Project Report: Service Desk Application

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. IDEATION PHASE

2.1 Problem Statement

Traditional complaint-handling methods are inefficient, lack transparency, and result in user dissatisfaction. Manual tracking leads to delays and poor communication between users and support staff. This project addresses the need for a real-time, accessible, and role-driven complaint resolution platform.

2.2 Empathy Map Canvas

Role	**Says**	**Thinks**	**Feels**	**Does**
User	"Why is no one responding?"	"I hope this gets resolved fast"	Frustrated, anxious	Submits complaint
Agent	"Too many complaints!"	"I must prioritize cases"	Pressured, responsible	Reviews and resolves complaints
Admin	"I need to balance workload"	"The system must stay organized"	Accountable, focused	Assigns roles, monitors activity

2.3 Brainstorming

Initial feature ideas:

- Complaint submission form with file uploads
- Real-time status updates
- Live chat between user and agent
- Role-based dashboards
- Notification system (future scope)
- Admin complaint assignment system
- Analytics for future improvement

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

User:

- Registers and logs in
- Submits a complaint with details
- Tracks complaint progress
- Communicates with the assigned agent
- Provides feedback after resolution

Agent:

- Logs in
- Views assigned complaints
- Communicates with users
- Updates complaint status
- Resolves and closes complaint

Admin:

- Logs in
- Monitors complaints
- Assigns complaints to agents
- Manages user and agent roles
- Accesses dashboard analytics

3.2 Solution Requirement

The system must:

- Authenticate users securely
- Allow complaint registration and updates
- Provide chat functionality

- Support role-based access
- Ensure real-time data consistency
- Be responsive and user-friendly

3.3 Data Flow Diagram

- User → [React UI] → [Express Backend] → [MongoDB]
- Chat: WebSocket → Backend → Socket.io Server → Client Chat Window

3.4 Technology Stack

Layer	**Technology**
Frontend	React.js, Tailwind CSS, Bootstrap
Backend	Node.js, Express.js
Database	MongoDB with Mongoose ODM
Authentication	JWT + Bcrypt
Real-Time	Socket.io
Tools	Postman, GitHub, VSCode, Nodemon

---

4. PROJECT DESIGN

4.1 Problem Solution Fit

The system bridges the gap between users and agents with live tracking and communication. It reduces resolution delays and enhances customer satisfaction through transparency and automation.

4.2 Proposed Solution

A single-page web application (SPA) powered by React for the frontend and Node.js/Express for the backend. MongoDB stores structured complaint and user data. Real-time messaging via Socket.io enhances responsiveness.

4.3 Solution Architecture

```
rust
CopyEdit
User <--> React Frontend <--> Express API <--> MongoDB
                         ↑                    ↓
                   Socket.io Client     Mongoose Models
```

- JWT for secure access
- REST API routes handle core CRUD operations
- WebSockets handle real-time messaging

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Milestone 1: Environment Setup and Project Initialization

Objective: Establish the foundational structure and development environment for both backend and frontend components.

Key Activities:

- Set up project directory structure (`/backend`, `/frontend`)
- Initialize Git repository with proper `.gitignore`
- Install Node.js, MongoDB, and supporting tools (VS Code, Postman)
- Create initial configuration files:
  - `.env` for environment variables
  - `package.json` for dependency tracking

Deliverables:

- Working dev environment with separate backend and frontend folders
- Dependencies installed and linked
- GitHub repository with initial commit

Milestone 2: Backend Development

Objective: Implement the server-side logic for user authentication, complaint handling, messaging, and role-based access.

Key Activities:

- Create Express server and configure middleware (CORS, body-parser, Helmet)
- Design MongoDB schemas using Mongoose:
  - User, Complaint, Message
- Set up JWT-based authentication and Bcrypt for password hashing
- Develop REST API routes for:
  - `auth`, `users`, `agents`, `admin`, `complaints`, `messages`
- Apply route-level middleware for authorization

Deliverables:

- Secure, modular REST API
- Connected MongoDB database

- JWT-protected route access
- Schema-based data validation

Milestone 3: Frontend Development

Objective: Develop a responsive user interface with React and integrate it with the backend API.

Key Activities:

- Create reusable UI components: Login, Dashboard, ComplaintForm, ChatWindow
- Configure routing using React Router
- Integrate Axios for API communication
- Apply styling with Tailwind CSS and Bootstrap
- Implement conditional rendering for user roles (User, Agent, Admin)

Deliverables:

- Fully functional SPA for all roles
- Dynamic dashboards based on role
- Complaint submission and listing integrated with backend
- Chat component UI setup

Milestone 4: Real-Time Messaging Integration

Objective: Enable live chat functionality between users and agents using WebSockets via Socket.io.

Key Activities:

- Integrate Socket.io server in Express backend
- Connect Socket.io client in React frontend
- Build real-time chat interface tied to specific complaint IDs
- Store messages in MongoDB using `Message` schema
- Ensure secure and scoped messaging per user and complaint

Deliverables:

- Bi-directional chat system
- Persistent message storage
- Realtime updates and notifications within dashboard

Milestone 5: Admin Features and Role Management

Objective: Implement the administrator’s dashboard and functionality for managing users, agents, and complaints.

Key Activities:

- Build admin panel to view all complaints, users, and agents
- Add interfaces for:
  - Role assignment
  - Complaint reassignment
  - User deactivation
- Implement filters and status grouping for complaint views

Deliverables:

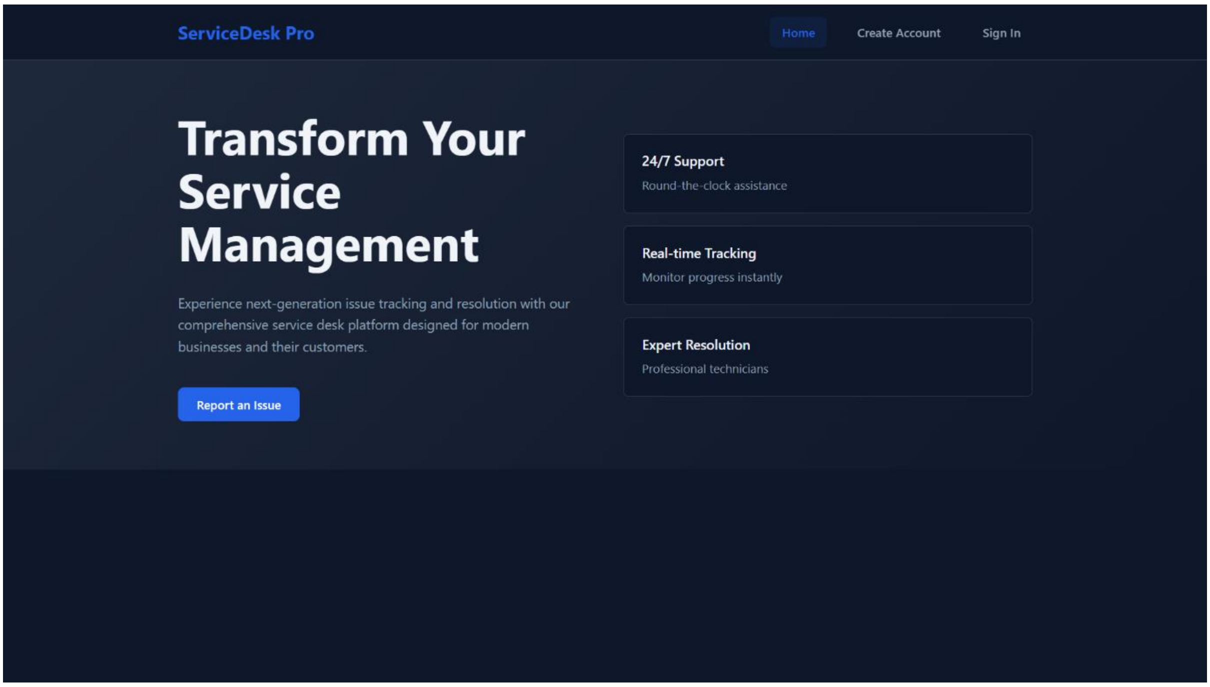
- Fully featured admin dashboard
- Functional role and complaint management tools
- Complaint assignment logic with validation

---

6. RESULTS

6.1 Output Screenshots

- Landing Page



Transform Your Service Management

Experience next-generation issue tracking and resolution with our comprehensive service desk platform designed for modern businesses and their customers.

24/7 Support

Round-the-clock assistance

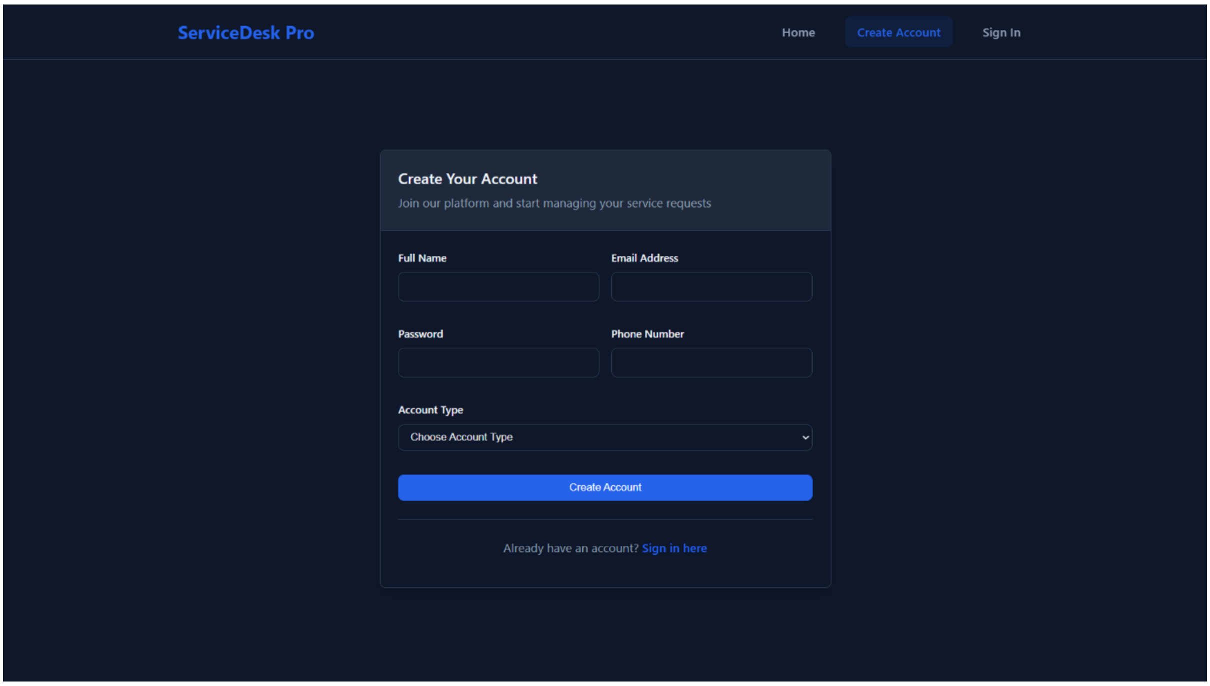
Real-time Tracking

Monitor progress instantly

Expert Resolution

Professional technicians

- Signup Page

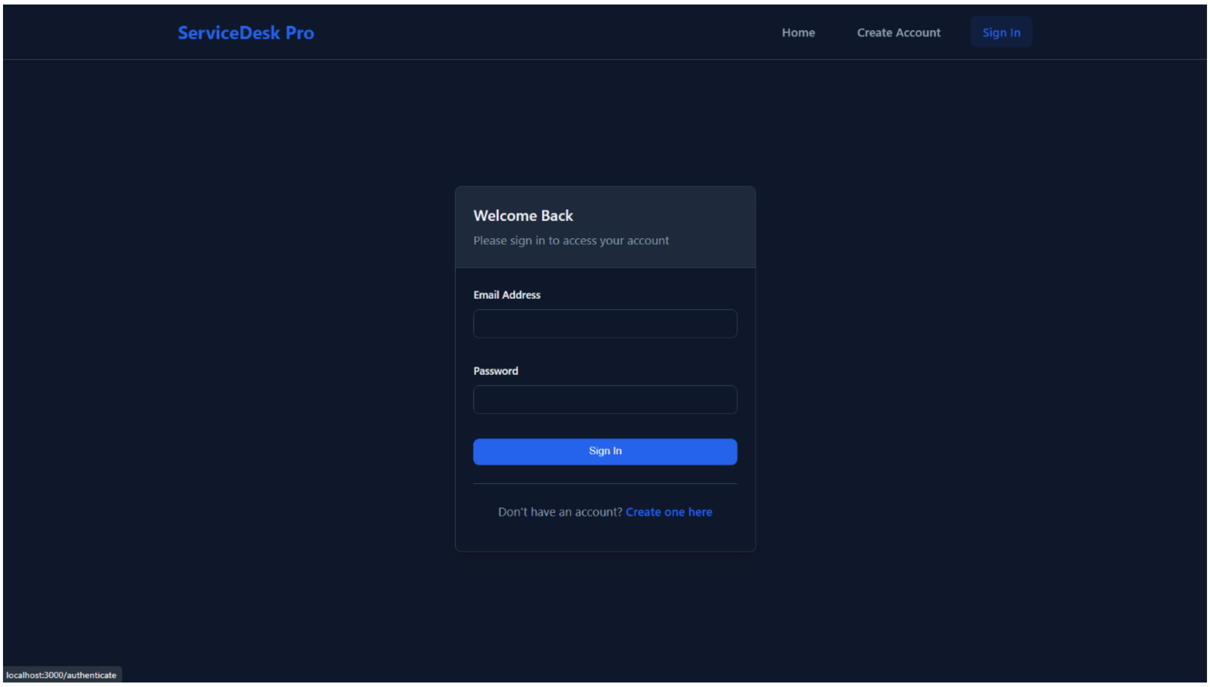


Create Your Account

Join our platform and start managing your service requests

Already have an account? Sign in here

- Login Page

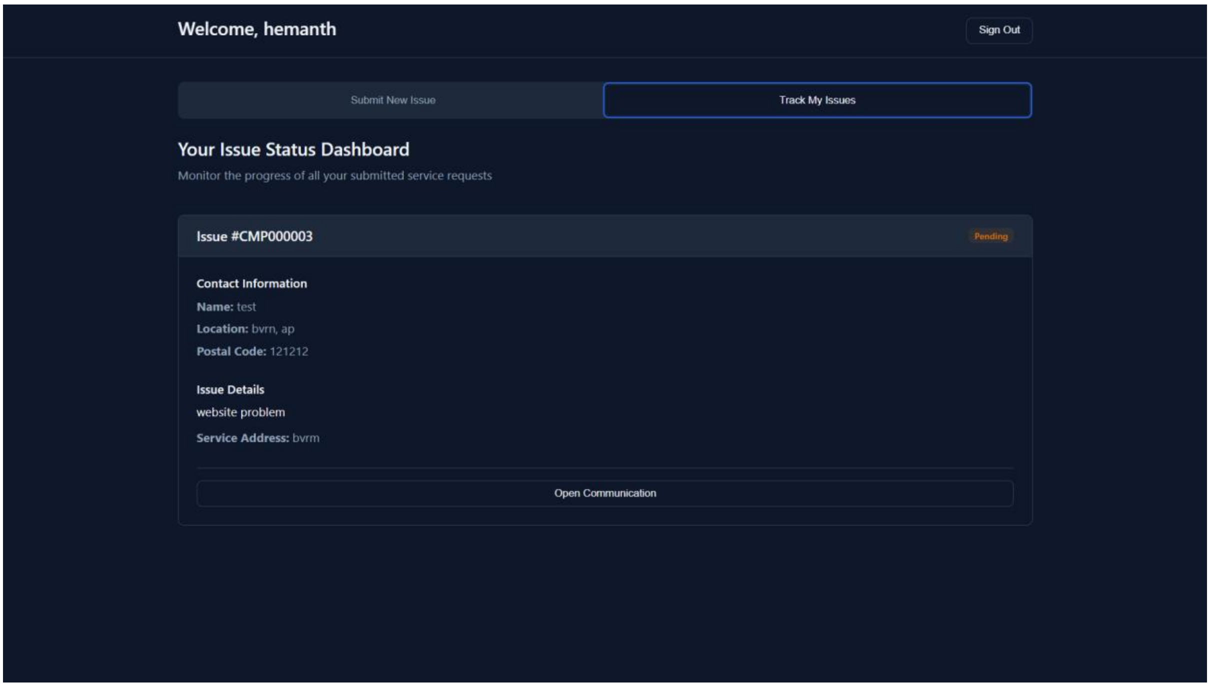


Welcome Back

Please sign in to access your account

Don't have an account? Create one here

- User Complaint Dashboard



Welcome, hemanth

Your Issue Status Dashboard

Monitor the progress of all your submitted service requests

Issue #CMP000003

Contact Information

Name: test

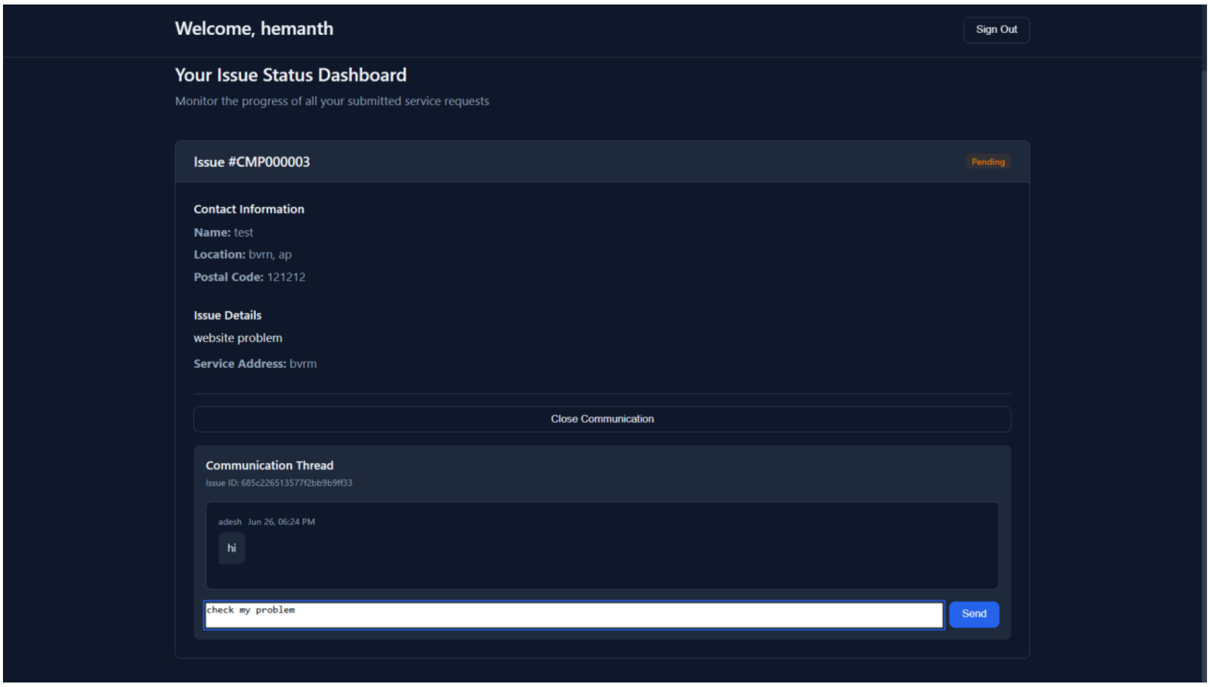
Location: bvrn, ap

Postal Code: 121212

Issue Details

website problem

Service Address: bvrm



Welcome, hemanth

Your Issue Status Dashboard

Monitor the progress of all your submitted service requests

Issue #CMP000003

Contact Information

Name: test

Location: bvrn, ap

Postal Code: 121212

Issue Details

website problem

Service Address: bvrm

Communication Thread

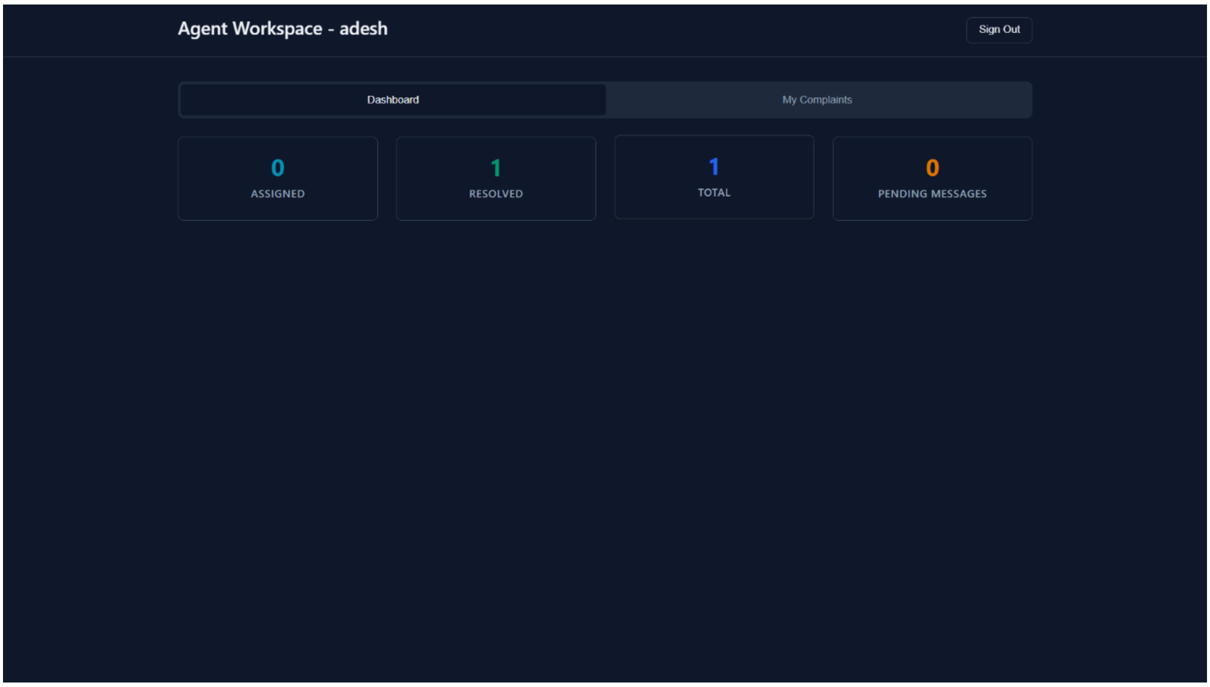
Issue ID: 685c226513577f2bb9b9ff33

adesh Jun 26, 06:24 PM

hi

check my problem

- Agent Complaint View



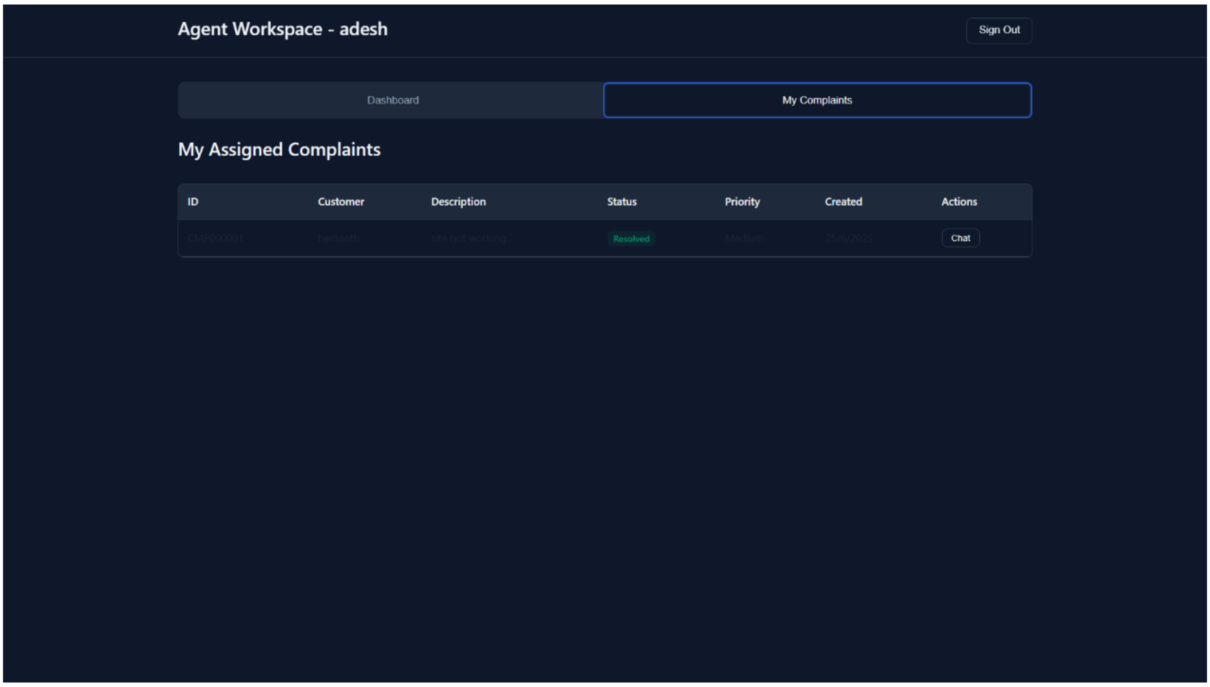
Agent Workspace - adesh

0 ASSIGNED

1 RESOLVED

1 TOTAL

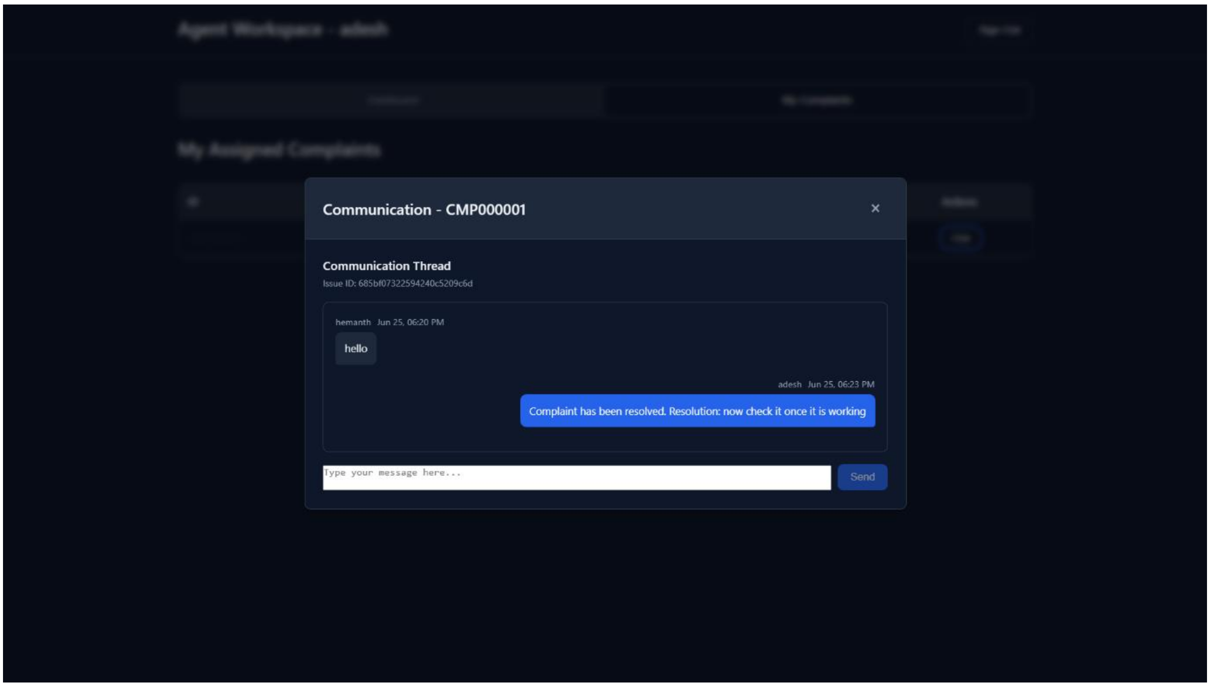
0 PENDING MESSAGES



Agent Workspace - adesh

My Assigned Complaints

ID	Customer	Description	Status	Priority	Created	Actions
CMP000001	hemanth	site not working...	Resolved	Medium	25/6/2025	Chat



Communication - CMP000001

Communication Thread

Issue ID: 685bf07322594240c5209c6d

hemanth Jun 25, 06:20 PM

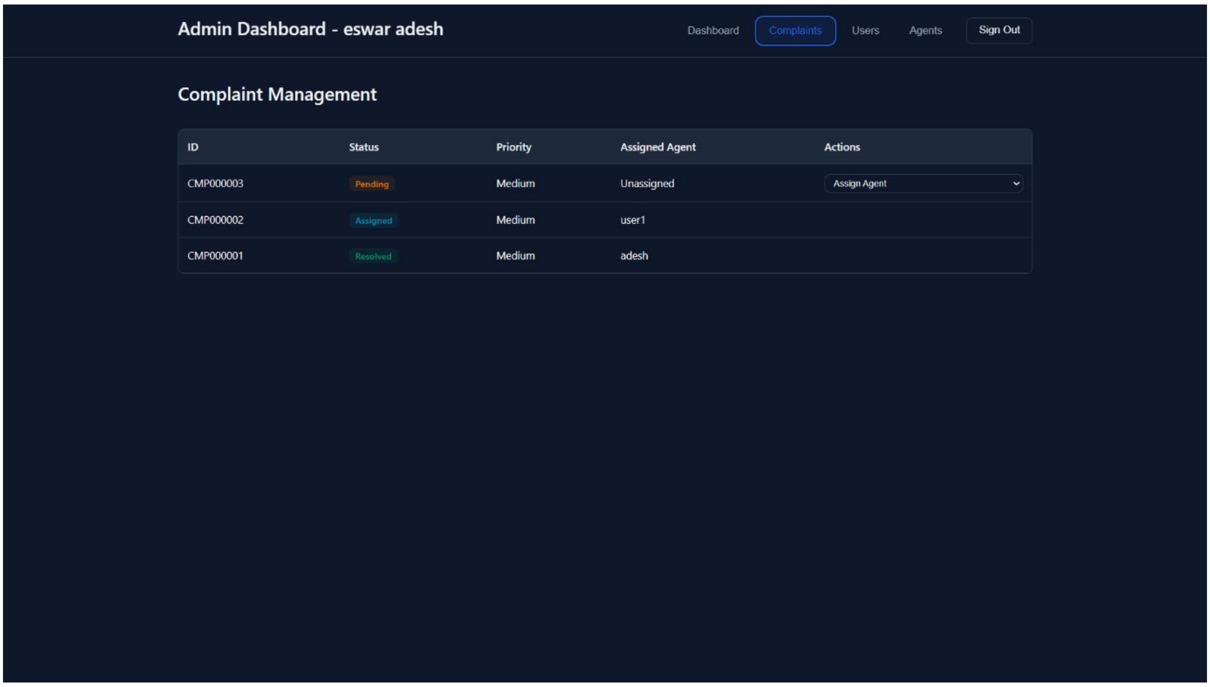
hello

adesh Jun 25, 06:23 PM

Complaint has been resolved. Resolution: now check it once it is working

Type your message here...

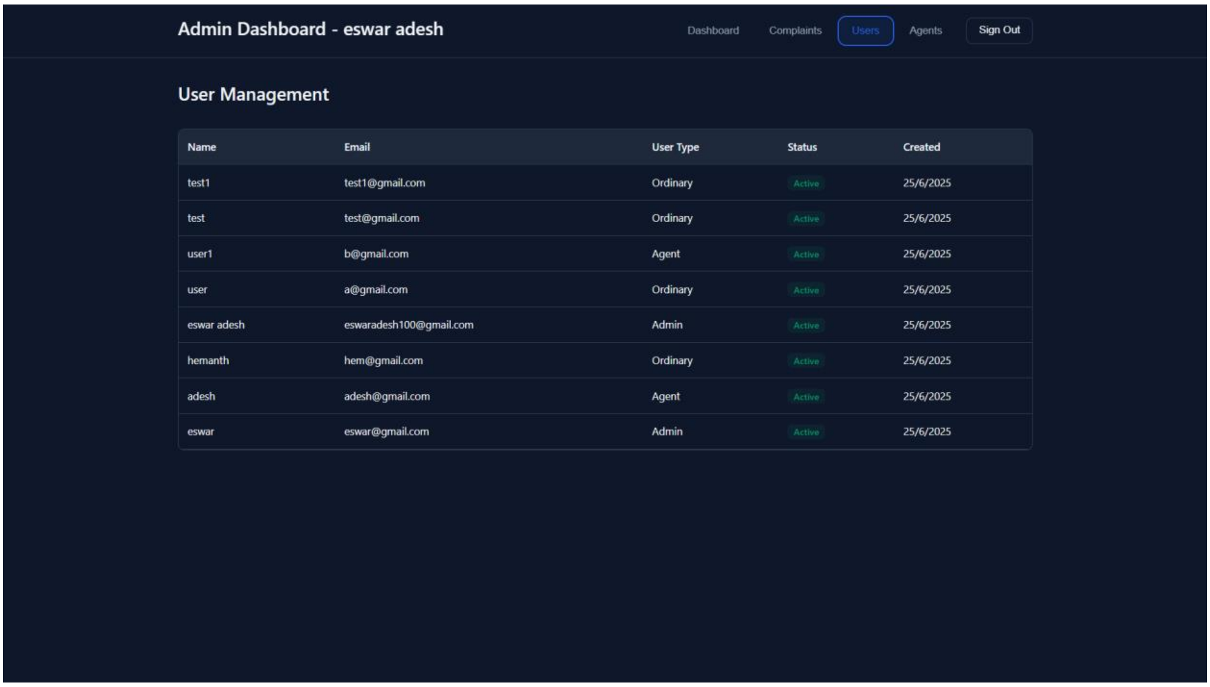
- Admin Dashboard



Admin Dashboard - eswar adesh

Complaint Management

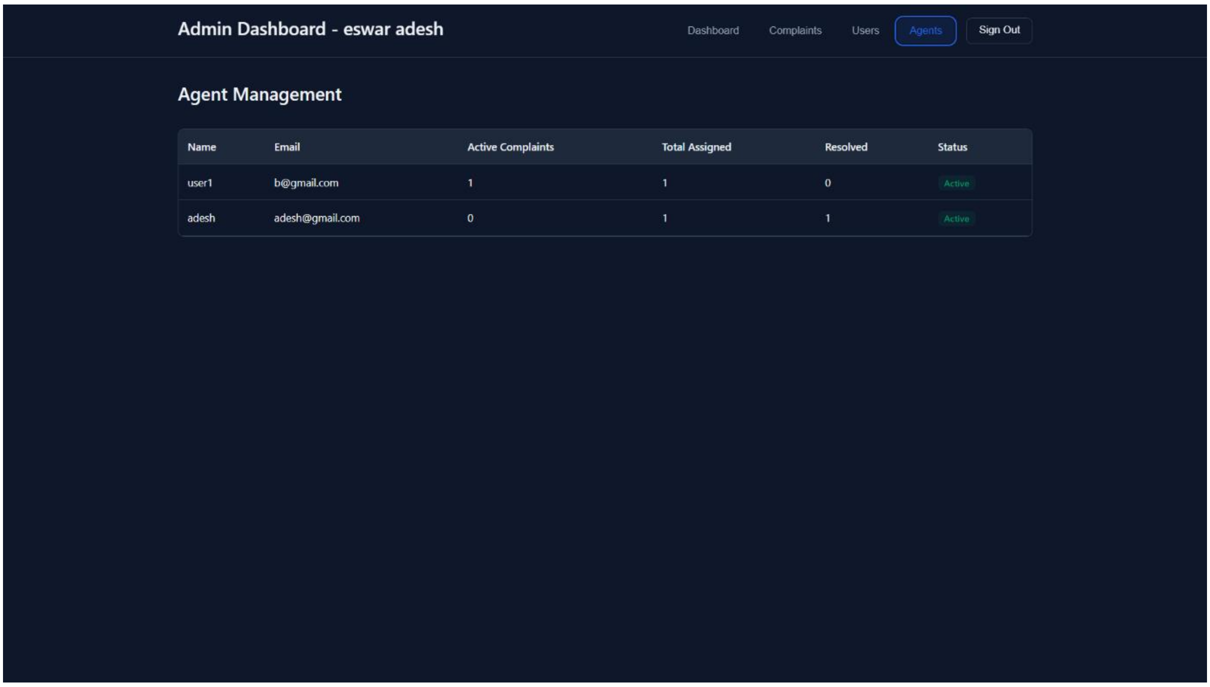
ID	Status	Priority	Assigned Agent	Actions
CMP000003	Pending	Medium	Unassigned	Assign Agent
CMP000002	Assigned	Medium	user1	
CMP000001	Resolved	Medium	adesh	



Admin Dashboard - eswar adesh

User Management

Name	Email	User Type	Status	Created
test1	test1@gmail.com	Ordinary	Active	25/6/2025
test	test@gmail.com	Ordinary	Active	25/6/2025
user1	b@gmail.com	Agent	Active	25/6/2025
user	a@gmail.com	Ordinary	Active	25/6/2025
eswar adesh	eswaradesh100@gmail.com	Admin	Active	25/6/2025
hemanth	hem@gmail.com	Ordinary	Active	25/6/2025
adesh	adesh@gmail.com	Agent	Active	25/6/2025
eswar	eswar@gmail.com	Admin	Active	25/6/2025



Admin Dashboard - eswar adesh

Agent Management

Name	Email	Active Complaints	Total Assigned	Resolved	Status
user1	b@gmail.com	1	1	0	Active
adesh	adesh@gmail.com	0	1	1	Active

7. ADVANTAGES & DISADVANTAGES

Advantages

- Real-time communication with agents
- Centralized dashboard for all roles
- Secure, scalable backend with REST APIs
- Responsive UI across devices
- Modular and maintainable architecture

Disadvantages

- Initial setup requires tech stack familiarity
- MongoDB must be continuously available
- Lacks offline mode and mobile app (yet)

---

8. CONCLUSION

The Service Desk Application meets the core requirements of modern complaint management systems. It empowers users with visibility and agents with accountability. The use of modern web technologies ensures speed, security, and scalability. The project is ready for institutional deployment and further enhancements.

---

9. FUTURE SCOPE

- Email and SMS notifications
- Mobile App (React Native / Flutter)
- Multi-language support
- AI-powered chatbot for complaint intake
- Advanced analytics and charts for admins
- Integration with CRM or ticketing tools

---

10. APPENDIX

Source Code

- **Backend Folder**: `./backend/`
  - Express API, Mongoose Models, JWT Middleware
- **Frontend Folder Source Code**: `./frontend/`
  - React SPA with role-based UI and Axios for API communication

GitHub & Project Demo Link

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