

Full Stack Development with MERN

1. Introduction

- **Project Title:** ResolveNow – Your Platform for Online Complaints.

- **Team Members**

Team ID: LTVIP2026TMIDS84913

Team Size: 4

The project was developed and managed by the following team members:

1. **Lahari Nalamati – Team Leader and Backend**
Responsible for project planning, requirement analysis, sprint coordination, task allocation, overall supervision, and final review of the project deliverables.
2. **Lokesh Gubbala – Support and Coordination Member**
Assisted in general coordination activities, participated in team discussions, and provided support during documentation review and presentation preparation.
3. **Lova Raju Molli – Frontend Developer**
Responsible for user interface design, form development, dashboard implementation, and ensuring responsive layout across devices.
4. **Pallela Raja – Backend and Testing Lead**
Responsible for database management, backend logic implementation, system integration, testing activities, and bug tracking.

2. Project Overview

- **Purpose:** ResolveNow is a full-stack online complaint management platform designed to simplify complaint registration, tracking, and resolution. It provides a centralized system connecting customers, agents, and administrators to ensure transparency and faster issue resolution.

Features:

- Online complaint submission
- File attachment uploads
- Complaint tracking dashboard
- Real-time chat system
- Admin complaint assignment
- Agent resolution updates

- Feedback & rating system
- Secure JWT authentication

3. Architecture

• Frontend (React.js):

- Built using React with Vite.
- Component-based reusable UI.
- Axios for API calls.
- Socket.io client for real-time chat.
- Material UI / Bootstrap styling.
- Handles:
- Customer Portal
- Agent Portal
- Admin Dashboard

• Backend(NODE JS+EXPRESS JS):

- REST API server using Express.js.
- Business logic & routing.
- Middleware handling.
- File uploads via Multer.
- Real-time chat via Socket.io.

Modules:

- Complaint & Assignment
- File Upload
- Authentication

• Database (MONGO DB):

- NoSQL database.
- Managed via Mongoose ODM.

Collections:

- Users
- Complaints
- Assignments
- Messages
- Feedback

Stores complaint data, chat history, attachments, and ratings.

4. Setup Instructions

• Prerequisites:

Install:

- Node.js
- MongoDB
- npm
- Git

• **Installation:**

Commands:

Run the command in the terminal: git clone

https://github.com/lovaraju37/ResolveNow_Your-Platform-for-Online-Complaints.git

Go to the Directory: cd ResolveNow_Your-Platform-for-Online-Complaints

Later on open two terminals:

One for to run backend:

- cd "Project Files/backend"
- npm install

other to run frontend:

- cd "../frontend"
- npm install

.env file:

```
PORT=5000  
  
MONGO_URI=mongodb://localhost:27017/resolvenow
```

5. Folder Structure

• **Client:**

node_modules/

- Stores all installed npm packages and dependencies.
- Automatically created when running **npm install**.
- Should not be modified manually.

public/

- Contains static files.

- These files are served directly to the browser.

assets/

- Stores images and media files.
- Used inside components for UI display.

components/

admin/

- Admin dashboard components.
- User management.
- Complaint assignment modules.

agent/

- Agent dashboard.
- Assigned complaints view.
- Complaint status update features.

user/

- Complaint submission forms.
- Complaint tracking pages.
- Chat interface for users.

App.jsx

- Root component of the React app.
- Handles routing and page navigation.
- Loads different components based on URL paths.

main.jsx

- Entry point of the React application.
- Renders the App component into the DOM using ReactDOM.

App.css

- Styles specific to the App component.

index.css

- Global styling file.
- Applies styles across the entire application.

index.html

- Root HTML template.
- Contains `<div id="root"></div>` where React mounts the app.

package.json

- Contains project metadata
- Lists dependencies and scripts

6. Running the Application

- Provide commands to start the frontend and backend servers locally.
 - o **Backend:** Open a terminal and navigate to the backend directory inside the “Project Files” folder. Run the command “npm install” to install all required dependencies.

After installation, run “npm start” to start the backend server. The backend will run on <http://localhost:5000>.

- o **Frontend:** Open a new terminal and navigate to the frontend directory inside the Project Files folder. Run “npm install” to install dependencies if not already installed.

Then run “npm start” to launch the frontend application. The frontend will run on <http://localhost:5173>.

7. API Documentation

The backend of the ResolveNow platform exposes RESTful APIs to handle authentication, complaints, assignments, messaging, feedback, and user management.

Base URL:

<http://localhost:5000>

1. Authentication APIs

POST /api/auth/register

Description: Registers a new user (Customer, Agent, or Admin).

Request Body: name, email, password, role

Response: Success message and user details.

POST /api/auth/login

Description: Authenticates user and returns a JWT token.

Request Body: email, password

Response: JWT token and user information.

POST /api/auth/logout

GET /api/auth/agents

Description: Returns list of available agents (Admin access).

2. Complaint APIs

POST /api/complaints

Description: Create a new complaint.

Request Body: title, description, category, attachment (optional)

Response: Created complaint object.

GET /api/complaints

Description: Retrieve all complaints (based on role).

GET /api/complaints/:id

Description: Retrieve a specific complaint by ID.

PUT /api/complaints/:id

Description: Update complaint details or status.

DELETE /api/complaints/:id

Description: Delete a complaint (Admin access).

3. Assignment APIs

POST /api/assigned

Description: Assign complaint to an agent.

Request Body: complaintId, agentId

Response: Assignment confirmation.

GET /api/assigned

Description: Get all complaint assignments.

GET /api/assigned/agent/:agentId

Description: Get complaints assigned to a specific agent.

4. Message APIs

POST /api/messages

Description: Send a chat message related to a complaint.

Request Body: complaintId, senderId, message

GET /api/messages/:complaintId

Description: Retrieve chat history for a complaint.

PUT /api/messages/read/:complaintId

Description: Mark messages as read.

GET /api/messages/unread/counts

Description: Get unread message count for a user.

5. Feedback APIs

POST /api/feedback

Description: Submit feedback after complaint resolution.

Request Body: complaintId, rating, comment

GET /api/feedback/complaint/:complaintId

Description: Get feedback for a specific complaint.

GET /api/feedback/agent/:agentId

Description: Get feedback received by an agent.

8. Authentication

Authentication and authorization in the ResolveNow platform are implemented using **JSON Web Tokens (JWT)** to ensure secure access to the system.

Login Authentication

When a user logs in with valid credentials (email and password), the backend verifies the details from the MongoDB database.

If the credentials are correct, the server generates a JWT token and sends it to the client.

This token contains encoded user information such as:

- User ID
- Email
- Role (Admin / Agent / Customer)

9. User Interface

The **ResolveNow – Online Complaints Platform** is designed with a clean, responsive, and user-friendly interface to ensure ease of use for citizens, support staff, and administrators. The UI focuses on simplicity, accessibility, and smooth navigation.

Below are the key UI screens to be included as screenshots or GIF demonstrations in the final documentation.

9.1 Login & Registration Page

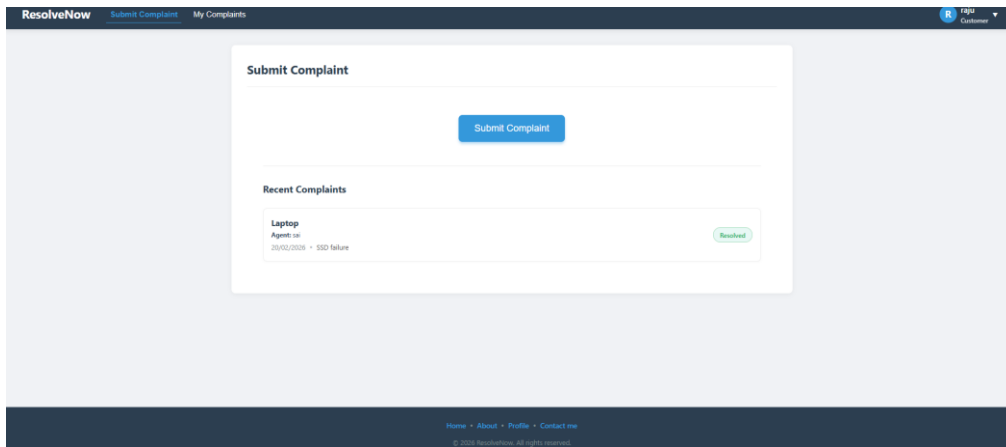
The screenshot shows a web page for 'ResolveNow' with a dark blue header and footer. The main content area is light blue. In the center, there is a dark blue card titled 'SignUp For Registering the Complaint' with the subtitle 'Please enter your details'. The card contains the following fields: 'Name' (text input), 'Email' (text input), 'Phone Number' (text input), 'Register As' (dropdown menu with 'Customer' selected), 'Password' (text input), and 'Confirm Password' (text input). Below these fields is a 'Sign Up' button. At the bottom of the card, there is a link: 'Already have an account? [Login](#)'. The footer contains navigation links: 'Home', 'About', 'Login', 'SignUp', 'Contact me' and a copyright notice: '© 2024 ResolveNow. All rights reserved.'

The screenshot shows a web page for 'ResolveNow' with a dark blue header and footer. The main content area is light blue. In the center, there is a dark blue card titled 'Login For Registering the Complaint' with the subtitle 'Please enter your credentials'. The card contains the following fields: 'Email' (text input) and 'Password' (text input). Below these fields is a 'Login' button. At the bottom of the card, there is a link: 'Don't have an account? [SignUp](#)'. The footer contains navigation links: 'Home', 'About', 'Login', 'SignUp', 'Contact me' and a copyright notice: '© 2024 ResolveNow. All rights reserved.'

UI Features:

- Responsive design
- Form validation alerts
- Clean layout with branding

9.2 User Dashboard



UI Features:

- Status color indicators
- Filter and search functionality
- Simple navigation sidebar

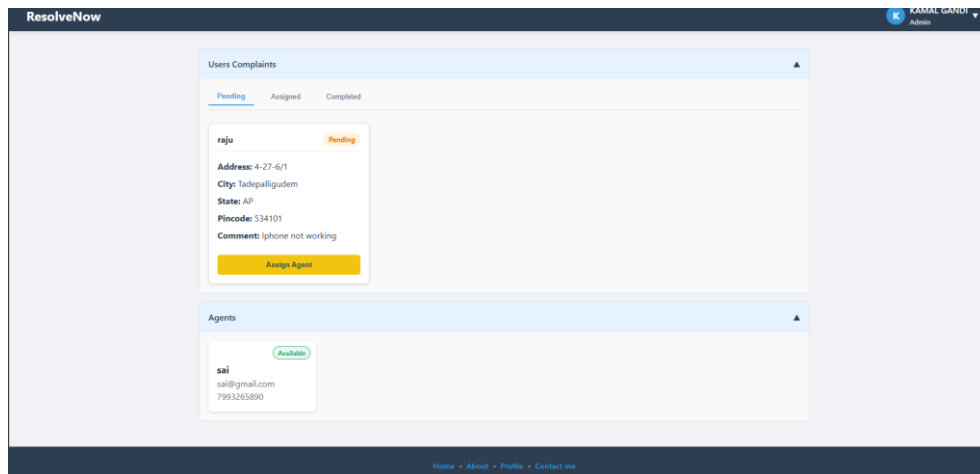
9.3 Raise New Complaint Page

 This screenshot shows the 'Submit Complaint' form on the ResolveNow website. The form is white and set against a light gray background. It includes a 'Back' button at the top left. The form fields are organized into sections: 'Product Name' (with a placeholder 'Enter product name'), 'Address' (with a placeholder 'House 165, Street Area'), and 'City'. Below these are 'State' and 'Pincode' fields. A large text area for 'Description of Complaint' has a placeholder 'Please describe your issue in detail...'. An 'Attachments (Optional)' section features a blue '+ Add attachment' button and a note about supported formats (JPG, PNG, PDF, DOC) and file size (Max size 5MB per file). A final 'Submit Complaint' button is at the bottom right. The header and footer are consistent with the previous screenshot.

UI Features:

- Input validation
- Category selection
- Confirmation message after submission

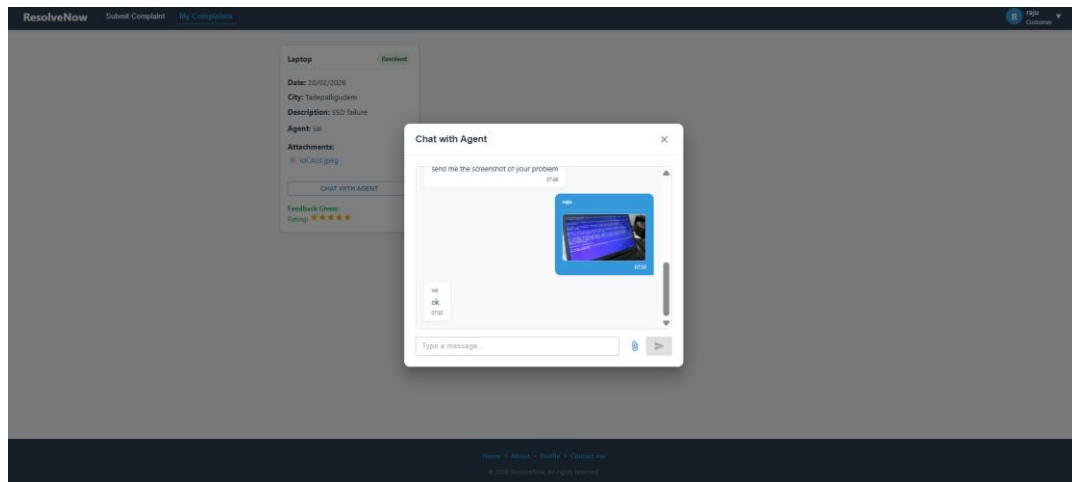
9.4 Admin Dashboard



UI Features:

- Real-time status update
- Priority highlighting

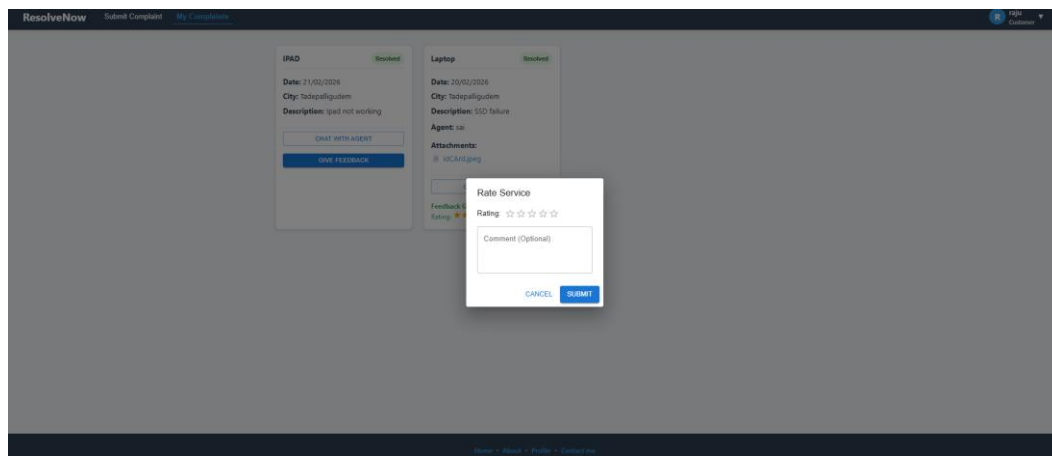
9.5 Chat / Communication Interface



UI Features:

- Clear conversation layout
- Auto-scroll to latest message
- Notification indicator

9.6 Feedback & Rating Page



UI Features:

- Star-based rating selection
- Submission confirmation

10. Testing

The ResolveNow – Online Complaints Platform followed a structured and systematic testing approach to ensure quality, reliability, and performance before deployment. Testing was conducted at multiple levels to validate both functional and non-functional requirements.

10.1 Testing Strategy

The project adopted a multi-level testing strategy as described below:

1. Unit Testing

- Individual modules were tested separately.
- Focused on validating small components such as login validation, complaint submission, and status updates.
- Ensured that each function works independently without errors.

2. Integration Testing

- Verified interaction between modules (e.g., complaint submission → admin dashboard → assignment → status update).

- Ensured smooth data flow between frontend and backend systems.

3. System Testing

- Tested the complete system as a whole.
- Validated end-to-end workflows including complaint registration, tracking, chat, and feedback.

4. User Acceptance Testing (UAT)

- Conducted by end users to ensure the system meets business requirements.
- Verified usability, functionality, and real-world scenarios.
- All critical test cases were executed and approved.

5. Regression Testing

- Performed after bug fixes to ensure that existing functionalities were not affected.

6. Exception & Negative Testing

- Tested invalid inputs (empty fields, incorrect credentials).
- Verified system behavior during unexpected errors.

10.2 Types of Testing Performed

- Functional Testing
- Non-Functional Testing
- Performance Testing
- Security Testing
- Usability Testing
- Compatibility Testing (Desktop & Mobile)

10.3 Testing Tools Used

Tool	Purpose
Manual Testing	Execution of functional test cases
Browser Developer Tools	Debugging frontend issues
Git	Version control and change tracking
JIRA / Excel	Bug tracking and test case management

11. Screenshots or Demo

Video Demonstration link: [ResolveNow Demo](#)

12. Known Issues

The following known issues have been identified in the current version of the ResolveNow – Online Complaints Platform. These issues are minor and do not affect the core functionality of the system.

1. Minor UI alignment issues may appear on smaller mobile screens. This does not impact functionality but may slightly affect visual appearance.
2. File upload size is limited (for example, up to 5MB). Files larger than the limit are not accepted and display a validation message.
3. Email notifications may experience slight delays depending on server response time.
4. The system is fully optimized for modern browsers such as Chrome and Edge. Minor display inconsistencies may occur in older browser versions.
5. The reporting module currently provides summary-level reports only. Advanced

filtering and detailed analytics are limited in the current version.

All critical and high-severity issues have been resolved. The system is stable and suitable for deployment.

13. Future Enhancements

The following enhancements are proposed for future versions of the system to improve performance, scalability, and user experience.

1. Development of a dedicated mobile application for Android and iOS platforms to increase accessibility.
2. Implementation of an advanced analytics dashboard with interactive charts, category-wise analysis, time-based reports, and export options such as PDF and Excel.
3. Integration of AI-based complaint categorization to automatically classify complaints and prioritize urgent cases.
4. Addition of SMS and push notification features to provide real-time updates to users.
5. Multi-language support to improve accessibility for users from different regions.
6. Enhanced role-based access control with more granular permission settings for administrators and sub-administrators.
7. Cloud deployment and load balancing to improve scalability and handle high user traffic efficiently.
8. Integration with third-party systems such as CRM platforms and support for external APIs.

These enhancements aim to transform the system into a more intelligent, scalable, and user-friendly complaint management platform in future releases.