TEAM MEMBERS:

- k.MADHAVI
- INGULLURU PAVITHRA
- K.HARSHA PRAVEEN

Introduction

1

1.1 Project Overview: ResolveNow – Online Complaint Registration and Management System

Overview

This system is a full-featured online platform built to streamline the handling of customer complaints with speed, transparency, and efficiency. It empowers both users and organizations by offering a centralized, real-time environment for issue reporting, status tracking, and resolution.

Core Features

- **User Registration:** Enables individuals to create secure accounts to lodge complaints and manage them within a personal dashboard.
- **Complaint Submission:** Offers a user-friendly interface to input detailed issue descriptions, attach relevant files, and submit personal information like address and contact details.
- **Tracking and Notifications:** Users are kept informed about the progress of their complaints with timely updates through email or SMS. Status changes, assignment to agents, and resolution events are automatically communicated.
- **Live Interaction with Agents:** Abuilt-in messaging system supports seamless conversations between users and the agents assigned to their cases, ensuring quicker clarifications and personalized service.
- **Assignment & Routing:** Complaints are automatically dispatched to the appropriate personnel or departments using smart routing logic, optimizing response time and resource distribution.

- **Security & Confidentiality:** The system ensures the protection of sensitive data through encryption, secure user authentication, access control mechanisms, and adherence to data protection laws.

- **Feedback and Analytics:** After complaint resolution, users can provide feedback on the handling process. The system aggregates these responses and generates analytical insights—such as resolution times, agent performance, complaint frequency, and user satisfaction metrics—which help organizations improve service quality and internal decision-making.

Operational Advantages

- **For Users:** A transparent complaint process, improved response time, and the ability to stay informed throughout resolution stages.

- **For Organizations:** Better visibility into customer issues, data-driven decision-making, improved compliance, and streamlined workflows.

- **For Administrators:** Tools to monitor overall complaint flow, assign responsibilities, enforce policy adherence, and ensure customer satisfaction goals are being met. This system lays the foundation for a more responsive, connected, and trust-building experience between users and service providers.

1.2 Purpose

The purpose of this system is to provide a streamlined, secure, and user-friendly digital platform that allows individuals and organizations to register, track, and resolve complaints efficiently. By centralizing the complaint-handling process, the platform enhances service quality, reduces resolution times, and fosters transparency between users and service providers. It also empowers organizations to manage issues in compliance with industry standards and data protection regulations, ultimately leading to increased trust and customer satisfaction.Ultimately, ResolveNow is designed to transform every complaint into an opportunity—to improve service quality, boost stakeholder trust, and drive continuous improvement through technology-enabled responsiveness.

2 IDEATION PHASE

2.1 Problem Statement

In many organizations and service-basedindustries, the process of handling customer complaints is fragmented, inefficient, and lacks transparency. Users often struggle with unclear reporting channels, delayed responses, and little to no visibility into the status of their complaints. Meanwhile, organizations face challenges in complaint tracking, prioritization, assignment, and ensuring timely resolution, often leading to decreased customer satisfaction and reputational damage.

There is a growing need for a centralized, secure, and streamlined platform that enables users to easily register complaints, track their resolution in real-time, and interact directly with assigned agents. Simultaneously, organizations require a smart system that efficiently routes complaints, monitors agent performance, ensures data security, and supports compliance with regulatory frameworks.

The absence of such a system creates communication gaps, slows resolution time-lines, and undermines trust between users and service providers—necessitating the development of an online complaint registration and management platform

2.2 Empathy Map Canvas

An **Empathy Map** is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Empathy Map Explanation

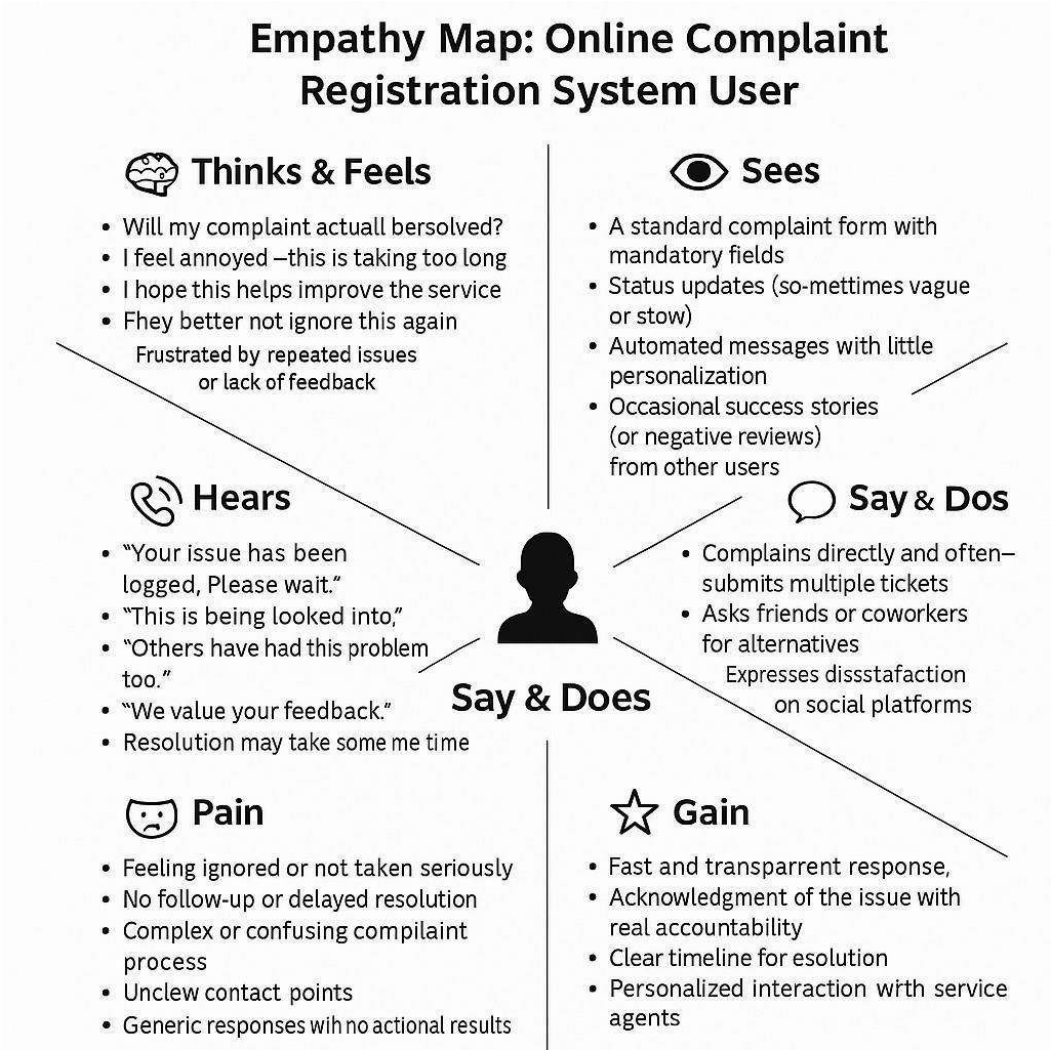


Fig. 1.1: Enter Caption

Thinks & Feels

This section captures the user's inner dialogue and emotions. It reveals their uncertainty about whether the complaint will be taken seriously, concerns about service quality, and emotional reactions like frustration and skepticism. These insights help identify emotional drivers that influence behavior.
subsection* **Hears** This reflects the messages the user receives from others or from the system—both automated andhumaninteractions. Phrases like "We'll look into it" or "Please wait" suggest users are exposed to generic, sometimes unhelpful feedback, which shapes their trust and expectations of the system.

Sees

This quadrant includes what the user visually encounters: structured forms, impersonal automated updates, and visible social proof (like reviews or status pages). These visuals affect their experience and perception of the system's reliability.

Says & Does

Here we capture external expressions and actions. Users may complain out loud, post negative feedback online, seek help from peers, or make repeated calls. These behaviors signal dissatisfaction and highlight the need for more responsive communication.

Pain

Pain points include delays, lack of updates, poor communication, and generic or unhelpful responses. These lead to users feeling ignored, overwhelmed, or disillusioned with the complaint process.

Gain

This quadrant focuses on the user's desired outcomes: quick resolution, transparency, personalized service, and visible improvements based on their feedback. Meeting these expectations builds trust and encourages repeat engagement.

Core Concepts Identified

- **Product Name**: ResolveNow
- **Purpose**: Digital platform for complaint registration and management
- **Primary Users**: General users/customers, service agents, administrators
- **Key Functionalities**:
  - Account creation & login
  - Complaint submission with attachments
  - Real-time tracking and notifications
  - Agent-user chat interaction
  - Admin workload management
  - Feedback loop

What Works Well

- **User-Centric Scenario**: "John's journey" provides strong narrative clarity—very helpful for product pitching and UI/UX design
- **Clear System Flow**: Registration → Submission → Assignment → Resolution → Feedback
- **Well-Chosen Tech Stack**:
  - Frontend: Bootstrap, Material UI, Axios
  - Backend: Express.js
  - Database: MongoDB
  - Real-time Communication: socket.io, WebRTC (though WebRTC may be better suited for video chat features)

Opportunities for Enhancement

1. **Clarify Scope**: Revise references to video conferencing unless video chat is an integrated feature.
2. **Role Clarity**: Define access levels for:
    - General users
    - Customer service agents
    - System administrators
3. **Advanced Features**:
    - NLP-based topic tagging
    - Predictive complaint assignment
    - Admin analytics dashboards
    - Sentiment analysis on feedback
    - Multilingual interface
4. **UI/UX Mockup Stage**:
    - Use user stories like “John” for screen designs
    - Ensure mobile responsiveness
5. **Data Compliance**:
    - Address GDPR and DPDP Act adherence
    - Implement audit logs and activity tracking

Possible Deliverables

- Empathy Maps

- Problem & Purpose Statements
- System Flow Diagrams
- ER Diagram for MongoDB Collections
- Tech Stack Justification
- UI Wireframes
- Test Cases & Acceptance Criteria
- Deployment Plan (CI/CD, hosting options)

documentclassarticle graphicx caption float

3 REQUIREMENT ANALYSIS

3.1 Customer Journey map

The ResolveNOW platform streamlines the complaint management process for customers, offering an efficient path from problem identification to resolution. Below is a visual representation of the end-to-end customer journey, highlighting each stage from complaint registration to final feedback. After the complaint is resolved and the user submits feedback, the system acknowledges their input and updates the internal metrics. The feedback is used by administrators to monitor agent performance and service quality trends. If necesssary, the platform can initiate follow-up actions such as a quality assurance check or a satisfaction survey. For unresolved or escalated cases, admins may intervene directly or reassign the case to a more senior agent. Over time, patterns in complaints can help the organization improve products or services proactively. The platform may also use this data to update FAQs or implement policy changes. For returning users, the system can offer a streamlined experience by pre-filling known data and allowing quick complaint duplication for recurring issues. Notifications and updates maintain transparency and reduce the need for repeated user inquiries. Ultimately, the system fosters trust by ensuring that users feel heard, informed, and supported throughout the process. This continuous feedback loop strengthens both user loyalty and organizational accountability.

Fig. 1.2: customer journey map-ResolveNow

1.1

3.2 solution requirement

1. User Authentication Security:

*user Registration/Login: Secure email-based sign-up and login with password .

*Email Verification: One-time verification link sent post-registration.

*Access Control: Role-based access for Admin, Agent, and User.

*Session Management: JWT tokens for secure sessions.

*Encryption Compliance: Secure transmission (HTTPS), data encryption at rest and in transit, compliance with relevant data protection laws (e.g., DPDP Act in India).

2.Complaint Lifecycle Management:

*Complaint Submission: A form capturing title, category, issue description, location, attachments.

*Unique Complaint ID: Auto-generated for each new complaint.

*Complaint Routing: Intelligent routing based on department or agent specialization.

*Status Tracking: Live updates with statuses like Pending, Assigned, In Progress,

Resolved, Closed.
*User-Agent Communication: Real-time chat/messaging interface with file sharing capability.

3. User Interface Dashboards:

*User Dashboard: Displays submitted complaints, status, and response history.
*Agent Dashboard: Lists assigned complaints, priority filters, response panel.
*Admin Dashboard: Overview of all complaints, user management, analytics, routing control.

4. Notifications Alerts:

*Email SMS Alerts: Triggered at complaint updates, agent assignment, resolution notice.
*Reminders Escalation: Alert users/admins if a complaint remains unresolved past SLA.

5.Data Insights Feedback

*User Feedback: Prompt users to rate their complaint resolution experience.
*Analytics Reporting: Generate charts or reports on resolution time, common complaint types, agent performance.

6. Integration Real-Time Features:

*Socket.IO/WebRTC: Enable live messaging and optional video support.
* RESTful API: For frontend-backend communication via Axios.
*Webhooks: For external system notifications or CRM integration (optional).

7. Administration Tools:

*User Management: Create, suspend, or assign roles to users.
*Agent Load Balancing: Automatically assign tickets based on workload or availability.
*Log Audit Trail: Track all user and system activity for transparency and debugging

3.3 Data flow Diagram

1.1.1

Fig. 1.3: Data flow Diagram

1.1.2

The data flow in ResolveNow follows a structured path from user interaction to backend processing and storage, ensuring seamless complaint management and resolution. Here's a step-by-step breakdown:

1. User Registration Authentication

*Input:User submits registration form (name, email, password).

*Process: - Frontend sends data via Axios to backend API ('/api/register').

- Backend (Express.js) validates input and hashes the password.

- User data is stored in MongoDB.

- A verification email is sent to the user

. **Output:User receives a confirmation email and can log in.

2. Complaint Submission

*Input:Logged-in user fills out complaint form (issue description, attachments, contact info).

*Process:

- Frontend sends complaint data to backend API ('/api/complaints/create').

- Backend stores complaint in MongoDB with status "Pending."

- System triggers notification to admin for assignment.
*Output: User sees confirmation and complaint ID; complaint appears in “My Complaints.”

3. Complaint Assignment Routing
*Input: Admin dashboard displays unassigned complaints.
*Process:
- Admin assigns complaint manually or system uses routing logic (based on agent workload/expertise).
- Complaint status updated to “Assigned” in MongoDB.
- Assigned agent receives notification.
*Output:Agent sees new complaint in their dashboard.

4. Agent Interaction Communication
*Input:Agent initiates chat with user via built-in messaging.
*Process:
- Messages exchanged using Socket.IO for real-time communication.
- WebRTC API may be used for optional video support.
- Chat history stored in MongoDB.
*Output: User and agent communicate to clarify and resolve the issue.

5. Notifications Updates
*Input: Any status change (e.g., “In Progress,” “Resolved”).
*Process:
- Backend triggers notification service (email/SMS).
- User receives updates via preferred channel.
*Output:User stays informed about complaint progress.

6. Resolution Feedback
*Input: Agent marks complaint as resolved and provides resolution details.
*Process:
- Backend updates complaint status in MongoDB.
- User receives resolution notification.
- User submits feedback via frontend form.
*Output: Feedback stored; admin can review for quality assurance.

7. Admin Monitoring Analytics
*Input: Admin accesses dashboard.
*Process:
- Backend aggregates data (complaint volume, resolution time, agent performance).
- Visualized using charts/tables on frontend.
*Output:Admin gains insights for operational improvements.

—

8. Security Compliance
- User authentication via JWT tokens.
- Data encrypted in transit (HTTPS) and at rest (MongoDB encryption).
- Role-based access control (User, Agent, Admin).
- GDPR-compliant data handling and retention policies.

3.4 Technology Stack

The frontend of ResolveNow is built using React.js, which provides a responsive, component-based user interface. To ensure a polished and accessible design, it incorporates Bootstrap and Material-UI, offering a blend of utility-first and pre-built styling components. For API communication between the frontend and backend, Axios is used to handle HTTP requests seamlessly. Additionally, Socket.IO is integrated on the client side to support real-time messaging features between users and agents. On the backend, the platform leverages Node.js and the Express.js framework to manage server-side logic and route handling. Real-time interactions are powered by Socket.IO on the server side as well, enabling instant updates and two-way communication. Optional video conferencing capabilities are facilitated by the WebRTC API. For user authentication and secure session handling, the system employs JWT (JSON Web Tokens). Data is stored in MongoDB, a flexible NoSQL database that is well-suited for managing dynamic complaint records, user profiles, and chat histories. Mongoose, an ODM library, is used to provide schema-based modeling and streamline database interactions. To maintain robust security, ResolveNow uses HTTPS for secure data transmission, and Bcrypt.js to hash user passwords. Access control is enforced through role-based permissions that distinguish users, agents, and administrators. Notifications are delivered via Nodemailer for email updates, and optionally through Twilio or a similar SMS gateway for mobile alerts. Finally, for deployment and operations, the stack supports Docker for containerization, and NGINX for load balancing and reverse proxy setup. Version control is handled through Git

and GitHub, and continuous integration and deployment pipelines can be managed using tools like GitHub Actions or Jenkins. Cloud hosting environments such as AWS, Heroku, or DigitalOcean can be used to scale and manage infrastructure efficiently

4. PROJECT DESIGN

4.1 Problem Solution Fit

Problem-Fit Solution for ResolveNow

Problem 1:
Disorganized and Delayed Complaint Handling Many organizations struggle with manual or inconsistent complaint tracking, leading to unresolved issues and poor customer satisfaction.

ResolveNow's Solution:
A centralized, web-based system that allows users to submit complaints and track their resolution in real time. Automated workflows and intelligent routing ensure that issues reach the right personnel quickly, reducingresolution time and eliminating miscommunication.

Problem 2:
Lack of User Engagement and Transparency** Customers often feel disconnected from the complaint process and are left in the dark about the status of their cases.

ResolveNow's Solution:
User dashboards with live tracking, status updates, and automated notifications via email/SMS ensure users are constantly informed. Direct interaction with assigned agents builds trust and improves satisfaction.

Problem 3:
Inefficient Resource Allocation** Manually assigning complaints often leads to imbalances in agent workloads and delayed responses.

ResolveNow'sSolution:
Intelligent routing algorithms and admin-level complaint distribution allow dynamic, skill-based allocation—ensuring faster response times and more efficient operations.

Problem 4:
Data Insecurity and Compliance Risks** Storing sensitive user and complaint data without proper security can lead to regulatory violations and breaches of trust.

ResolveNow's Solution:
The system is designed with robust security measures including JWT-based authentication, encrypted transmission (HTTPS), access control policies, and compliance with data protection standards (e.g., GDPR).

Problem 5:
Lack of Feedback for Continuous Improvement** Organizations often miss opportunities to improve service because there's no structured way to capture user experiences.

ResolveNow's Solution:
Post-resolution feedback forms enable users to rate their experience and highlight areas for improvement. Admins can analyze these insights to refine processes and train personnel more effectively.

Problem 6:
Poor Integration and Scalability in Legacy Systems** Many traditional complaint systems don't support modern tools or scale well with growing user demands.

ResolveNow's Solution:
Built using scalable technologies like MongoDB and Express.js, and equipped with real-time communication via Socket.IO and WebRTC, the platform is modular, maintainable, and cloud-ready—ensuring long-term adaptability.

4.2 Proposed solution

The proposed solution to current challenges in complaint handling is ResolveNow—a streamlined, secure, and intelligent online platform designed for efficient complaint registration and resolution. This solution enables users to register and log in seamlessly, submit detailed complaints, and track their resolution status in real time. Complaints are intelligently routed to appropriate departments or agents based on urgency, type, and agent availability, ensuring swift resolution. Through an intuitive dashboard, users are kept informed via email or SMS notifications. Built-in real-time messaging and optional video communication tools create a responsive channel for user-agent interaction. Admins oversee the entire complaint pipeline, manage assignments, and analyze performance metrics. The platform ensures data protection through encrypted storage, secure authentication, and compliance with industry

standards. Under the hood, ResolveNow uses a robust tech stack that includes React.js for the user interface, Express.js and Node.js for backend logic, and MongoDB for scalable data storage. Real-time communication is supported by Socket.IO and WebRTC, and notifications are handled by services like Nodemailer and Twilio. This comprehensive solution modernizes how complaints are tracked, managed, and resolved, leading to greater transparency, accountability, and customer satisfaction You could improve scalability and UX further by building a Progressive Web App (PWA) with multilingual support and using Redux or Zustand for state management. Ensure compliance with data regulations by incorporating GDPR features such as user consent flows, encryption, and retention policies. If video interaction is part of the resolution process, WebRTC integration with TURN/STUN servers will support real-time communication while safeguarding privacy. Lastly, incorporating sentiment analysis or a simple chatbot can streamline complaint intake, making your system even more intuitive and responsive. Let me know if you'd like this outlined in a diagram or LaTeX format as well!

4.3 solution architecture

The image represents a high-level conceptual diagram of a complaint management system's technical architecture. It follows a client-server model where the frontend interacts with the backend through RESTful APIs. On the frontend, technologies like React are paired with UI libraries such as Material UI and Bootstrap to deliver a responsive and consistent user experience. Users can register, log in, submit complaints, and track their status from this interface. The backend, built with Express.js, manages server-side logic, routes requests, and handles operations like complaint assignment and status updates. MongoDB serves as the database for storing user details, complaint records, and communication logs. Additionally, Socket.IO and WebRTC are integrated for real-time features such as live chat or video conferencing between users and agents. The system emphasizes modularity, data security, role-based access, and scalability to support seamless complaint resolution and effective administrative oversight.

1.1.3

5. PROJECT PLANNING SCHEDULING

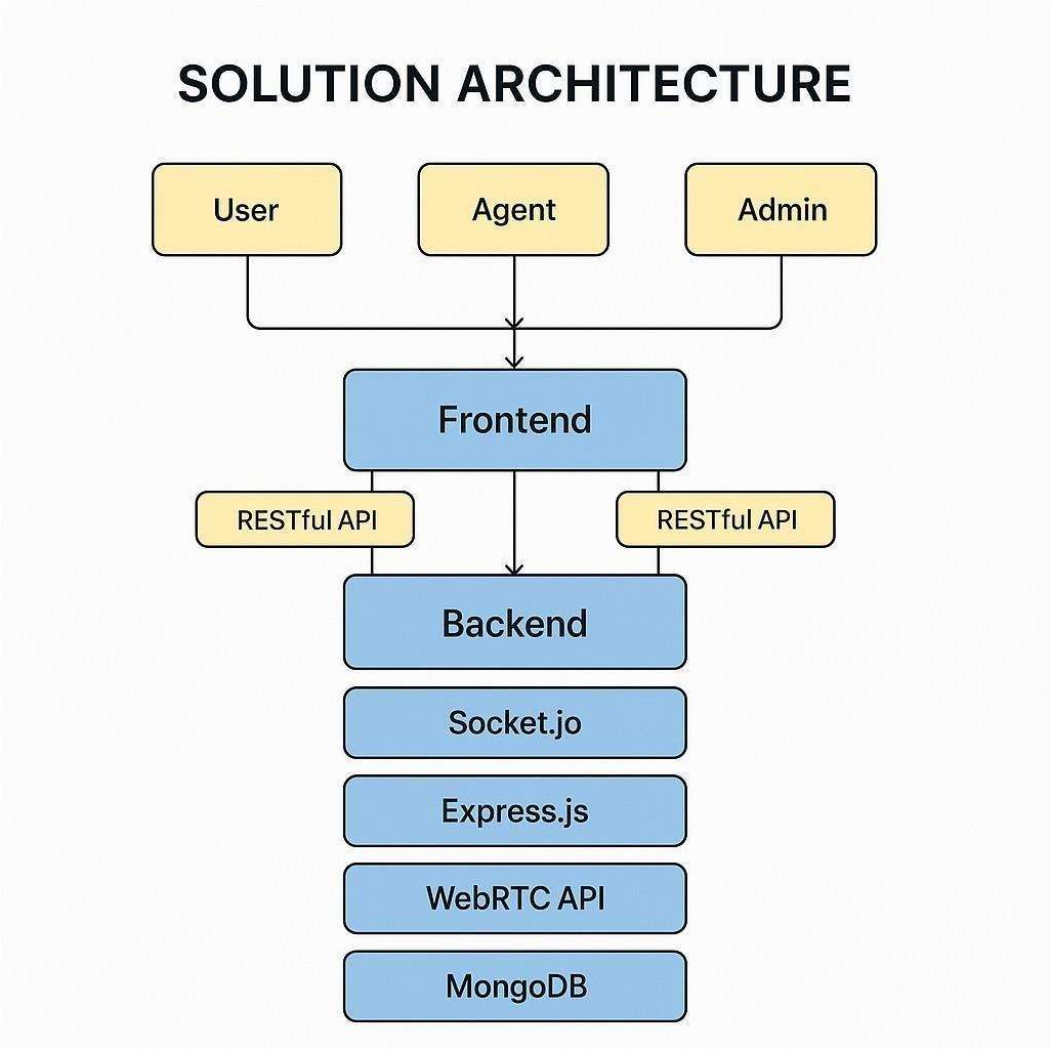


Fig. 1.4: proposed solution

5. Project Planning Scheduling for ResolveNow

Phase 1:week1

Requirements Gathering Planning

Objective: Finalize scope, identify stakeholders, define key features

- Analyze target users and roles (Admin, Agent, User)
- Draft user stories, workflows, and database schema
- Create empathy maps and complaint resolution journey maps
- Tools: Miro/Figma for journey maps, Draw.io for system architecture

Phase 2:week2

UI/UX Design

Objective: Develop responsive, intuitive layouts for each role

- Create wireframes and mockups using Figma or Adobe XD
- Design components with Bootstrap Material UI
- Define UI states for registration, complaint submission, chat, tracking, etc.

Phase 3:weeek 3-4

Frontend Development

Objective: Build the user interface with key interactions

- Technologies: HTML, CSS, JavaScript, React.js
- Implement user registration/login, complaint form, dashboard, and agent chat
- Integrate React Router, axios for API calls, and state management (e.g., Redux)
- Begin implementing form validation and error handling

Phase 4:week 5-6

Backend Development

Objective: Set up server-side logic and API services

- Technologies: Node.js, Express.js
- Create RESTful APIs for users, complaints, messaging, authentication
- Implement role-based access control
- Integrate MongoDB with Mongoose for data modeling
- Add secure authentication using JWT or OAuth

Phase 5:week 7

Real-Time Features Communication

Objective: Enable user-agent interaction with live updates

- Integrate Socket.IO for real-time messaging
- Implement WebRTC if enabling video support
- Add notifications engine (email/SMS) using nodemailer or Twilio

Phase 6:week 8

Testing Debugging

Objective: Validate functionality, security, and UX flow

- Perform unit testing (Jest, Mocha) and integration testing
- Run bug fixing sprints and code reviews

- Conduct user testing sessions with feedback iterations

Phase 7: week 9

Deployment Documentation

Objective: Finalize launch and prepare for maintainability

- Use platforms like Heroku, Vercel, or AWS for deployment
- Write developer and API documentation
- Prepare training manual for agents and admin dashboard use
- Ensure compliance features are in place (privacy, encryption, retention policy)

Phase 8:week 10

Feedback Loop Enhancements (Week 10)

Objective: Collect live feedback and plan future sprints

- Monitor system behavior and collect user reviews
- Add analytics dashboard for complaint trends and satisfaction
- Identify advanced features (AI-based ticket prioritization, chatbot intake

6. Funtional and performance testing

FUNCTIONAL TESTING :

1. User Management Module

- Test Cases:
- Successful user registration and login/logout
- Validation of form inputs (e.g., email format, password rules)
- Role-based access enforcement (User, Agent, Admin)

2. Complaint Registration Module

- Test Cases:
- Complaint submission with/without file attachments
- Required fields validation
- Automatic complaint ID generation
- Predefined categories and dynamic form behavior

3. Complaint Tracking Notifications

- Test Cases:
- Complaint status changes (Submitted → Assigned → Resolved)
- Email/SMS notifications on state change
- Real-time updates reflected in user dashboard

4. Agent Interaction System

- Test Cases:
- Live chat functionality between agent and user via Socket.IO
- WebRTC-enabled video communication (if implemented)
- Agent reassignment by admin

5. Admin Dashboard
- Test Cases:
- Viewing and managing all complaints
- Assigning/reassigning complaints based on load or priority
- Viewing analytics (feedback scores, SLA reports)
6. Security Compliance
- Test Cases:
- Session management (timeout, concurrent sessions)
- Password hashing and secure storage
- Data access restrictions by role
- GDPR/DPDP consent flow

Performance testing

1. Load Testing
Goal:
Determine how many concurrent users the system can support before performance degrades. Tools:
Apache JMeter, Locust, Artillery
Scenarios:
- 500+ users submitting complaints simultaneously
- Multiple agents interacting with users in real time

2. Stress Testing
Goal:
Push the system beyond its limits to identify breaking points.
Scenarios:
- Simulating a sudden spike of 1000+ complaints within a short time window
- Testing database write-heavy scenarios

3. Scalability Testing
Goal:
Evaluate how well the system scales when additional resources are added.
Approach:
- Containerize backend services with Docker
- Test horizontal scaling by spinning up more instances

4. Stability Soak Testing
Goal:
Ensure the system remains stable over extended periods.
Scenarios:
- Running agents and users in live sessions for 24+ hours
- Monitoring memory leaks and database performance

5. Response Time Throughput
Metrics to Capture:
- API response time under peak usage
- Chat message delivery latency via Socket.IO
- Time to load the dashboard on slow connections

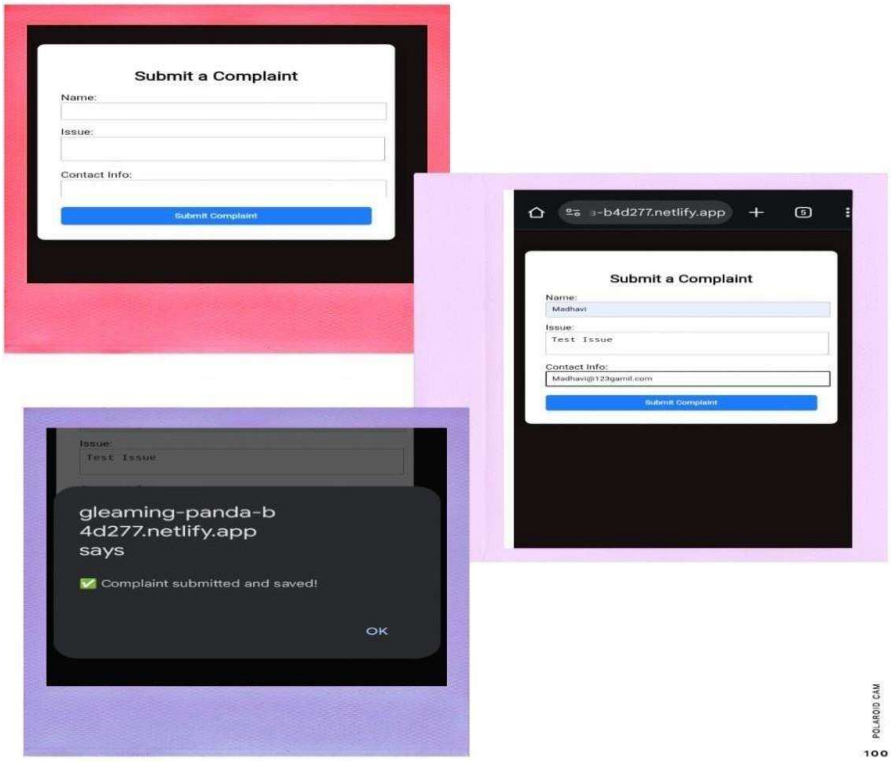
7. RESULTS

The first screen shows a blank form prompting the user to enter their name, issue, and contact info. - The second screen displays the form filled with sample data—"Madhavi," "Test Issue," and an email address. - The third screen presents a confirmation pop-up stating the complaint has been successfully submitted and saved. The image displays a user flow for a web-based complaint registration system, broken down into three clear stages across screenshots.

The first screen shows the initial state of a complaint form. It's a blank interface prompting the user to provide basic details like their name, the issue they're facing, and a method of contact. This layout appears simple and user-friendly, encouraging straightforward data entry.

The second screen illustrates the form filled out with example input. In this case, the user has entered a name ("Madhavi"), a placeholder issue ("Test Issue"), and a sample email address. This stage showcases how the interface accommodates user data and highlights proper field structure and validation readiness.

The third screen confirms the action: a modal popup or alert appears notifying the user that their complaint has been successfully submitted and saved. This immediate feedback helps reinforce a smooth user experience and ensures the user that their input has been recorded.



8. ADVANTAGES DISADVANTAGES

Advantages

Functional Benefits:

Centralized complaint handling streamlines user input, agent interaction, and admin oversight, improving transparency and responsiveness. Real-time tracking keeps users informed throughout the complaint lifecycle. Agent-user messaging enables direct communication, reducing turnaround time and enhancing resolution quality. User feedback ensures continuous improvement and service quality assessment. Smart routing algorithms match complaints with the most suitable agents or depart- ments.

Security and Compliance: User authentication and data encryption safeguard per- sonal and complaint-related data. Platform design supports adherence to data protection regulations, offering trust and legal compliance.

Technical Efficiency: A modular stack using Express.js and MongoDB enables scala- bility and flexibility. Material UI and Bootstrap contribute to a modern, responsive interface for different user roles. Socket.io and WebRTC provide real-time communi- cation and interactive experiences. MongoDB supports fast retrieval and dynamic schema handling, ideal for complaint-related data

Disadvantages

Functional Limitations: Users unfamiliar with digital platforms may feel overwhelmed by multiple features. Real-time interactions depend on agent availability, which can cause delays during peak hours.

Technical Challenges: Live features such as chat and video expose the system to security threats like XSS and WebSocket vulnerabilities. Persistent socket connec- tions increase server maintenance complexity. Browser compatibility and firewall restrictions may affect the performance of video chat via WebRTC. Without proper indexing, MongoDB queries may become inefficient with scale.

Operational Concerns: The routing mechanism requires well-maintained metadata and trained personnel to ensure accuracy. Maintaining compliance with privacy laws across regions adds legal and administrative overhead.

9. CONCLUSION

ResolveNow stands as a robust, user-centric solution for efficient and transparent complaint management in both consumer and organizational contexts. By integrating intuitive user interfaces, real-time communication tools, intelligent complaint routing, and a secure backend architecture, it empowers users to voice their concerns while ensuring swift, accountable responses from relevant agents or departments. The platform not only enhances customer trust and satisfaction but also equips organizations with valuable insights through feedback and system-generated analytics. Its scalable technical foundation—built on Express.js, MongoDB, Material UI, and WebRTC—ensures long-term adaptability, performance, and compliance with modern regulatory standards. In a landscape where responsiveness and user experience are paramount, ResolveNow is more than a digital complaint box—it's a complete ecosystem for managing resolution with clarity, speed, and empathy.

10. FUTURE SCOPE

1. **AI-Powered Complaint Classification and Prioritization:**
   Implementing machine learning models to automatically categorize complaints and assign priority levels based on urgency, sentiment analysis, or impact. This reduces manual triage and speeds up resolution timelines.
2. **Predictive Analytics for Issue Trends**:
   Leveraging historical complaint data to detect recurring issues, regional trends, or product-specific patterns. This would aid in preventive measures product

Multi-Language Support and Localization:
Enhancing accessibility by offering multilingual interfaces and localized

3. **Integration with CRM and ERP Systems:**
syncing with Customer Relationship Management (CRM) tools and Enter- prise Resource Planning (ERP) systems for holistic handling of customer data and workflows.

4. **Omnichannel Complaint Registration:**
Expanding submission channels via WhatsApp, voice bots, social media, or even IoT-enabled devices, making it more convenient and inclusive for users Seamless to lodge complaints from any device or platform.

5. **Self-Service Portals and Smart Chatbots:**
Introducing AI-driven chatbots for instant query resolution and a knowledge base of FAQs or previous cases to empower users with self-service capabilities before reaching agents.

6. **Blockchain for Tamper-Proof Logs:**
Integrating blockchain technology to create immutable logs of complaints and their resolution paths, ensuring transparency and building user trust—especially in high-stakes industries like insurance or healthcare.

7. **Mobile-First Experience:**
Building a fully responsive Progressive Web App (PWA) or native mobile app to ensure smooth, on-the-go access and complaint submission, particularly for remote and rural users.

8. **Real-Time Agent Analytics Dashboard:**
Providing agents and admins with live dashboards to monitor complaint resolution rates, backlog status, and user sentiment—enabling data-driven decision-making.

9. **Sentiment-Based Escalation Policies:**
Using sentiment analysis on user feedback or message tone to auto-escalate unresolved issues or unsatisfactory responses to supervisors for proactive resolution.

9. APPENDIX

WEBSITE LINK(FRONTEND): https://gleaming-panda-b4d277.netlify.app/
BACKEND: https://backend-ayyk.onrender.com
DEMO LINK:
https://drive.google.com/file/d/1gtnsfKyO7fMDGA3AMgTH23y3EZCl83CS/view?pli=1