ONLINE COMPLAINT REGISTRATION

Project Title:

ResolveNow : Your Platform for Online Complaints

Team ID : LTVIP2025TMID42578

Team Members

Name	Email	Hall Ticket No	Branch	Track
Allaka Hari Prakash	Hariprakash28703@gmail.com	22P31A0573	Computer Science & Engineering	Full Stack (MERN)
Amarthi Satya	amarthisatya@gmail.com	22P31A0574	Computer Science & Engineering	Full Stack (MERN)
Bade Navya	Navya3424@gmail.com	22P31A0581	Computer Science & Engineering	Full Stack (MERN)
Challa Bhargav	Challabhargav06@gmail.com	22P31A0584	Computer Science & Engineering	Full Stack (MERN)

1. INTRODUCTION

🔷 1.1 Project Overview (Integrated for ResolveNow)

ResolveNow is a platform built to revolutionize how customer complaints are handled, from submission to resolution. Traditional systems rely heavily on **manual intervention**, which leads to delays, misrouted complaints, and uneven agent workloads. These inefficiencies directly affect user satisfaction and strain support teams.

To address these issues, ResolveNow integrates **automated, skill-based, and rule-driven complaint routing** that intelligently directs complaints to the most qualified agent or department. This approach brings multiple benefits:

- **Faster Response Time:** Complaints are automatically routed based on issue type, category, and urgency—reducing delays.
- **Skill-Based Assignment:** Issues are matched with agents who have the exact skills required to handle them.
- **Balanced Workload Distribution:** The system avoids overloading some agents while others are idle, ensuring fair and efficient operations.
- **Reduced Reassignments:** Smart routing logic reduces misrouted tickets, cutting down on time wasted in reassignments.

By implementing intelligent ticket handling, ResolveNow ensures that users experience a **faster, smoother, and more transparent resolution process**, while support teams operate with **greater efficiency and visibility**.

🔷 1.2 Purpose (Integrated for ResolveNow)

The primary purpose of the ResolveNow platform is to **eliminate inefficiencies in traditional complaint management** by introducing a streamlined, automated system that routes complaints intelligently and reduces manual dependency.

Traditional systems often suffer from:

- **Manual triage and routing**, which delays resolution.
- **Skill mismatches**, where agents receive complaints outside their expertise.
- **Uneven workload distribution**, leading to agent burnout or underutilization.
- **Lack of visibility**, preventing real-time monitoring of assignment and resolution effectiveness.

ResolveNow addresses these issues by building a **scalable, intelligent assignment framework** that automates complaint distribution based on:

- Complaint category and urgency,

- Agent skills and availability,
- Real-time workload data,
- User location or department (if applicable).

This approach enhances the platform's ability to:

- **Improve resolution time**,
- **Maximize agent productivity**,
- **Boost end-user satisfaction**, and
- **Ensure better SLA (Service Level Agreement) adherence**.

Ultimately, the purpose of ResolveNow is to create a **more responsive, transparent, and efficient complaint handling system** that benefits both end-users and support teams alike.

2. IDEATION PHASE

🔷 2.1 Problem Statement

Despite the availability of digital platforms, many online complaint systems continue to suffer from **manual processes**, **misrouted tickets**, and **lack of automation**, which leads to inefficiencies and poor user experiences.

The key problems identified in current complaint handling systems include:

- **Excessive Manual Intervention:** Frontline agents or admins often have to manually review and assign complaints, causing delays and increased chances of human error—especially during peak hours.
- **Misassignment and Re-routing:** Without intelligent logic, complaints are frequently routed to agents lacking the necessary expertise, leading to repetitive reassignments and extended resolution times.
- **Uneven Workload Distribution:** Some agents are overburdened with complaints, while others are underutilized. This imbalance can cause burnout and inefficiency in support operations.
- **Lack of Skill-Based Matching:** Traditional systems fail to map complaints to agents based on specific technical or functional skills, causing wasted time and poor user satisfaction.
- **Limited Transparency & Reporting:** There is minimal real-time visibility into how complaints are being assigned, which agents are overloaded, or where delays are occurring. This hampers managerial decision-making and performance optimization.
- **Scalability Constraints:** As the number of users and complaints grows, manual or basic routing mechanisms become increasingly unsustainable, slowing down the entire support process.

These problems result in **longer complaint resolution times**, **low agent productivity**, and **frustrated end-users**. The ResolveNow platform addresses these challenges by introducing an **intelligent, rule-based complaint assignment system** designed for speed, accuracy, and fairness.

🧭 2.2 Empathy Map

Understanding the emotions and challenges faced by key stakeholders helps us design a more effective and user-centered complaint handling experience in ResolveNow. Below are empathy maps for each role in the complaint process.

🧑‍💼 End User (Customer)

Says	Thinks	Does	Feels
"Why hasn't anyone responded yet?"	"Did my complaint even go through?"	Refreshes the status page repeatedly	Anxious, ignored
"They reassigned my complaint again?"	"Do they even know what they're doing?"	Tries to escalate or contact support	Frustrated, impatient
"I just want this resolved quickly."	"Is anyone even working on this?"	Logs into the portal for updates	Hopeless, undervalued

🧑‍💻 Agent (Support Staff)

Says	Thinks	Does	Feels
"This complaint isn't even in my area."	"Why do I keep getting irrelevant complaints?"	Reassigns tickets or consults with other agents	Overloaded, underutilized
"I'm already swamped."	"Others are free, why is everything coming to me?"	Rushes responses to clear backlog	Burned out, pressured
"I wish complaints matched my actual skill set."	"There has to be a smarter way to assign work."	Spends time figuring out the right assignee	Unproductive, discouraged

🧑‍💼 Admin / Manager

Says	Thinks	Does	Feels
"Our resolution time is too slow."	"Why are we missing SLA targets?"	Reviews metrics and agent performance	Stressed, under pressure
"Too many reassignments lately."	"Is our routing logic broken?"	Manually reassigns or audits complaints	Frustrated, reactive
"Some agents are overwhelmed, others are idle."	"How do I balance workload efficiently?"	Tries redistributing work manually	Concerned about morale & fairness

These insights helped shape the ResolveNow platform to be more **empathetic, intelligent, and automated**, aiming to improve **user experience, agent efficiency**, and **managerial control**.

💠 2.3 Brainstorming

To design a smarter, faster, and more reliable complaint management system, our team conducted structured brainstorming sessions focused on identifying root problems and innovative solutions. Stakeholders included developers, UI/UX designers, and support agents, ensuring diverse perspectives.

🔍 Key Themes & Ideas Generated

1. **Automated Complaint Assignment**
    - Use category, subcategory, and keywords from user-submitted complaints to automatically assign them to the most appropriate agent or department.
    - Integrate logic using rule-based assignment and condition-based flows.
2. **Skill-Based Agent Routing**
    - Define skills (e.g., “Delivery Issues”, “Billing Problems”, “Technical Errors”) for each agent.
    - Assign complaints based on both category and agent skill match.
3. **Real-Time Workload Management**
    - Track the number of active complaints per agent and distribute new complaints fairly.
    - Use round-robin or least-busy strategies to prevent overload.

4. **Dynamic Reassignment Logic**
    - Automatically reassign unresolved complaints after a timeout.
    - Escalate priority complaints to more experienced agents.
5. **User Experience Improvements**
    - Notify users immediately after complaint assignment with the agent/team handling their case.
    - Display live updates and statuses through the dashboard or via SMS/email notifications.
6. **Manager Dashboards & Reporting**
    - Build visual dashboards to monitor:
        - Average assignment time
        - Reassignment rates
        - Agent workload
        - SLA compliance
    - Use this data to optimize performance and adjust rules dynamically.
7. **Future Enhancements**
    - Explore integration with AI/ML to suggest assignment paths using historical data (predictive assignment).
    - Enable voice note and image upload features in complaints to enrich data for routing.

🔷 3. REQUIREMENT ANALYSIS

To effectively implement an intelligent and efficient complaint handling system in ResolveNow, a clear set of functional and non-functional requirements was defined. These requirements ensure that the platform remains scalable, secure, and user-centric.

✅ 3.1 Functional Requirements

Code	Requirement	Description
FR1	Automated Complaint Assignment	The system must auto-assign complaints based on category, keywords, and urgency.

Code	Requirement	Description
FR1.1	Category-Based Routing	Complaints must be routed based on pre-defined categories (e.g., Billing, Product Issue).
FR1.2	Keyword Analysis	Keywords from user descriptions help determine assignment group or agent.
FR2	Skill-Based Routing	Assign complaints to agents who possess relevant skills.
FR2.1	Skill Mapping	Admins must be able to define and assign skills to agents.
FR2.2	Match Skills to Complaint	System matches skill requirement with available agents.
FR3	Workload Balancing	Assignments must consider agent load to prevent overburdening.
FR4	Re-assignment Logic	Allow automated or manual reassignment based on timeout or escalation rules.
FR5	Manual Override	Admins or managers can override assignments when necessary.
FR6	Notifications	Notify both agent and user upon assignment or reassignment.
FR7	Audit Logging	Every assignment/reassignment must be logged in the activity history.
FR8	Dashboard & Reporting	System should support dashboards showing assignment metrics (time, reassignment rate, load distribution).

⚙ 3.2 Non-Functional Requirements

Code	Requirement	Description
NFR1	Performance	Assignment logic must execute in < 5 seconds after complaint submission.
NFR2	Scalability	System must handle growth in user base and complaint volume.
NFR3	Configurability	Admins must be able to update assignment rules without code changes.
NFR4	Reliability	Assignment logic must function consistently across complaint types.
NFR5	Maintainability	Code should be modular and easy to update or debug.

Code	Requirement	Description
NFR6	Security	Only authorized users can configure rules or access sensitive data.
NFR7	Usability	The interface for managing assignments should be intuitive and admin-friendly.

🔷 3.3 Customer Journey Map

To ensure a smooth and satisfactory user experience, we mapped the typical complaint journey from a customer's perspective. This helped us identify pain points and opportunities for automation.

🎗️ Scenario: User Submits a Complaint About a Product Issue

Step	User Action	User Goal	Current Pain Point	ResolveNow Improvement
1. Submission	User fills out the complaint form describing the issue.	Log a complaint easily.	Uncertainty about whether it's received or assigned.	Instant acknowledgment with automated assignment.
2. Initial State	Complaint enters the system, status set to "New".	Receive confirmation.	Waiting with no updates.	Status auto-updated to "Assigned to Agent" with tracking.
3. Assignment	System matches complaint with the right agent.	Get help quickly.	Often misrouted or delayed.	Skill-based, auto-routing based on category and keywords.
4. Response	Agent contacts the user to resolve the issue.	Communicate with support.	No clear ownership of issue.	Assigned agent's name and contact shown to user.
5. Resolution	Agent resolves and closes the complaint.	Problem solved.	Reassignments delay resolution.	First-contact resolution is prioritized.

🔷 3.4 Technology Stack

ResolveNow is developed using a **modern full-stack architecture** optimized for real-time data flow, dynamic user interaction, and scalability.

🛠️ Core Technologies Used

Layer	Technology	Purpose
Frontend	React.js	Component-based UI for user and agent dashboards
UI Libraries	Bootstrap, Material UI	Responsive design and consistent styling
Backend	Node.js + Express.js	REST API server and routing logic
Database	MongoDB	Document-based storage for users, complaints, messages
Realtime Engine	Socket.io	Live chat between users and agents
Video API	WebRTC	Enables future integration of real-time video support
Authentication	JWT	Secure login and role-based access
Dev Tools	Git, VS Code, Postman	Version control and testing tools

🔷 3.5 Architecture Overview

ResolveNow follows a **client-server architecture** with clearly separated layers for user interface, business logic, and data storage.

□ Key Architectural Highlights

- **Frontend (React.js):** Handles user interaction, form submission, complaint status updates, and role-based dashboards (User, Agent, Admin).
- **Backend (Express.js + Node.js):** Receives API calls, performs business logic, authenticates users, and manages complaint routing.
- **Database (MongoDB):** Stores structured data on users, complaints, assignments, messages, and agent skills.
- **Communication Layer:** Uses Socket.io for real-time chat and WebRTC for planned video calling.
- **Security Layer:** Implements JWT-based auth, input validation, and access control middleware.

📌 **Note:** The architecture also allows for future microservice decomposition if scale demands grow.

🔷 3.6 Data Flow Design

The ResolveNow system is designed around a **real-time, event-driven data flow** model that ensures complaints are submitted, processed, and routed efficiently. The flow integrates frontend user actions, backend logic, database transactions, and live communication.

□ High-Level Data Flow Steps

1. Complaint Submission (User Side)

- **Trigger:** A user fills out and submits a complaint via the React-based UI.
- **Data Captured:** name, email, category, description, attachments (if any), location, urgency, etc.
- **Frontend Action:** Axios sends the form data as a POST request to the /api/complaints endpoint.

2. Complaint Creation (Backend API)

- **Server Processing:**
    - Validates input data.
    - Stores complaint in MongoDB.
    - Triggers assignment engine.
- **Database Write:** Complaint is saved in the complaints collection with a default status of "new".

3. Intelligent Assignment Logic

- **Trigger Point:** On complaint creation.
- **Assignment Engine Workflow:**
    - Reads category, keywords, location, and user role.
    - Checks available agents with matching skills.
    - Evaluates current agent workloads.
    - Assigns complaint to the best-fit agent.
- **Outcome:** Updates the complaint document with assignedTo and assignedGroup.

4. Notification & Updates

- **Real-Time Update:** Assigned agent and user are notified via:
    - Email
    - In-app notification (toast/message)
    - Optional SMS (if configured)
- **Socket.io Trigger:** If the agent is online, a real-time notification appears on their dashboard.

5. Agent Interaction

- Agent opens assigned complaints and initiates a chat.
- The conversation is stored in the messages collection using references like complaintId and userId.

6. Complaint Resolution

- After the issue is resolved, the agent changes the complaint status to "resolved" or "closed".
- Resolution details and timestamps are logged.
- The system updates the complaint record in the database and triggers final notifications.

📊 Involved Entities (Collections/Tables)

Entity	Purpose
users	Stores user profiles and roles
complaints	Main collection for complaint records
assignments	Optional: Logs history of assignments
messages	Stores chat history between user and agent
agents	Contains agent metadata including skills
skills	Maps agents to skill tags
notifications	Manages real-time and scheduled alerts

🔁 Summary Flowchart

[User Submits Complaint]

↓

[Backend Validates & Saves]

↓

[Assignment Engine Activated]

↓

[Agent Selected Based on Skill + Load]

↓

[Complaint Updated with Assignment Info]

↓

[User & Agent Notified]

↓

[Agent Resolves Complaint]

↓

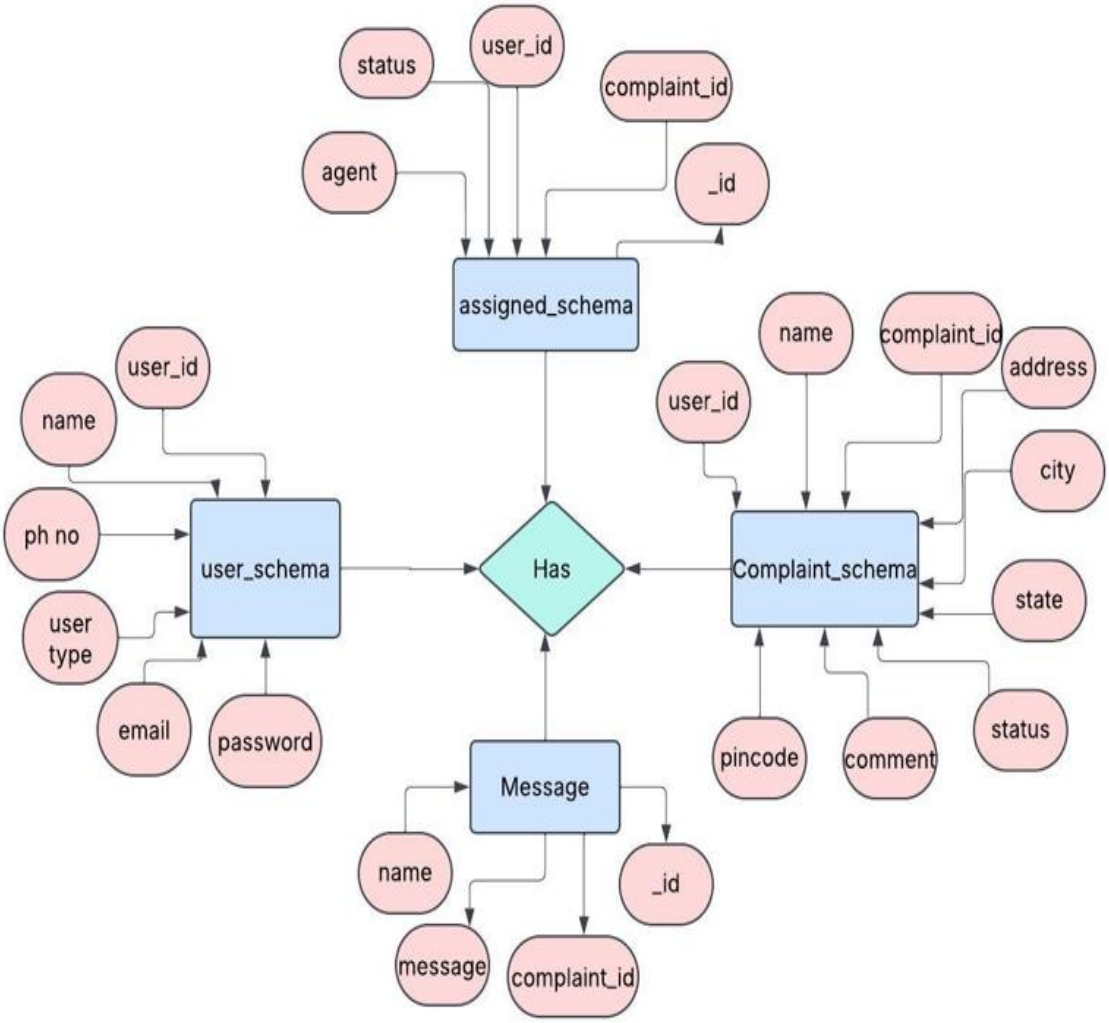
[Status Updated → Resolved/Closed]

↓

[Final Notification Sent]

This data flow ensures **speed, traceability, and automation** at every stage of the complaint lifecycle, improving both customer experience and operational efficiency.

ER Diagram :



4. PROJECT DESIGN

4.1 Problem-Solution Fit

The core design of ResolveNow directly addresses the key inefficiencies found in traditional complaint handling systems. Below is a breakdown of how each problem is solved through features in the platform.

Identified Problem	ResolveNow Solution	How It Helps
Manual complaint assignment	Automated, rule-based and skill-based routing	Reduces human error and speeds up response time
Misassignment & re-routing	Assignment logic uses category, keywords, and skills	Ensures right agent is chosen initially

Identified Problem	ResolveNow Solution	How It Helps
Agent overload & idle time	Real-time workload tracking with round-robin logic	Ensures fair complaint distribution
Lack of skill-based routing	Agent profiles include skill sets; matched with complaints	Improves first-contact resolution rate
Low visibility into operations	Dashboards for admins show workload, assignment time, and trends	Empowers proactive management
No scalability with increasing tickets	Modular architecture and automated routing	Supports growth without adding staff

4.2 Proposed Solution

The ResolveNow platform introduces a **multi-layered, automated complaint routing system** that includes:

🛠 **Core Components:**

- **Rule-Based Assignment:** Uses complaint categories and urgency levels to trigger routing logic.
- **Skill Mapping:** Complaints are matched with agents possessing the required technical or functional expertise.
- **Load-Balancing:** Prevents agent overload through real-time tracking of active complaints.
- **Auto-Reassignment:** Escalates unresolved tickets to more senior agents or alternate teams after timeout.
- **Manual Override:** Admins and managers can override automatic assignments when needed.
- **Real-Time Notifications:** Users and agents are immediately notified on assignment and updates.
- **Dashboards & Logs:** All assignment activity is logged for audit, and metrics are visualized for monitoring performance.

□ Optional Enhancements (Future Scope):

- **Predictive Assignment (AI-based):** Learn from past complaint patterns to suggest the best agent/team.

- **Media Upload Support:** Allow users to submit images or voice notes for richer complaint descriptions.
- **Voice Chat Integration:** For escalated complaints, enable voice chat with agents.

🔷 4.3 Solution Architecture

The ResolveNow platform is built with a **modular, layered architecture** that ensures scalability, maintainability, and performance. It integrates **frontend technologies**, **backend logic**, and **real-time communication** into a seamless complaint management system.

🧱 Architectural Layers

1. Presentation Layer

- **Components:** React.js, Material UI, Bootstrap
- **Users:** General users (complainants), agents, and admins
- **Functions:**
    - Complaint submission
    - Real-time chat
    - Dashboard access
    - Assignment visibility (who is handling the complaint)
    - Admin views for tracking and analysis

2. Application Logic Layer

- **Technology:** Node.js with Express.js
- **Functions:**
    - Handles REST API logic
    - JWT-based user authentication and role verification
    - Complaint lifecycle management (submit, assign, update, close)
    - Agent skill verification and workload monitoring
    - Manual override and admin escalations

3. Data Layer

- **Database:** MongoDB
- **Collections Used:**

- users – Stores user info and roles
- complaints – All complaint data
- messages – Chat history
- assignments – Agent-complaint mappings
- skills – Agent skill sets
- notifications – Stores and sends alerts/updates

- **Design:** Document-oriented with relationships using IDs (userId, agentId, complaintId)

4. Real-Time Communication Layer

- **Technology:** Socket.io
- **Functions:**
  - Enables live chat between user and agent
  - Provides real-time status updates and alerts
  - Powers future features like live typing indicators or video support

5. Security Layer

- **Components:**
  - JWT for secure user sessions
  - Role-based access control (User, Agent, Admin)
  - API validation middleware
  - Input sanitization to prevent XSS and injection attacks

📡 System Workflow (Simplified Overview)

1. User logs in and submits a complaint.
2. Complaint is saved to MongoDB.
3. Assignment logic checks category, keywords, and agent skills.
4. Complaint is routed to a matching agent.
5. User and agent are both notified.
6. Agent resolves the complaint and updates status.
7. Logs and dashboards are updated in real-time.

4.4 SETUP INSTRUCTIONS

PRE-REQUISITES

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, and 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, the status of the complaints, and user interfaces for the admin dashboard. To make better UI we have also used some libraries like Material UI and Bootstrap.

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 the 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/satya-amarthi04/ResolveNow-Platform-For-Online-Complaints/tree/main

INSTALLATION

• Navigate into the cloned repository directory:

```
cd ComplaintCare-MERN
```

• Install the required dependencies by running the following commands:

```
cd frontend
npm install
cd ../backend
npm install
```

Start the Development Server:

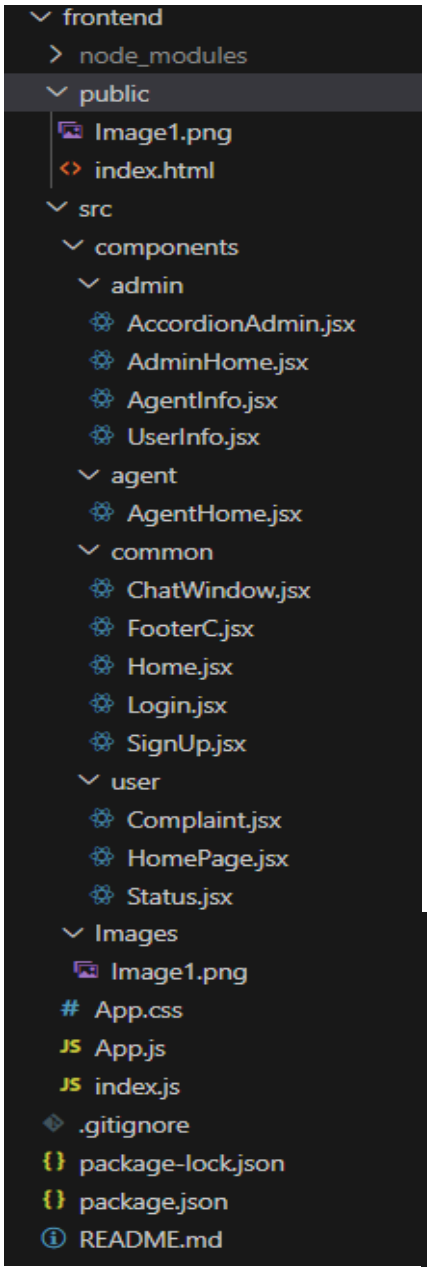
• To start the development server, execute the following command:

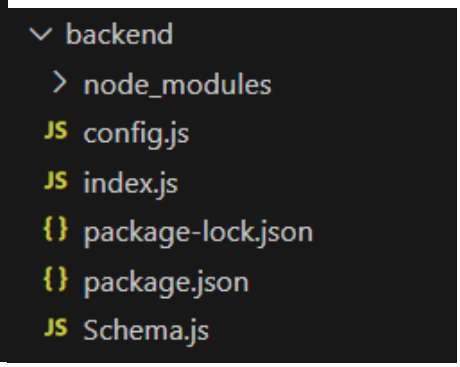
```
npm start
```

• 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

4.5.FOLDER STRUCTURE





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

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

4.6.RUNNING THE APPLICATION

To run the Online Complaint Registration and Management System, follow the instructions below to set up and launch both the frontend and backend servers locally.

Step 1: Start the Backend Server

1. Open a terminal and navigate to the backend directory:
   cd server

2. Install backend dependencies:
   npm install

3. Set up environment variables by creating a .env file in the root of the server folder.
Example:
```
PORT=5000
MONGO_URI=mongodb://localhost:27017/complaintDB
JWT_SECRET=your_secret_key
```

4. Start the backend server:
   npm start

The backend server should now be running at: http://localhost:5000

Step 2: Start the Frontend (React Application)

1. Open a new terminal window and navigate to the frontend directory:
   cd client

2. Install frontend dependencies:
   npm install

3. (Optional) Add a .env file in the client folder if you're configuring environment variables:
```
REACT_APP_API_URL=http://localhost:5000/api
```

4. Start the frontend application:
   npm start

The frontend will now be available at: http://localhost:3000

Final Output

Once both servers are running:

- The frontend application will interact with the backend API through RESTful services using Axios.
- You can access the full functionality — including user registration, complaint tracking, agent interaction, and admin management — from the UI.

🔷 5. PROJECT PLANNING & SCHEDULING

To ensure the timely and efficient delivery of the ResolveNow platform, the development process followed an **Agile methodology**, allowing for iterative development, early feedback, and flexibility to adapt throughout the project lifecycle.

5.1 Project Planning

🎯 Project Objectives

- Reduce average complaint assignment time by 50%
- Improve first-contact resolution accuracy to 90%
- Eliminate manual routing bottlenecks
- Ensure equitable workload distribution across support agents
- Enhance overall user satisfaction through faster resolution and communication

📦 Project Scope

In Scope	Out of Scope (for initial phase)
Complaint submission, routing, and tracking	Integration with external CRM systems
Skill-based and rule-based assignment logic	AI-based predictive complaint assignment
Role-based dashboards (User/Agent/Admin)	Video calling and voice chat features
Chat functionality between user and agent	Knowledge base or self-help integration
JWT-based secure login & authentication	Mobile app (planned in future phase)

👥 Key Roles and Responsibilities

Role	Responsibility
Project Manager	Coordinate tasks, timelines, and resources
Frontend Developer	Build user dashboards, complaint forms, and chat UI
Backend Developer	Implement API endpoints, assignment logic, and database models
Database Engineer	Design and optimize MongoDB schema

Role	Responsibility
QA/Test Engineer	Perform manual and automated testing
UI/UX Designer	Create clean, responsive, and intuitive interfaces
Team Leads (Admin)	Provide real-world routing rules and test feedback

5.2 Project Scheduling

The ResolveNow system was developed in phases to ensure steady progress and early validation.

Phase	Duration	Deliverables
Phase 1: Requirements & Planning	Week 1	Finalize features, user roles, system flow
Phase 2: UI/UX Prototyping	Week 2	Wireframes, dashboard mockups, routing visuals
Phase 3: Frontend Development	Weeks 3–4	Complaint form, login/register, chat module
Phase 4: Backend & DB Setup	Weeks 5–6	REST APIs, MongoDB schema, JWT auth, routing logic
Phase 5: Integration	Week 7	Connect frontend with backend via Axios and sockets
Phase 6: Testing	Week 8	Manual + API testing using Postman & MongoDB Compass
Phase 7: Final Review & Deployment	Week 8	Launch locally, documentation, video demo recording

🔷 6. TESTING

Testing played a vital role in ensuring that ResolveNow delivers a smooth, bug-free experience across all user roles — **users, agents, and admins**. A combination of **manual**, **automated**, and **API-level testing** was used.

6.1 Manual Testing

Manual testing was carried out to simulate real-world usage scenarios and validate that all system features behave as expected.

🔧 Tested Functionalities

- User registration and login
- Complaint form validation and submission
- Complaint status updates
- Skill-based agent assignment
- Real-time chat between user and agent
- Role-based dashboard rendering
- Admin-level controls and overrides

□ Tools Used

Tool	Purpose
Postman	Testing REST API endpoints (e.g., /api/complaints, /api/login)
MongoDB Compass	Checking backend data integrity (e.g., complaints, user roles)
Browser DevTools	Verifying responsiveness and UI behavior
Socket Debugger	Testing real-time message delivery using WebSocket events

6.2 Sample Manual Test Cases

Test Case ID	Description	Expected Result	Status
TC-001	User registers with valid data	User created, redirected to login	✅ Pass
TC-002	User submits a complaint	Complaint saved, assigned to agent	✅ Pass

Test Case ID	Description	Expected Result	Status
TC-003	Agent logs in and views assigned complaints	Dashboard displays only assigned cases	✅ Pass
TC-004	Chat between user and agent	Real-time message exchange	✅ Pass
TC-005	Admin reassigns a complaint manually	Complaint updates in database and UI	✅ Pass
TC-006	Complaint submitted without category	Validation error shown	✅ Pass

6.3 API Testing (Postman)

APIs were tested with various payloads to ensure data correctness, token handling, and response formats.

✅ **Examples:**

- **POST /api/register**: Checked with missing fields, invalid email, and valid data
- **POST /api/login**: Validated JWT issuance and error handling
- **GET /api/complaints/user/:id**: Ensured only relevant complaints return for each user
- **POST /api/messages**: Tested real-time logging of user-agent conversations

6.4 Key Findings & Fixes

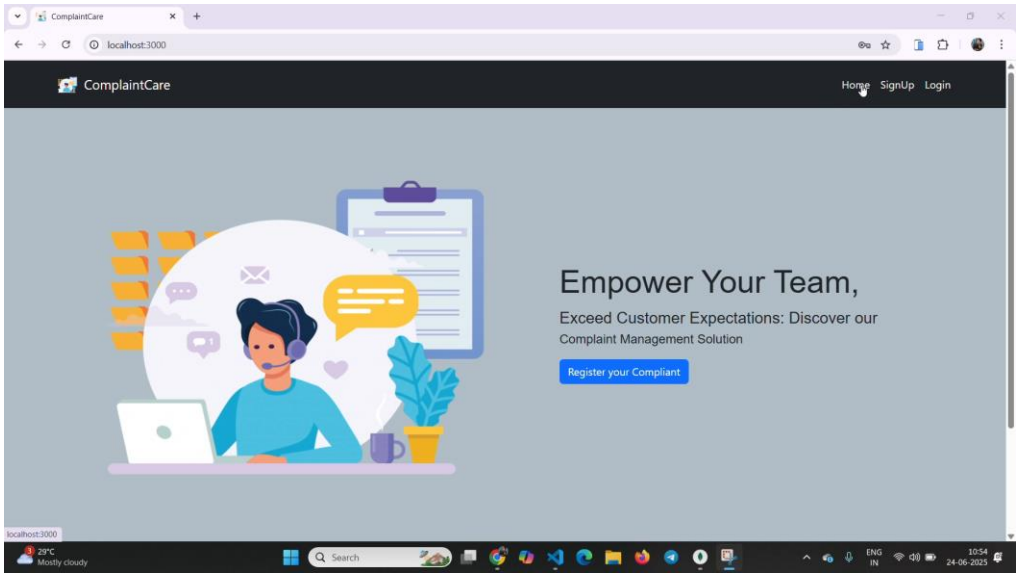
Issue Found	Resolution
Some complaints not assigned due to missing category	Added default fallback routing group
Messages not appearing in real-time occasionally	Implemented socket reconnect logic
Duplicate complaint submission on slow networks	Added debounce + submission lock

🔷 7. SCREENSHOTS OR DEMO

To visualize the functionality and user experience of ResolveNow, key interface screens were captured from the live, running application. These screenshots represent the core features available to users, agents, and admins.

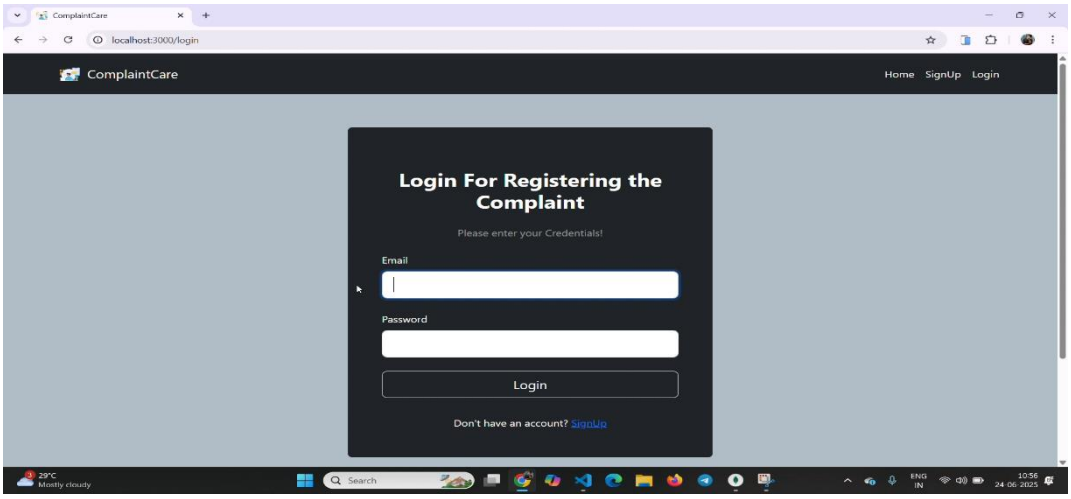
📄 7.1 Landing Page

- The homepage introduces ResolveNow and its mission to simplify complaint resolution.
- Provides options to **Register**, **Login**, or **Learn More**.



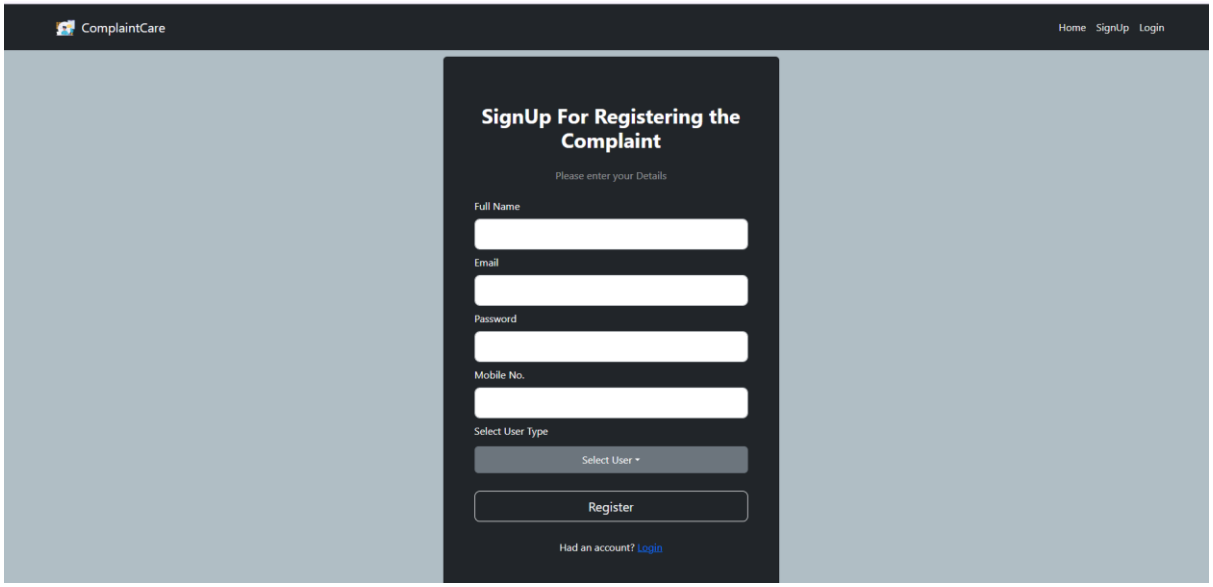
🔐 7.2 Login Page

- Users, Agents, and Admins log in using their credentials.
- JWT is used for secure authentication.
- Invalid login attempts trigger client-side validation messages.



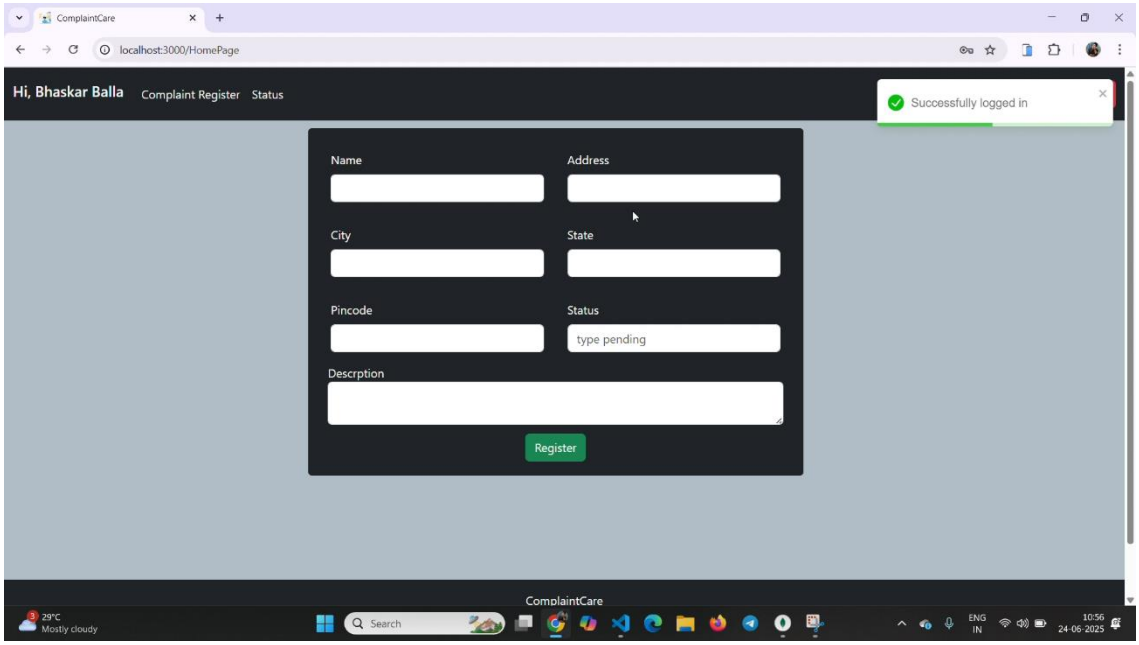
7.3 Registration Page

- New users can register by entering name, email, password, and selecting a role.
- Admins are created manually in the backend or seeded during development.



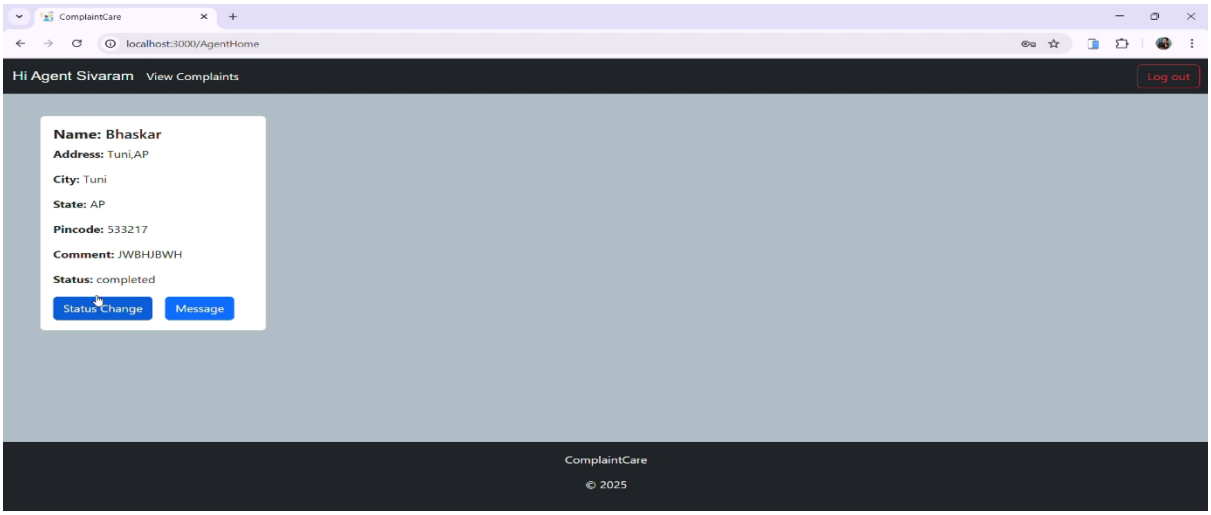
7.4 Complaint Submission Form

- Registered users can:
  - Enter complaint details
  - Select a category
  - Attach optional files or screenshots
- Complaints are routed automatically after submission.



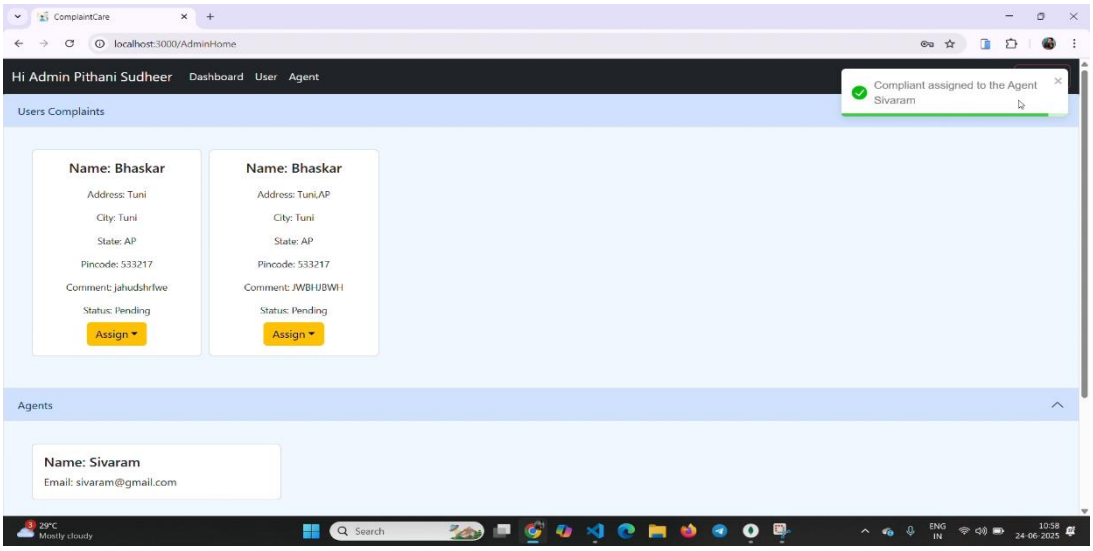
🖥️ 7.5 Agent Dashboard

- Displays:
    - Assigned complaints
    - Live chat window
    - Complaint status update options (e.g., In Progress, Resolved)
- Agent can communicate directly with users via chat.



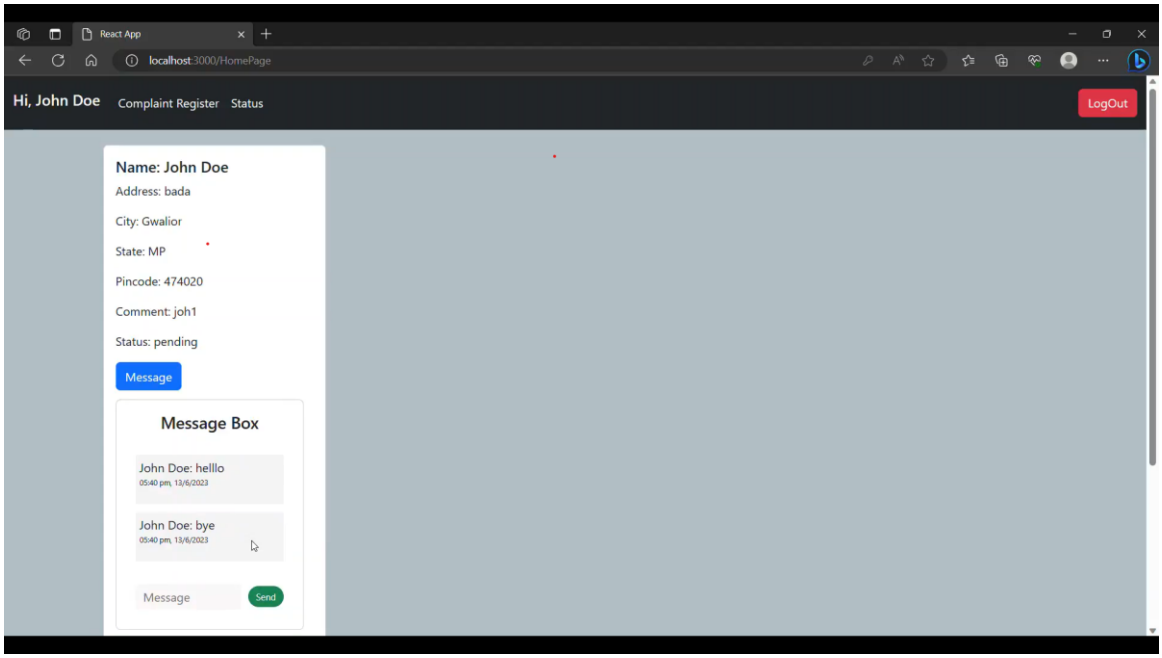
🛠️ 7.6 Admin Dashboard

- Admins can:
    - View all complaints in the system
    - Manually assign or reassign complaints
    - View agent workloads
    - Track resolution timelines and complaint status



💬 7.7 Chat Interface (User ↔ Agent)

- Real-time messaging powered by **Socket.io**
- Automatically links conversations to complaint ID
- Agent and user can exchange messages live, reducing the need for email follow-up



📹 7.8 Project Demo (Video)

- A full walkthrough demo video is available:
    - **Link:** Click to Watch Demo

These visuals demonstrate how ResolveNow bridges the gap between users and support teams through a well-structured interface, automated routing, and real-time interaction.

🔷 8. KNOWN ISSUES

While the ResolveNow platform successfully delivers its core functionalities — complaint submission, smart assignment, and user-agent communication — several **known issues and limitations** were identified during development and testing. These will be addressed in upcoming updates.

Current Limitations

- **No File Type Validation**
- Users can upload any type of file (images, PDFs, executables, etc.) without restriction.
- Risk: This may introduce security or storage concerns.

- Planned Fix: Add file type and size validation logic on both frontend and backend.

2. **Lack of Analytics**
    - The admin dashboard currently shows complaint lists and status, but lacks:
        - Visual summaries
        - Trends over time
        - Agent performance metrics
    - Planned Fix: Integrate charts and metrics using Chart.js or D3.js in the admin panel.
3. **Text-Only Chat**
    - The messaging system supports only plain text.
    - Limitation: Users cannot share screenshots or voice messages, which can delay issue clarification.
    - Planned Fix: Allow media attachments in chat (images, PDFs, voice notes).
4. **No Email Alerts for New Messages**
    - Real-time chat works while online, but users don't receive email alerts when they're offline and receive a new message.
    - Planned Fix: Implement notification service with optional email alerts.
5. **No Role-Based Analytics**
    - The system lacks role-specific visual dashboards (e.g., average response time per agent, resolved tickets per week).
    - Planned Fix: Extend dashboard filters and reports per role and time period.
6. **Static Skill Assignment**
    - Agent skills are currently assigned manually during setup.
    - Planned Fix: Create an admin UI to add/edit agent skills dynamically.

These issues do not affect the **core functionality** of the platform but are important areas for improvement to enhance usability, security, and performance in future releases.

🔷 9. FUTURE ENHANCEMENTS

To make ResolveNow a more powerful, intelligent, and enterprise-ready platform, several advanced features are planned for future development. These enhancements aim to improve usability, automation, scalability, and overall customer satisfaction.

🚀 Planned Features

1. AI-Based Complaint Routing

- Use machine learning to predict the best agent/team for a new complaint based on:
    - Complaint history
    - Language patterns (NLP)
    - Resolution success rate of agents
- Benefit: Increases routing accuracy over time without manually updating rules.

2. Mobile App (Android/iOS)

- Build dedicated mobile applications for both users and agents to:
    - Submit or respond to complaints on-the-go
    - Receive push notifications
    - Use voice-to-text complaint input
- Benefit: Improves accessibility and convenience.

3. Multilingual Support

- Add support for regional languages (Hindi, Tamil, etc.) and auto-translation in chat.
- Benefit: Makes the platform more inclusive and user-friendly.

4. Chat Attachments

- Enable users and agents to send:
    - Screenshots
    - Audio clips
    - PDF reports
- Benefit: Speeds up issue resolution with richer context.

5. Service-Level Agreement (SLA) Tracking

- Allow admins to set time-based SLAs per complaint category.

- Auto-highlight or escalate complaints that violate SLA deadlines.
- Benefit: Helps teams meet response/resolution benchmarks.

6. Self-Help Knowledge Base

- Integrate a searchable FAQ or help center.
- Show suggested articles while typing a complaint.
- Benefit: Reduces complaint volume and empowers users to resolve minor issues themselves.

7. Sentiment Analysis in Chat

- Monitor real-time sentiment during user-agent interactions using NLP.
- Trigger escalation or alerts for negative sentiment.
- Benefit: Enables proactive intervention to improve customer satisfaction.

These enhancements will position ResolveNow as a **smart, scalable, and AI-driven complaint resolution ecosystem**, ready for deployment in public, private, or enterprise environments.

🔷 10. CONCLUSION

The ResolveNow platform was developed to modernize and streamline the way customer complaints are managed, routed, and resolved. By integrating intelligent assignment mechanisms, real-time communication tools, and a user-friendly interface, the system addresses key pain points such as manual routing, misassignments, and resolution delays.

Through a structured approach—from problem identification and brainstorming to architectural planning and testing—the project delivers a robust foundation for a scalable and efficient complaint management system. The use of **rule-based logic**, **skill mapping**, and **workload balancing** ensures that complaints are handled by the most suitable agents, improving both productivity and customer satisfaction.

While the initial implementation focuses on core functionality, the project has been designed with future expansion in mind. Planned enhancements such as AI-based routing, multilingual support, and SLA tracking will further elevate the system's capability and reach.

In conclusion, ResolveNow successfully transforms traditional support operations into a smart, automated, and user-centered process—laying the groundwork for a next-generation complaint resolution ecosystem.