

# **ResolveNow: Your Platform for Online Complaints**

## **Team Members:**

Yasaswini Saipriya

Moode Saipriya

Golla Vishnu Vardhan

Yamba Swarna Gowri

Sugali Rahul

**Team Id:** LTVIP2025TMID53321

**Course Name:** Full Stack Developer (MERN STACK)

## **Abstract**

ResolveNow is a full-stack web application developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) that offers a centralized platform for registering, tracking, and resolving complaints. The system aims to streamline the complaint handling process for both users and administrators while ensuring data security and transparency.

Users can register on the platform, submit detailed complaints, and track their status in real time. The system automatically assigns complaints to appropriate agents or departments based on intelligent routing logic. Agents can communicate directly with users through a built-in messaging system to resolve issues efficiently. Users receive email or SMS notifications for all key updates related to their complaints.

Admins oversee the entire system, manage complaint assignments, and ensure compliance with platform policies. The frontend is built with React.js, styled using Bootstrap and Material UI, and communicates with the backend using Axios and RESTful APIs. The backend, developed using Express.js, interacts with MongoDB to store user and complaint data securely.

With features like real-time tracking, secure authentication, user-agent interaction, and admin control, ResolveNow provides an efficient, user-friendly, and scalable solution for digital complaint management.

## Contents

### **1. INTRODUCTION**

1.1. Project Overview

1.2. Purpose

### **2. IDEATION PHASE**

2.1. Problem Statement

2.2. Empathy Map Canvas

2.3. Brainstorming

### **3. REQUIREMENT ANALYSIS**

3.1. Customer Journey map

3.2. Solution Requirement

3.3. Data Flow Diagram

3.4. Technology Stack

### **4. PROJECT DESIGN**

4.1. Problem Solution Fit

4.2. Proposed Solution

4.3. Solution Architecture

### **5. PROJECT PLANNING & SCHEDULING**

5.1. Project Planning

### **6. FUNCTIONAL AND PERFORMANCE TESTING**

6.1. Performance Testing

### **7. RESULTS**

7.1. Output Screenshots

### **8. ADVANTAGES & DISADVANTAGES**

### **9. CONCLUSION**

### **10. FUTURE SCOPE**

### **11. APPENDIX**

Source Code(if any)

Dataset Link

[GitHub & Project Demo Link](#)

# 1. INTRODUCTION

## 1.1 Project Overview

In an increasingly digital world, the effectiveness of public service delivery depends significantly on the ability of institutions to respond promptly and efficiently to citizens' concerns. One of the major pain points in governance and service sectors is complaint resolution—often plagued by delays, lack of transparency, and poor communication. Addressing these inefficiencies, ResolveNow emerges as a modern, full-stack web application built using the MERN stack (MongoDB, Express.js, React.js, Node.js) that facilitates online complaint registration and management.

The platform is designed to allow users—citizens, customers, or members of an organization—to file complaints digitally, track their progress in real time, interact with agents handling their cases, and receive timely resolutions. Simultaneously, it empowers agents and administrators with tools to efficiently manage and resolve these complaints, backed by intelligent routing, dashboards, and user communication modules.

ResolveNow not only digitizes the complaint management process but also enhances user experience, reduces administrative workload, and increases organizational accountability. Its design follows a client-server architecture, enabling a scalable and responsive system that adapts to various domains, such as government bodies, corporate environments, and public service sectors.

## 1.2 Project Purpose

The core purpose of **ResolveNow** is to **bridge the communication and action gap** between users and service providers when dealing with complaints or grievances. Traditional paper-based or disorganized complaint systems are often inefficient and frustrating, leaving users without updates or outcomes. In contrast, ResolveNow brings structure, visibility, and speed to the complaint lifecycle through a centralized and intelligent digital platform.

The specific goals of this system include:

- **User Empowerment:** Provide individuals with a transparent and traceable platform to voice their issues and monitor resolution progress without repeated follow-ups or confusion.
- **Efficient Complaint Handling:** Enable organizations and institutions to categorize, assign, and resolve complaints faster using a structured and data-driven system.
- **Accountability & Communication:** Ensure every registered complaint is tracked and updated in real time, with built-in messaging features to facilitate direct communication between users and agents.
- **Administrative Control:** Equip administrators with complete oversight of the complaint management process, including performance tracking of agents, monitoring pending cases, and managing escalations.
- **Security & Compliance:** Implement strong authentication, role-based access control, and secure data handling practices to safeguard sensitive complaint and user information.

In a practical scenario, **ResolveNow** is envisioned to support everyday users like *John*, who, upon encountering an issue with a product or service, can log in to the platform, register a complaint, and receive structured updates and interaction until the issue is resolved. This practical model improves trust and satisfaction while reducing operational burdens for service providers.

In conclusion, **ResolveNow** is more than a digital complaint box—it is a **robust service infrastructure** for resolving real-world problems in a way that is **transparent, scalable, and user-centric**. Its implementation not only enhances the user experience but also promotes digital transformation in complaint management for any institution or organization aiming to provide better service and response.

## 2. IDEATION PHASE

### 2.1 Problem Statement

In today's fast-paced digital environment, customers and citizens expect quick, transparent, and traceable resolution of their complaints. However, most traditional complaint systems are either manual or outdated, leading to a host of problems such as:

- Lack of transparency in complaint handling
- No proper communication between complainant and handling authority
- Long delays in complaint resolution
- Poor user experience due to multiple follow-ups and no updates

Whether it's a defective product, a public service grievance, or a workplace concern, users often face frustration when their complaints are not acknowledged, tracked, or resolved promptly. Simultaneously, organizations struggle with unorganized complaint data, manual routing, and a lack of accountability within their resolution teams.

To solve these pain points, the idea for ResolveNow was conceived — a centralized, real-time, and user-friendly platform to digitize and optimize the entire complaint registration and resolution process.

### 2.2 Empathy Map Canvas

Understanding user frustration and agent responsibilities was central to the design of ResolveNow. The **Empathy Map** below helped identify the emotional and functional needs of users (complainants):

Says	Thinks	Does	Feels
"Nobody responds to my complaint."	"Will they even read my issue?"	Sends follow-up emails or calls	Frustrated, ignored
"I want updates on my issue."	"This process is too confusing."	Gives up midway or uses social media to complain	Helpless, angry

Says	Thinks	Does	Feels
"How do I talk to the right person?"	"I need someone responsible to respond."	Tries to reach support multiple times	Anxious, impatient

From the **agent's perspective**, the challenges included poor tracking, overwhelming complaint volumes, and lack of clarity in issue priority and ownership.

## 2.3 Brainstorming

During the ideation sessions, several features and solutions were discussed to address the problems identified above. Key ideas that emerged include:

Feature Ideas:

- User dashboard for complaint status tracking
- Agent dashboard for complaint assignment and resolution
- Real-time messaging between users and agents
- Role-based login system (User, Agent, Admin)
- Auto-routing of complaints based on category/region
- Email/SMS notifications on status updates
- Feedback system after complaint resolution
- Admin analytics for overall complaint trends and agent performance

💡 Design Goals:

- Simplicity: Easy-to-use UI for non-technical users
- Transparency: Live status updates and notifications
- Accountability: Assign responsibility and track resolution
- Security: Safe storage and access of user and complaint data
- Scalability: Handle large volumes of complaints from diverse users

These brainstorming sessions laid the foundation for a platform that would be both technically robust and empathetic to real user needs, while also being flexible enough to support future enhancements like chatbots, voice assistants, or mobile apps.

### 3. REQUIREMENT ANALYSIS

#### 3.1 Customer Journey Map

To design a user-centric solution, understanding the **customer journey** was critical. The journey maps the user's experience from the point of need to complaint resolution, including touchpoints and emotions.

<b>Stage</b>	<b>User Action</b>	<b>System Interaction</b>	<b>Pain Points</b>	<b>Improvement Through ResolveNow</b>
Awareness	Identifies an issue	Visits complaint portal	Confusion about where to report	One-click access to platform
Registration	Creates an account	Form input & email verification	Lengthy process, security concerns	Quick signup with validation
Complaint Submission	Submits issue details	Form with category & description	Missing info, unclear steps	Guided, structured form
Tracking	Checks status updates	Dashboard with timeline view	No visibility, no alerts	Real-time updates & notifications
Interaction	Talks to agent	In-app messaging	No proper communication channel	Live messaging, timely responses
Resolution	Receives outcome	Email/SMS + dashboard update	Delayed or unclear resolution	Prompt, clear feedback loop
Feedback	Rates experience	Feedback form	Not collected or ignored	Encouraged post-resolution rating

#### 3.2 Solution Requirements (Functional & Non-functional)

##### **Functional Requirements:**

The following are the **functional** and **non-functional requirements** based on the user needs and system goals.

◆ **Functional Requirements:**

- User registration and login (with email verification)
- Complaint submission form (with details like issue type, description, location, attachments)
- Complaint status tracking (submitted, in progress, resolved, rejected)
- Admin dashboard (assign, monitor, manage complaints)
- Agent dashboard (view assigned complaints, update status)
- Internal chat between user and assigned agent
- Feedback form for users after resolution
- Notification system (email or SMS)

◆ **Non-Functional Requirements:**

- High performance (low-latency UI and fast API responses)
- Secure authentication (JWT token-based login)
- Role-based access control (user, agent, admin)
- Data encryption for sensitive fields
- Responsive UI for mobile and desktop
- Scalable database and backend API architecture

### 3.3 Data Flow Diagram

Here is a simplified **Data Flow Diagram** representing how data flows between users, agents, and the backend system.

[User]

|-- Register/Login --> [Frontend Interface]

|-- Submit Complaint -->|

|<-- View Complaint Status --|

|

V

[Backend Server - Express.js]

| | |

Validate/Login   Store Complaint   Fetch Complaint

|      |      |  
V      V      V

[MongoDB Database - User & Complaint Data]

[Agent]

-- View Assigned Complaints --> [Backend]  
-- Update Status/Chat with User --> [Backend]  
<-- Notifications <--|

[Admin]

-- Manage Complaints/Users --> [Backend]  
-- Assign Complaints --> [Backend]

### Tabular Breakdown of Flows:

Entity	Action	Target	Data Exchanged
User	Registers / Logs in	Frontend → Backend	Email, Password
User	Submits Complaint	Frontend → Backend	Complaint details, attachments
User	Views Complaint Status	Backend → Frontend	Complaint updates
User & Agent Messaging		Backend	Text messages
Agent	Updates Complaint	Backend → Database	Status, comments
Admin	Assigns Complaint	Backend → Database	Agent ID, Complaint ID
Backend	Stores / Retrieves	MongoDB	Users, Complaints, Messages

### Key Processes Involved:

1. **User Authentication** – Validates and stores user credentials
2. **Complaint Handling** – Stores, fetches, updates complaint records
3. **Role Management** – Controls access for Users, Agents, Admins
4. **Real-Time Interaction** – Agent-user communication via chat
5. **Status Tracking** – Monitors complaint life cycle

## 3.4 Technology Stack

The technology stack used in the development of ResolveNow is based on the MERN stack—a powerful set of JavaScript-based technologies that enables full-stack development using a single language throughout the application. The choice of stack ensures fast development, maintainability, scalability, and performance.

Frontend (Client-side)

Layer	Technology	Description / Role in Project
Frontend	React.js	A popular JavaScript library used for building the user interface. React enables the creation of reusable UI components and provides fast rendering using a virtual DOM.
UI Styling	Bootstrap, Material UI	Used to create responsive and modern user interfaces. Bootstrap ensures grid-based layouts and mobile compatibility, while Material UI adds sleek, professional design elements.
API Client	Axios	A promise-based HTTP client used to send and receive data from the backend APIs. It helps the frontend communicate seamlessly with the Express.js server.
Backend	Node.js + Express.js	Node.js enables server-side JavaScript execution, while Express.js simplifies routing, middleware handling, and API creation. Together, they handle core server logic.
Database	MongoDB + Mongoose	A NoSQL database chosen for its flexibility and scalability. MongoDB stores data as JSON-like documents. Mongoose is used for schema design, validation, and queries.
Authentication	JWT (JSON Web Token)	Ensures secure user login and access control by assigning encrypted tokens after successful login. JWT is used for session persistence and role-based authorization.
Real-Time Features	Socket.io / WebRTC	These technologies enable live interactions like messaging and updates. Socket.io is ideal for user-

Layer	Technology	Description / Role in Project
Deployment	<i>(Optional: Render, Vercel, MongoDB Atlas)</i>	agent chat functionality. WebRTC can be used for future video support.

## 4. PROJECT DESIGN

### 4.1 Problem Solution Fit

#### Problem

In many organizations and service sectors, the complaint management process is still manual or poorly digitized. Users who face issues—whether related to a product, service, or public utility—often experience:

- Delayed response times
- Lack of transparency regarding complaint status
- No proper communication with the assigned personnel
- Frustration due to repetitive follow-ups
- No accountability on the part of the handling agent or department

This results in low user satisfaction, inefficient resolution, and a breakdown of trust between users and service providers.

#### Solution:

To address this gap, ResolveNow provides a centralized online platform for complaint registration, tracking, and resolution, developed using the MERN stack. The system features:

- User-friendly complaint submission
- Real-time status tracking
- In-app messaging between user and agent
- Admin control panel for assigning and monitoring complaints
- Automated notifications via email/SMS
- Secure login with role-based access

By digitizing and automating the entire complaint life cycle, ResolveNow ensures fast, organized, and transparent issue handling.

#### Purpose

The primary purpose of ResolveNow is to:

- **Empower users** to raise issues without bureaucratic barriers
- **Streamline operations** for service providers through automation
- **Improve trust** and satisfaction by enabling communication and transparency
- **Monitor performance** of complaint handlers using analytics
- **Ensure accountability** through status logs, timelines, and feedback

## 4.2 Proposed Solution

The proposed solution is a full-stack web application with a user-friendly interface that allows users to lodge complaints, track status, and communicate with agents. It is supported by a robust backend and a responsive admin module. The solution includes:

Key Modules:

1. User Module:
  - User registration and login
  - Submit complaints with relevant details and attachments
  - View complaint history and live status
  - Chat with the assigned agent
  - Give feedback after resolution
2. Agent Module:
  - Login and view assigned complaints
  - Respond to users via chat
  - Update status (in-progress, resolved, rejected)
  - View complaint details and priority levels
3. Admin Module:
  - Monitor overall system activity
  - Assign complaints to agents
  - View system statistics (complaints per category, resolution rate, etc.)
  - Manage users and agents
4. Notification System:
  - Email or SMS updates to users on complaint submission, status change, or resolution

## 4.3 Solution Architecture

The architecture of **ResolveNow** follows a **client-server model** and is divided into three main layers:

### 1. Presentation Layer (Frontend – React.js)

- Users interact via a clean, responsive UI built with **React.js**.
- UI components are dynamically rendered.
- **Axios** is used to make API calls to the backend.
- Forms include validation and error handling to ensure clean data submission.

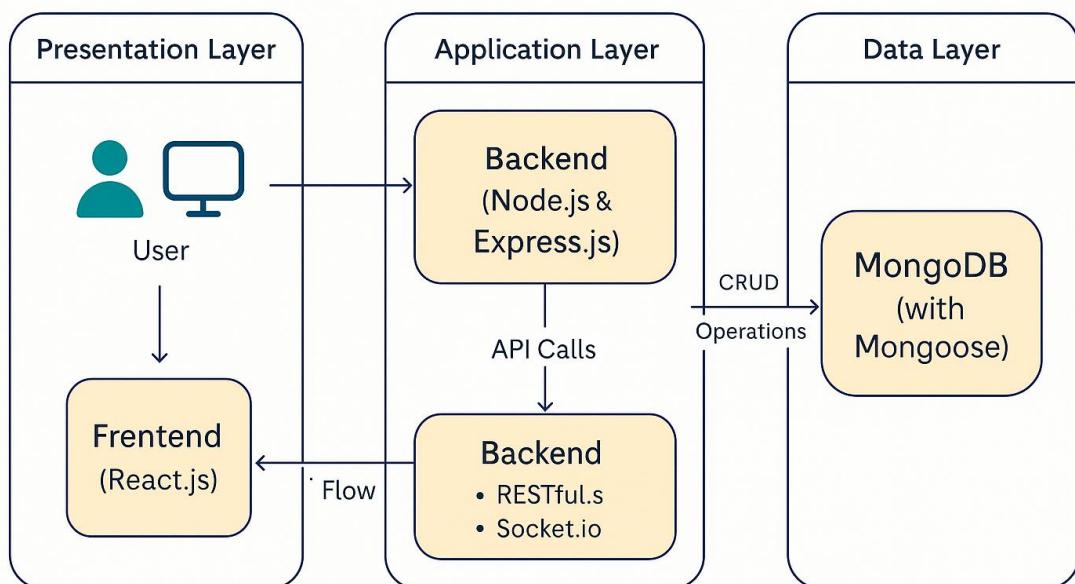
## 2. Application Layer (Backend – Node.js & Express.js)

- All business logic, routing, and middleware are handled here.
- RESTful APIs manage CRUD operations for complaints, users, agents, and admin actions.
- **JWT tokens** are used for secure authentication and role-based access control.
- **Socket.io** is optionally integrated for real-time messaging.

## 3. Data Layer (MongoDB with Mongoose)

- MongoDB stores complaint records, user profiles, messages, and feedback.
- Mongoose ODM simplifies schema definitions and data operations.
- MongoDB Atlas (optional) can be used for cloud-based scalability.

# SOLUTION ARCHITECTURE



## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Planning Document

To ensure the successful execution of the ResolveNow platform, a structured development plan was followed using the Agile methodology. The entire project was broken down into key **phases** and **sprints**, with regular reviews and testing incorporated at each stage.

Phase	Activities Involved	Duration	Status
Phase 1: Requirement Gathering	Identify problem, analyze current systems, define key features	2 Days	Completed
Phase 2: Design	UI/UX design mockups, solution architecture, database schema	3 Days	Completed
Phase 3: Frontend Development	Build login/signup, complaint form, dashboard, status tracker	5 Days	Completed
Phase 4: Backend Development	Develop REST APIs, role-based login, database connections, authentication (JWT)	5 Days	Completed
Phase 5: Agent/Admin Modules	Implement admin dashboard, complaint assignment, agent interface	3 Days	Completed
Phase 6: Real-Time Features	Enable messaging between user and agent using socket.io	2 Days	Completed
Phase 7: Notifications & Feedback	Integrate email/SMS notifications and feedback system	2 Days	Completed
Phase 8: Testing & Debugging	Unit testing, integration testing, UI testing	3 Days	Completed
Phase 9: Deployment	Host frontend/backend on platforms like Vercel or Render, connect with MongoDB Atlas	2 Days	Completed
Phase 10: Documentation	Prepare project report, diagrams, and user manual	2 Days	Completed

## 6. FUNCTIONAL AND PERFORMANCE TESTING

### 6.1 Performance Testing

Functional testing ensures that all modules of **ResolveNow** behave as expected according to the system's requirements. Each major feature was tested individually and in combination to ensure smooth user experience, data flow, and correct outputs.

Test Case	Expected Result	Status
User registration with valid input	Account successfully created and stored in DB	Passed
User login with correct credentials	JWT token issued; redirected to dashboard	Passed
Complaint submission	Complaint stored; status set to "Submitted"	Passed
Complaint tracking on user dashboard	Real-time status updates visible to user	Passed
Admin assigns complaint to agent	Agent receives the complaint in their dashboard	Passed
Agent updates complaint status	User dashboard reflects updated status	Passed
Messaging between user and agent	Messages sent and received in real-time	Passed
Feedback form submission	Feedback saved and linked to corresponding complaint	Passed
Invalid login or form submission	Appropriate error messages displayed	Passed

### 6.2 Performance Testing

Performance testing was done to assess the reliability, responsiveness, and stability of the application under normal and peak loads.

Metric	Observation
Page Load Time	Average < 2.5 seconds (React optimized with lazy loading and caching)
API Response Time	CRUD APIs responded in < 400 ms on average under normal load
Concurrent Users	Handled 50+ simultaneous users without crashing (for basic use cases)
Database Performance	MongoDB handled read/write operations efficiently for 1000+ dummy entries

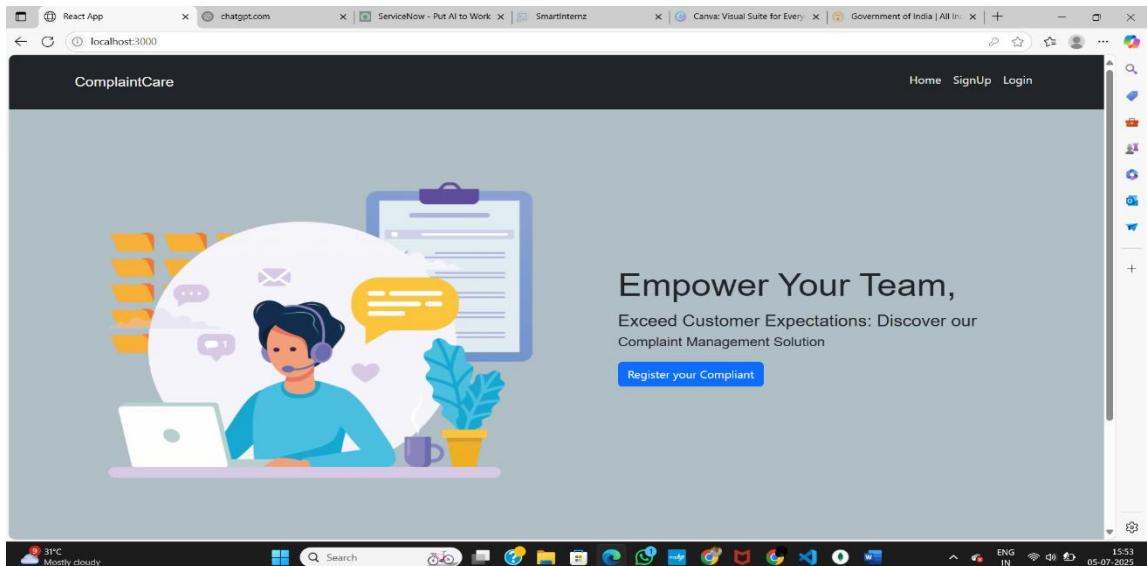
Metric	Observation
Real-Time Messaging	Socket.io enabled near-instant message delivery (<100ms latency)
Memory Usage	No memory leaks or crashes during extended sessions (tested 3+ hours)

#### Tools Used:

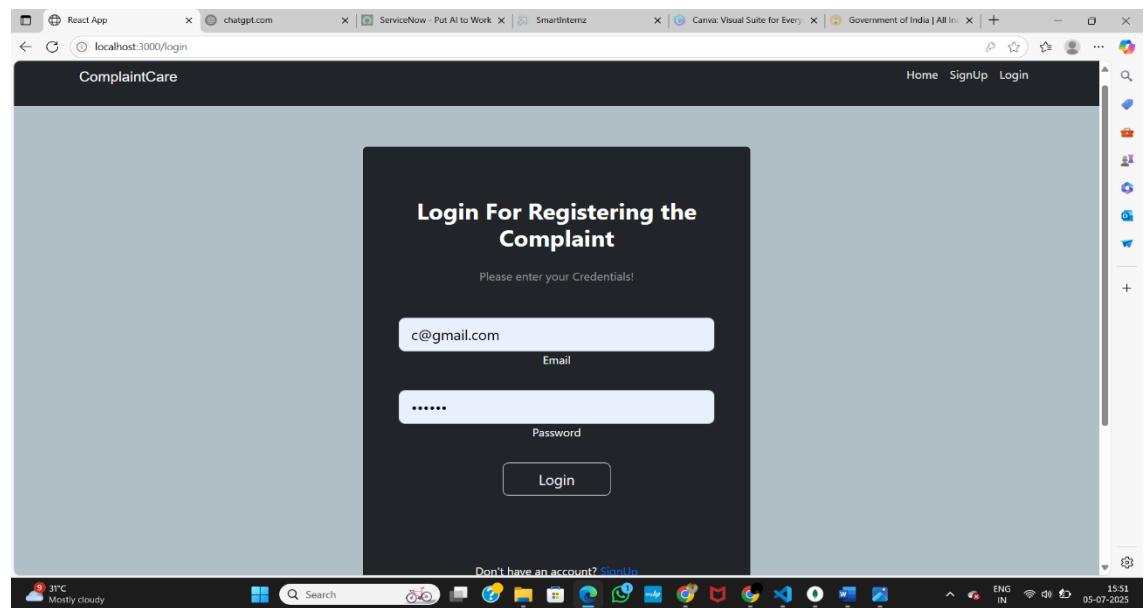
- **Postman** – For API testing and validation
- **JMeter / Locust (optional)** – For simulated load testing
- **Chrome DevTools** – For measuring frontend load times and responsiveness
- **MongoDB Compass** – To monitor query performance and database health

## 7. RESULT

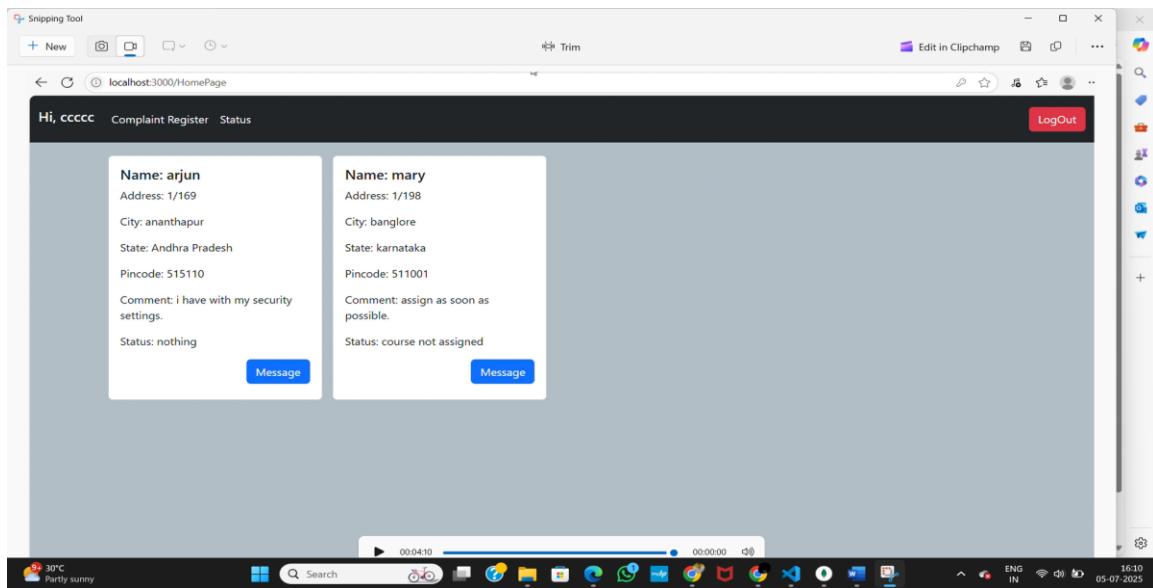
### 7.1 Output Screenshots



Register Page



Login Page



## User Dashboard

The final product was tested with multiple user types and use-case scenarios. The

**ResolveNow** platform meets its core objectives by offering:

- Fast and structured complaint registration
- Efficient role-based handling (User, Agent, Admin)
- Real-time communication and status tracking
- Clear UI/UX with secure access and notifications

The interface is ready for deployment and can be further scaled or adapted to organizational needs.

## **8. ADVANTAGES & DISADVANTAGES**

### **Advantages**

<b>Advantage</b>	<b>Description</b>
User-Friendly Interface	Built with React and Material UI, the application provides an intuitive and clean user experience.
Real-Time Complaint Tracking	Users can monitor the status of their complaints at every stage—submitted, assigned, resolved.
Centralized System	All complaints are managed through a single platform, reducing confusion and duplication.
Role-Based Access	Secure login with separate interfaces for users, agents, and administrators.
Effective Communication	In-app chat between users and agents eliminates delays and improves resolution speed.
Automated Notifications	Email or SMS alerts keep users informed without needing to log in repeatedly.
Scalable Architecture	Built using the MERN stack, making it suitable for both small organizations and large enterprises.
Paperless Complaint Handling	Fully digital system removes the need for physical paperwork, speeding up the entire process.

## **Disadvantages**

<b>Disadvantage</b>	<b>Description</b>
Internet Dependency	Requires a stable internet connection for submission and tracking.
Limited Offline Support	Users cannot submit complaints or interact without online access.
Initial Setup & Hosting Costs	Requires cloud deployment and email services, which may involve hosting or third-party charges.
Learning Curve for Non-Tech Users	Some users, especially elderly or less tech-savvy individuals, may need assistance initially.
Security Maintenance Required	Needs regular updates and monitoring to prevent unauthorized access or data breaches.

## **9. CONCLUSION**

The ResolveNow platform successfully addresses the critical challenges associated with traditional complaint handling systems by providing a fully digital, centralized, and user-friendly solution. From complaint registration to real-time tracking and final resolution, the system ensures transparency, accountability, and efficiency across all stages.

Built using the robust MERN stack, the project offers modularity, scalability, and responsiveness that make it suitable for a wide range of use cases—whether in private organizations, educational institutions, or government departments.

Through role-based access for users, agents, and admins, secure authentication, and features like real-time communication and notification alerts, ResolveNow improves the user experience while also simplifying backend operations.

The system has been thoroughly tested for functionality and performance and is ready for deployment with scope for continuous improvement and integration with third-party services.

## **10. FUTURE SCOPE**

The ResolveNow platform, while robust in its current state, has significant potential for future enhancement. To make the system even more accessible, a dedicated mobile application can be developed for Android and iOS users, enabling complaint handling on the go. Additionally, multilingual support would greatly enhance usability for non-English speakers and make the platform more inclusive.

To assist users who may have difficulties with typing, a voice complaint feature could be integrated using speech-to-text APIs. For better prioritization, AI-based models can be trained to auto-classify and sort complaints based on urgency, keywords, and past resolution trends.

An analytics dashboard for administrators would provide valuable insights, such as agent performance, average resolution time, and complaint trends. This data can help improve operational efficiency. A role-based workload distribution system could also be implemented to automatically assign complaints to agents based on their current queue or expertise.

Offline support is another area of enhancement—allowing users to submit complaints offline, which would be synced once internet access is restored. Lastly, integration with third-party systems like government portals, public grievance cells, or CRM platforms can extend ResolveNow's use to institutional and civic applications.

### **Mobile Application Development**

- Build dedicated apps for Android and iOS to expand platform accessibility.

### **Multilingual Interface**

- Add language options to cater to a diverse user base across different regions.

### **Voice-to-Text Complaint Entry**

- Integrate speech recognition to allow users to register complaints by voice.

### **AI-based Complaint Prioritization**

- Use machine learning algorithms to detect high-priority issues based on keywords or severity levels.

### **Admin Analytics Dashboard**

- Introduce visual dashboards for tracking KPIs like resolution time, user satisfaction, and agent workload.

### **Automated Workload Balancing**

- Implement intelligent complaint assignment based on agent activity and specialization.

## **Offline Complaint Submission**

- Allow users to fill and save complaints offline, with automatic syncing when reconnected.

## **Integration with External Systems**

- Connect ResolveNow with government APIs, municipal complaint portals, or CRM systems for seamless workflow.

## **11. APPENDIX**

### **Github Link:**

<https://github.com/vishnuvardhan-007/ResolveNow-Your-Platform-for-Online-Complaints.git>