly and transparently, fostering customer trust. For businesses, it not only improves operational efficiency but also provides actionable insights to enhance service quality over time. The system transforms complaint management into a streamlined, data-driven process that strengthens relationships and supports continuous improvement, making it an essential tool for delivering reliable and responsive service in today's competitive landscape.

"If you take care of your customers, they will take care of you," -- Richard Branson

PRESENTED BY:

MOHAMED FAHIM I

MOHAMMED AZEEM R

MOHAMED ASIF M

MOHAMED FAWZAN S

