Project Report: Service Desk for Customer Complaint Resolution

# PROJECT TEAM DETAILS

|  |  |
| --- | --- |
| **Team ID** | LTVIP2025TMID57241 |
| **Team Leader** | Gundumogula Lakshmi Prasanna |
| **Team Member** | Eswar Adesh Ch |
| **Team Member** | Dumpa Charmika Reddy |
| **Team Member** | Eppili Goutham |

**1. INTRODUCTION**

**1.1 Project Overview**

The Service Desk Application is a full-stack web-based complaint management system designed to streamline the lifecycle of service requests. It provides a centralized platform for users to register complaints, for agents to respond and resolve issues, and for administrators to manage roles and oversee the overall operations.

**1.2 Purpose**

The purpose of this report is to document the entire development cycle of the Service Desk Application—from ideation to deployment. It outlines problem identification, solution design, requirement specifications, planning, development, and testing to provide a holistic understanding of the system.

**2. PROJECT OVERVIEW**

**2.1 Purpose**

ResolveNow is a comprehensive platform that streamlines complaint handling and service request management for organizations. It bridges the communication gap between users and agents, offers real-time tracking, and provides administrative oversight tools.

**2.2 Features**

* Secure JWT-based authentication for multiple roles (user, agent, admin)
* Complaint registration and live tracking
* Real-time messaging via Socket.io
* Role-based dashboards
* Complaint assignment and status update system
* Admin panel for user and agent management

**3. SYSTEM ARCHITECTURE**

**3.1 Frontend**

* Built with React.js and Tailwind CSS
* Role-based components under src/components/ for client, technician, and supervisor
* Real-time updates using WebSockets (Socket.io-client)
* Reusable UI elements (forms, dashboards, communication windows)

**3.2 Backend**

* Developed using Node.js and Express.js
* Organized in modular folders (models, routes, middleware)
* RESTful API endpoints secured with JWT middleware
* Real-time communication using Socket.io

**3.3 Database**

* MongoDB with Mongoose ODM
* **Collections**: users, complaints, messages
* **Schemas**:
  + User.js: name, email, password (hashed), role
  + Complaint.js: subject, description, status, assigned agent, timestamps
  + Message.js: sender, recipient, message text, timestamps

**4. SETUP INSTRUCTIONS**

**4.1 Prerequisites**

* Node.js (v16+)
* MongoDB installed or hosted (e.g., MongoDB Atlas)
* Git

**4.2 Installation**

* Clone the repository
* git clone https://github.com/EswarAdeshCh/Service\_Desk.git
* cd Service\_Desk
* Setup backend
* cd backend
* npm install
* touch .env
* # Add MongoDB URI, JWT\_SECRET, etc. in .env
* Setup frontend
* cd ../frontend
* npm install

**5. FOLDER STRUCTURE**

**5.1 Client (Frontend)**

frontend/

├── public/

│ └── placeholder images and index.html

├── src/

│ ├── components/

│ │ ├── auth/

│ │ │ ├── AuthenticationPage.jsx

│ │ │ └── RegistrationPage.jsx

│ │ ├── client/

│ │ │ ├── ClientPortal.jsx

│ │ │ ├── IssueSubmissionForm.jsx

│ │ │ └── IssueTracker.jsx

│ │ ├── common/

│ │ │ ├── CommunicationWindow.jsx

│ │ │ └── WelcomePage.jsx

│ │ ├── supervisor/

│ │ │ ├── ClientManagement.jsx

│ │ │ ├── SupervisorHub.jsx

│ │ │ └── TechnicianManagement.jsx

│ │ └── technician/

│ │ └── TechnicianWorkspace.jsx

├── app.jsx

├── index.js

**5.2 Server (Backend)**

backend/

├── middleware/

│ └── auth.js

├── models/

│ ├── Complaint.js

│ ├── Message.js

│ └── User.js

├── routes/

│ ├── admin.js

│ ├── agents.js

│ ├── auth.js

│ ├── complaints.js

│ ├── messages.js

│ └── users.js

├── server.js

**6. RUNNING THE APPLICATION**

**6.1 Frontend**

cd frontend

npm start

**6.2 Backend**

cd backend

npm start

**7. AUTHENTICATION**

**7.1 Token-based Authentication**

* JWTs issued during login
* Stored in localStorage/cookies on frontend
* auth.js middleware verifies tokens and grants role-based access

**7.2 Password Encryption**

* Bcrypt used to hash user passwords before storing in MongoDB

**8. USER INTERFACE**

**8.1 Role-based Pages**

* **Login/Registration Pages**
* **User Dashboard**: Complaint submission and tracking
* **Agent Dashboard**: Assigned complaints and chat interface
* **Admin Dashboard**: Complaint assignment, user/agent management

**9. TESTING**

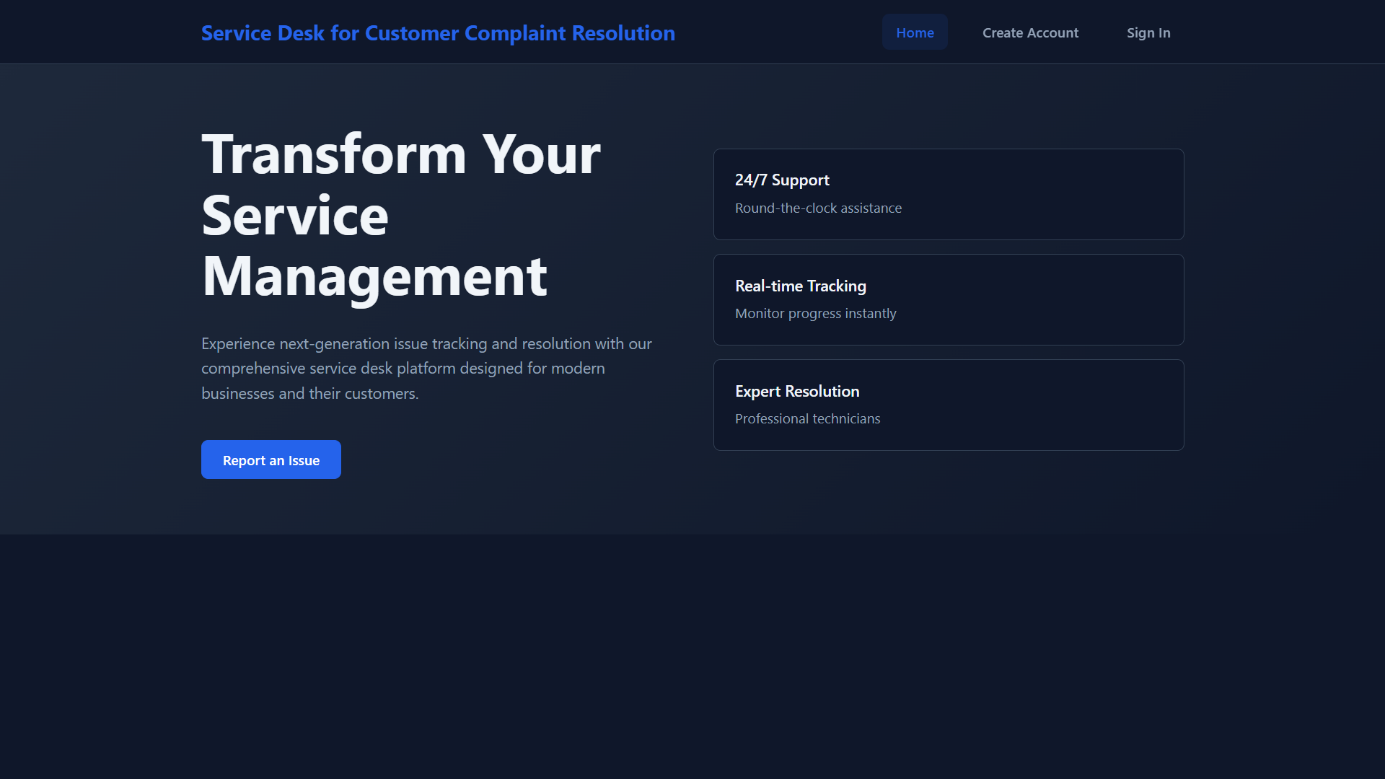
**9.1 Manual Testing Tools**

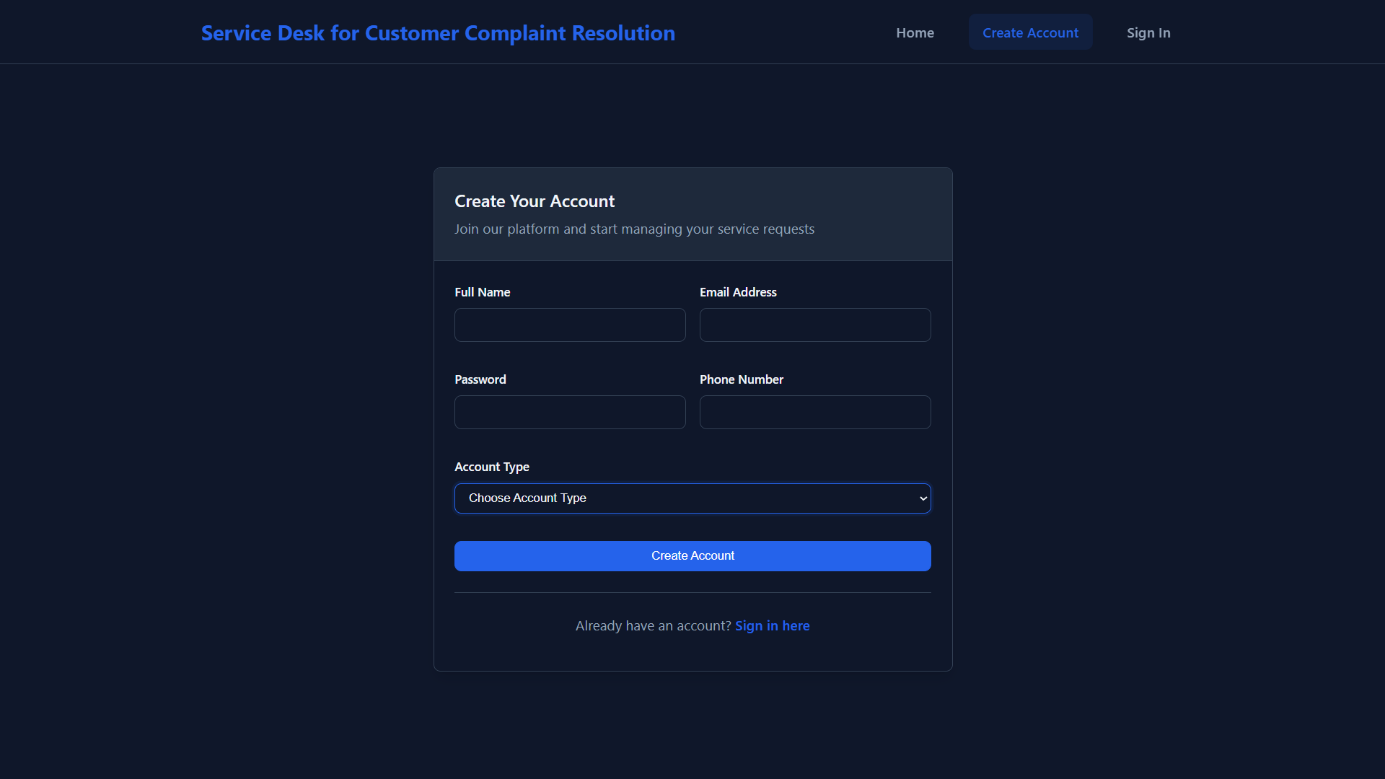
* Postman (API testing)
* Browser DevTools (UI & WebSocket testing)

**9.2 Automated Testing (Future Scope)**

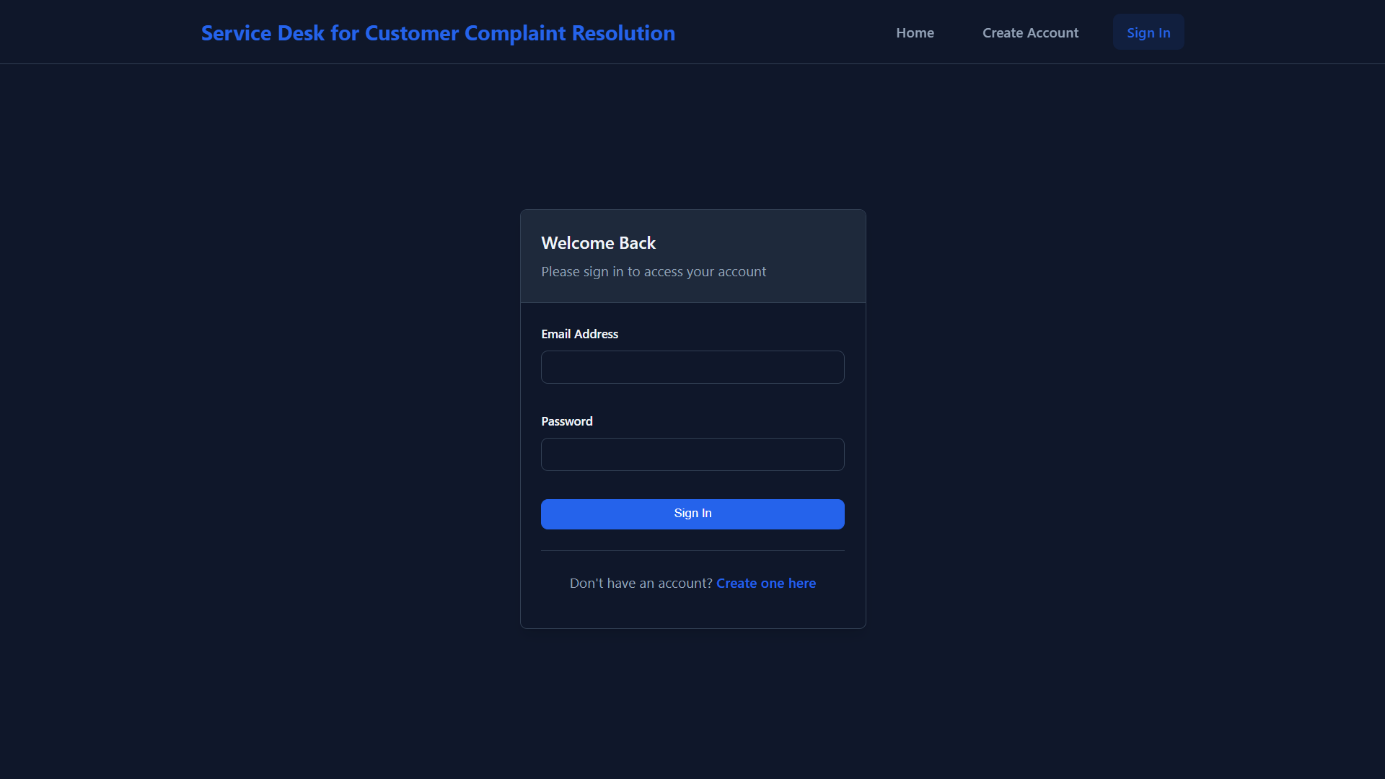
* Jest for backend
* React Testing Library for frontend

**10. SCREENSHOTS OR DEMO**

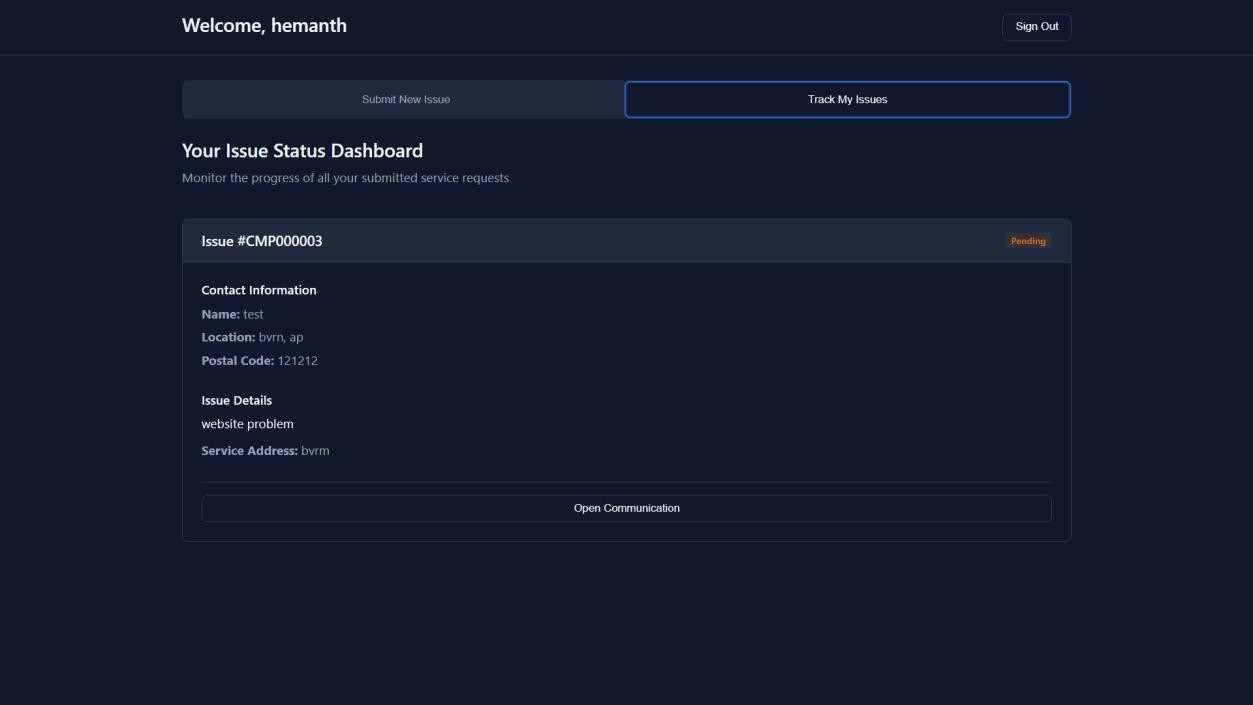
* + - Landing Page
    - Signup Page

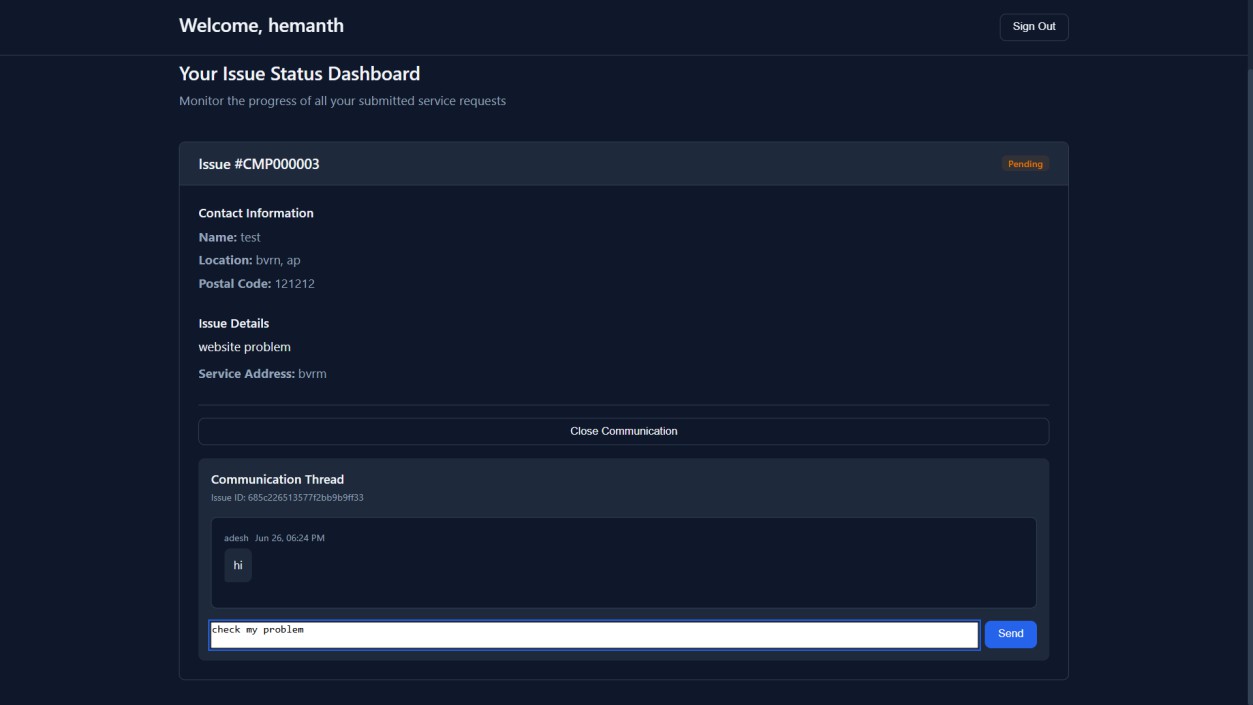


* + - Login Page

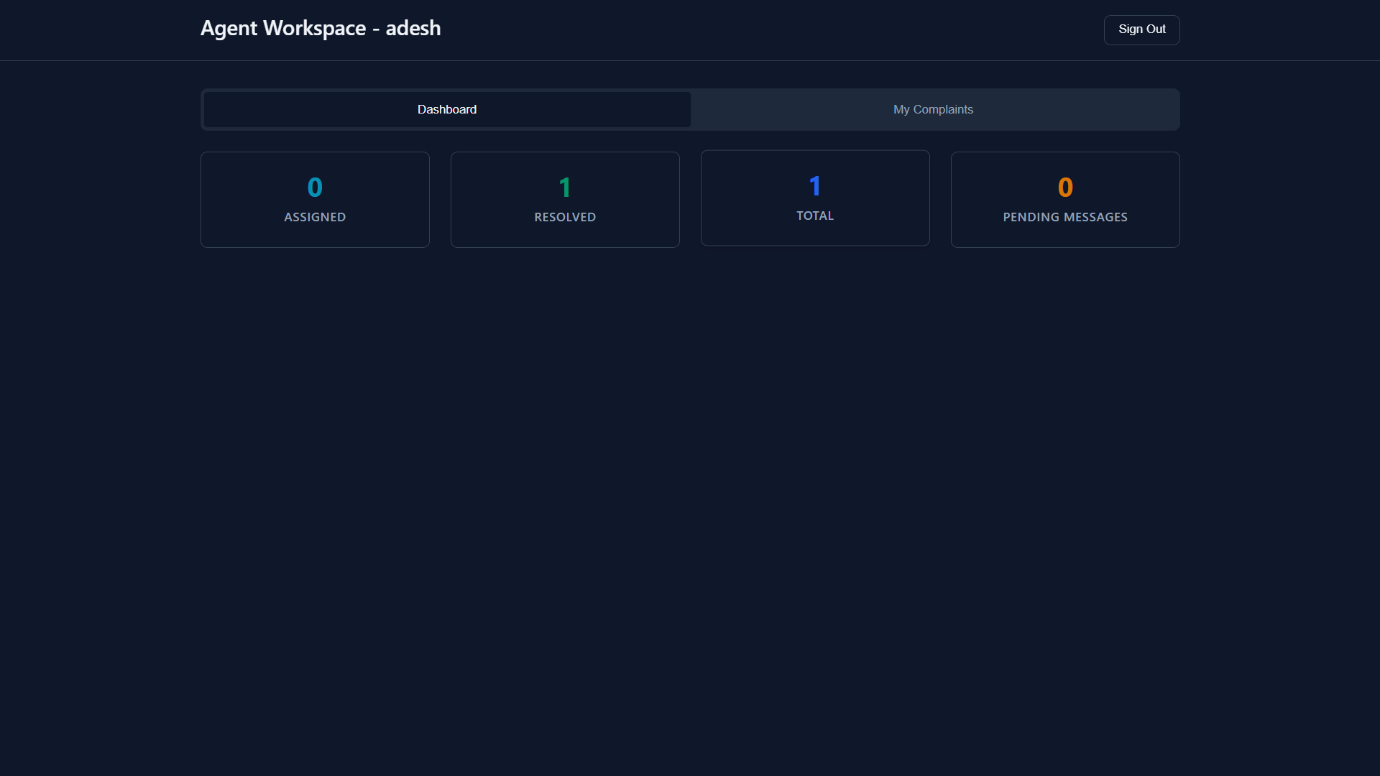


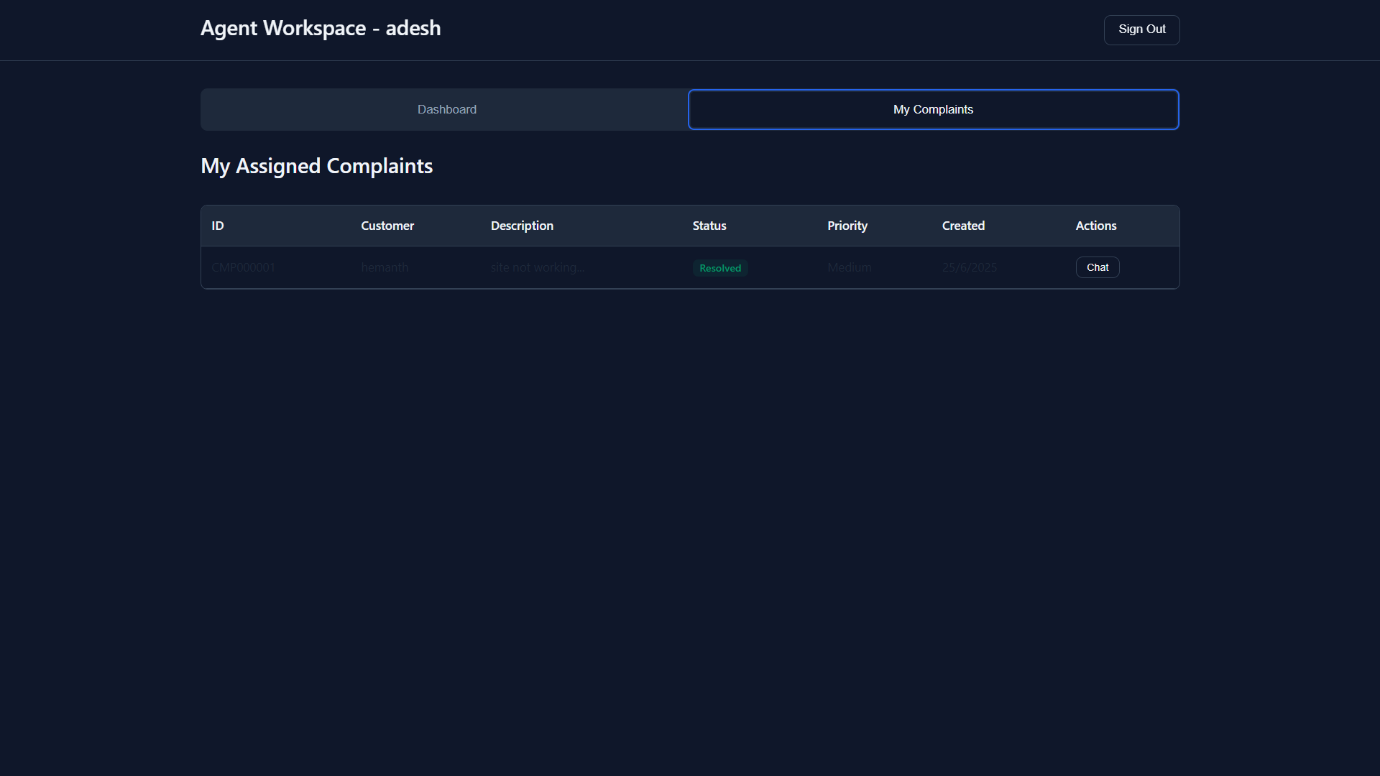
* + - User Complaint Dashboard

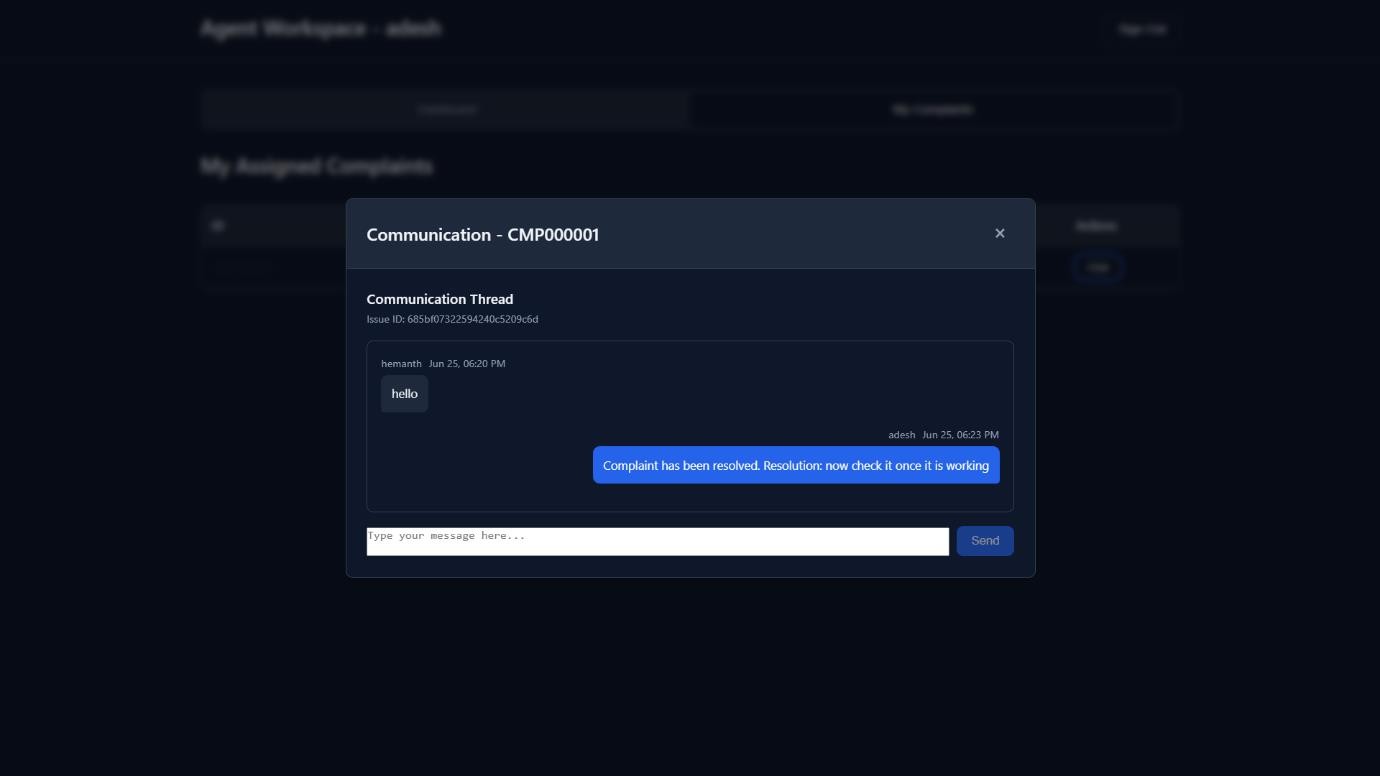




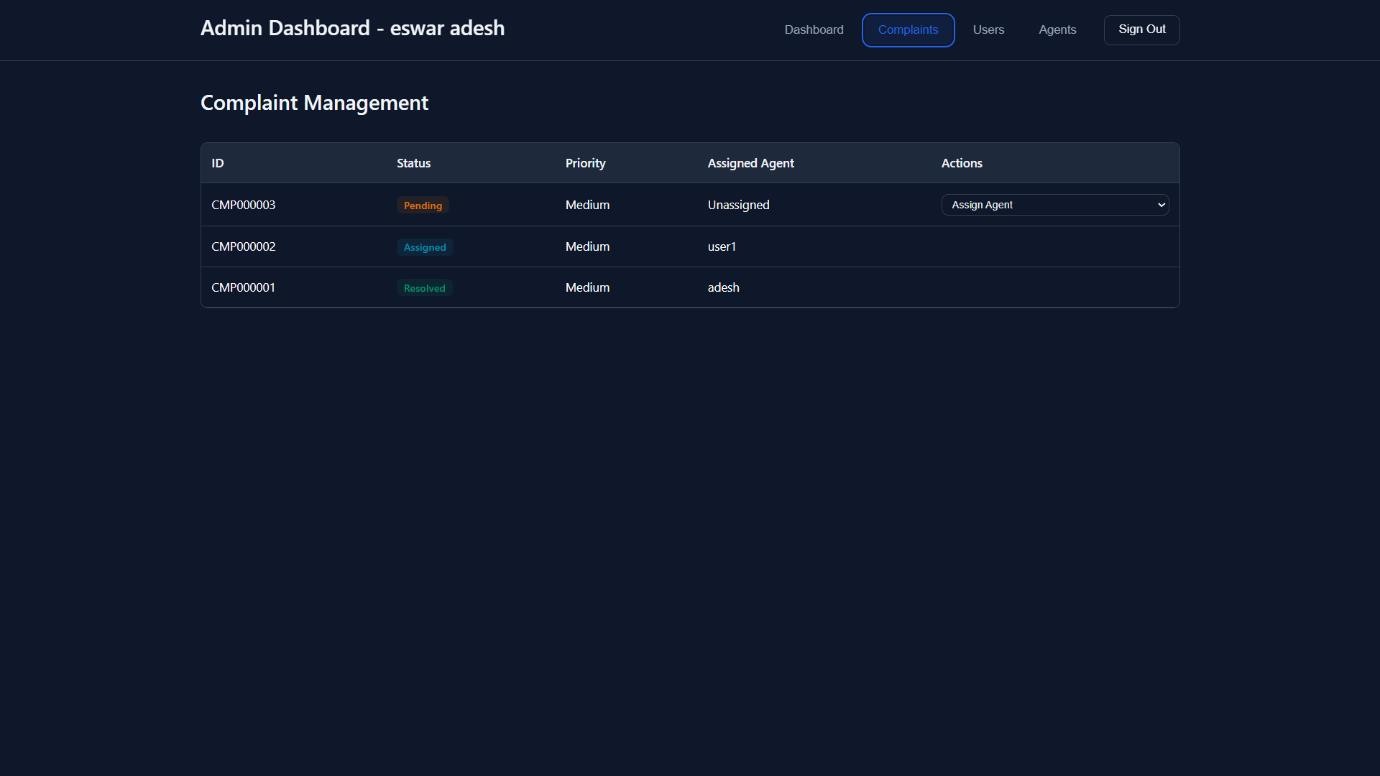
* + - Agent Complaint View

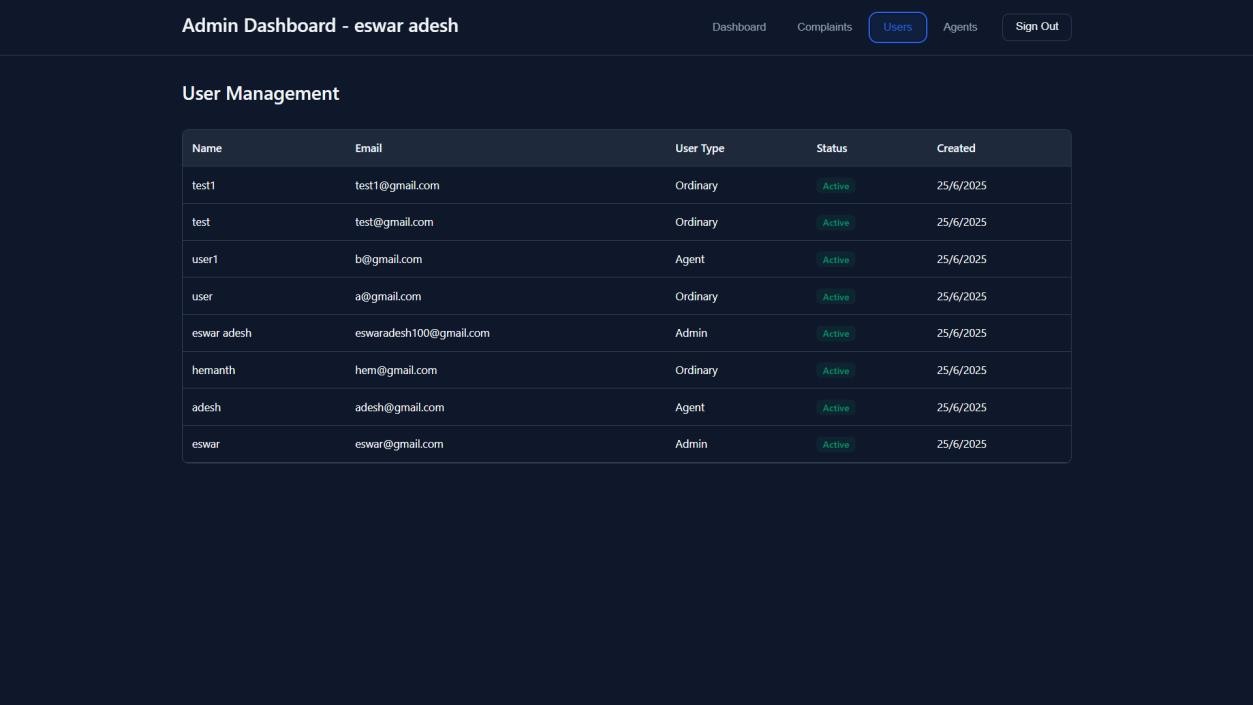


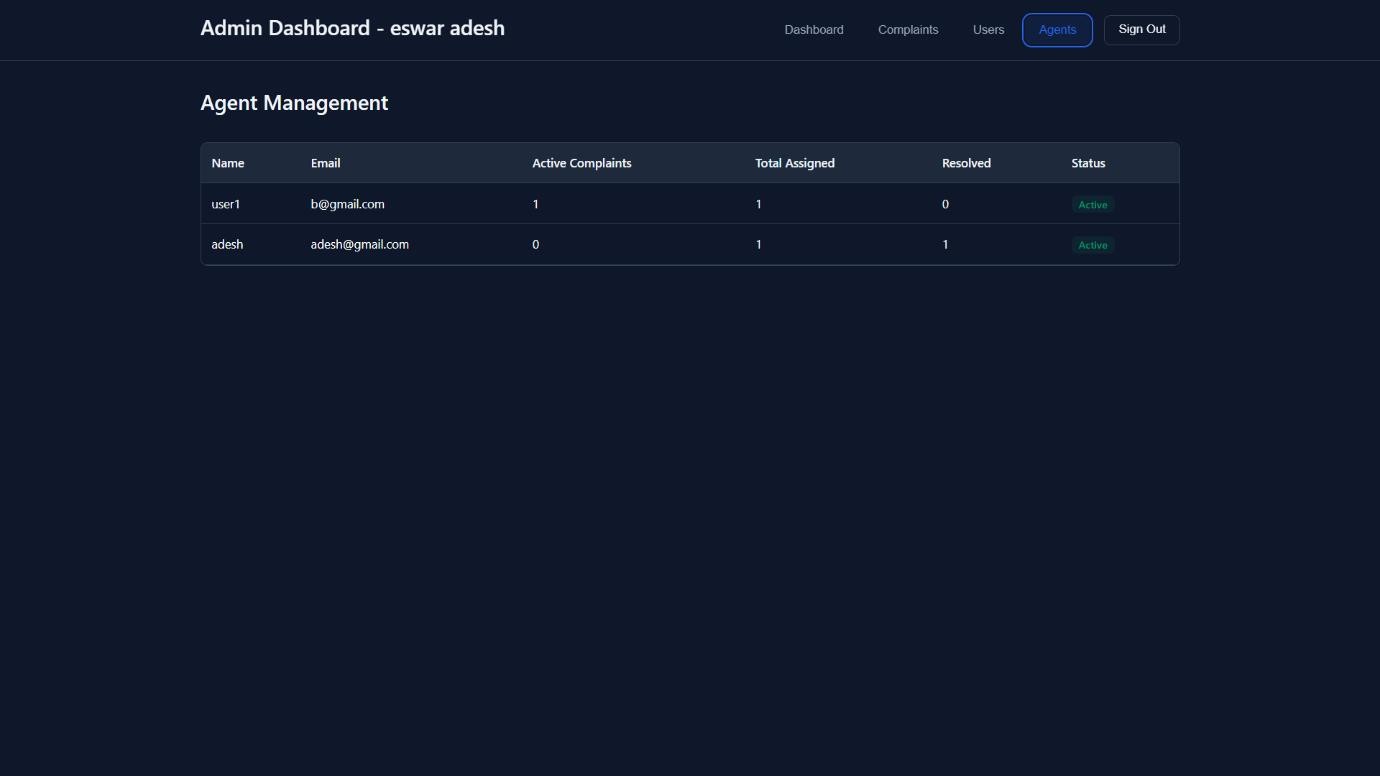




* + - Admin Dashboard







**11. GITHUB & PROJECT DEMO LINK**

* **Source Code GitHub Link :** <https://github.com/EswarAdeshCh/Service_Desk>
* **Demo Video Link:** <https://youtu.be/xfFbtcBX9Cg?si=P-yqe0JoEMIMbId3>

**12. Known Issues**

* No offline support
* Initial deployment requires stack familiarity
* Mobile responsiveness needs further refinement

**13. Future Enhancements**

* Mobile app using React Native or Flutter
* AI chatbot integration for complaint intake
* SMS and email notifications
* Multi-language support
* Advanced analytics and visual reports
* Integration with CRM systems