**Project Report: Service Desk for Customer**

**Complaint Resolution**

# INTRODUCTION

## Project Overview

The **Service Desk Complaint Management System** is an integrated platform for users to register complaints, track resolutions, and interact with support agents. It includes customer-facing interfaces (mobile and web), agent dashboards for resolving issues, and an admin dashboard for monitoring overall operations. The system supports user registration, login, complaint submission, live chat, analytics, and user management.

## Purpose

The purpose is to streamline complaint handling, reduce resolution time, and enhance customer satisfaction by providing an organized, transparent, and interactive system that allows users to raise issues and receive real-time support.

# IDEATION PHASE

## Problem Statement

Existing systems lack transparency and responsiveness, causing customer dissatisfaction. There is no centralized mechanism for users to file complaints and get real-time assistance or updates.

## Empathy Map Canvas

* **Says:** “My issue is unresolved.”
* **Thinks:** “Will I get help this time?”
* **Does:** Sends complaints via phone/email
* **Feels:** Frustrated, ignored, helpless

## Brainstorming

* Mobile-friendly interface
* Complaint tracking
* Admin dashboard
* Real-time chat
* Auto-assign to agents
* Analytics for management

# REQUIREMENT ANALYSIS

## Customer Journey Map

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **User Actions** | **Emotions** | **System Interaction** |
| Awareness | Learns about the app | Curious | Loads landing page |
| Registration | Fills registration form | Confident | Registers account |
| Complaint | Files a complaint | Hopeful | Form submitted |
| Resolution | Waits for reply | Anxious | Agent replies |
| Closure | Gets issue resolved | Satisfied | Status: Resolved |

## Solution Requirements

* Functional: Registration, Login, Chat, Complaint form
* Non-Functional: Responsive UI, Data security, Fast performance
* Admin: Assign complaints, Analytics dashboard
* Agents: View & respond to complaints

## Data Flow Diagram

**Level 1 DFD**

User → [Login/Register] → System

System ↔ Database

User → [Submit Complaint] → System → [Assign Agent]

Agent ↔ System ↔ Chat

Admin ↔ Dashboard Analytics

## Technology Stack

* **Frontend:** React.js (web), Flutter (optional mobile)
* **Backend:** Node.js with Express
* **Database:** MongoDB
* **Authentication:** JWT, bcrypt

# PROJECT DESIGN

## Problem Solution Fit

Bridges the gap between customers and support teams through a centralized, transparent, and trackable complaint platform.

## Proposed Solution

A full-stack application where:

* Customers can register, log in, submit complaints, and chat.
* Agents manage assigned tickets and respond.
* Admins monitor and route complaints, manage users, and view analytics.

## Solution Architecture

Client (Web/Mobile)

↓

Backend API (Express.js)

↓

MongoDB (Database)

↓

Authentication Layer (JWT)

↓ Admin/Agent/Customer Dashboards

# PROJECT PLANNING & SCHEDULING

## Project Planning

**Start Date:** 15 June 2025

**End Date:** 7 July 2025

|  |  |  |
| --- | --- | --- |
| **Phase** | **Start Date** | **End Date** |
| Team Formation & Ideation | 15 June 2025 | 17 June 2025 |
| Requirement Gathering & Analysis | 18 June 2025 | 19 June 2025 |
| Design Phase | 20 June 2025 | 24 June 2025 |
| Development Phase | 25 June 2025 | 29 June 2025 |
| Internal Testing (Unit/System) | 30 June 2025 | 4 July 2025 |
| User Acceptance Testing | 5 July 2025 | 7 July 2025 |

# FUNCTIONAL AND PERFORMANCE TESTING

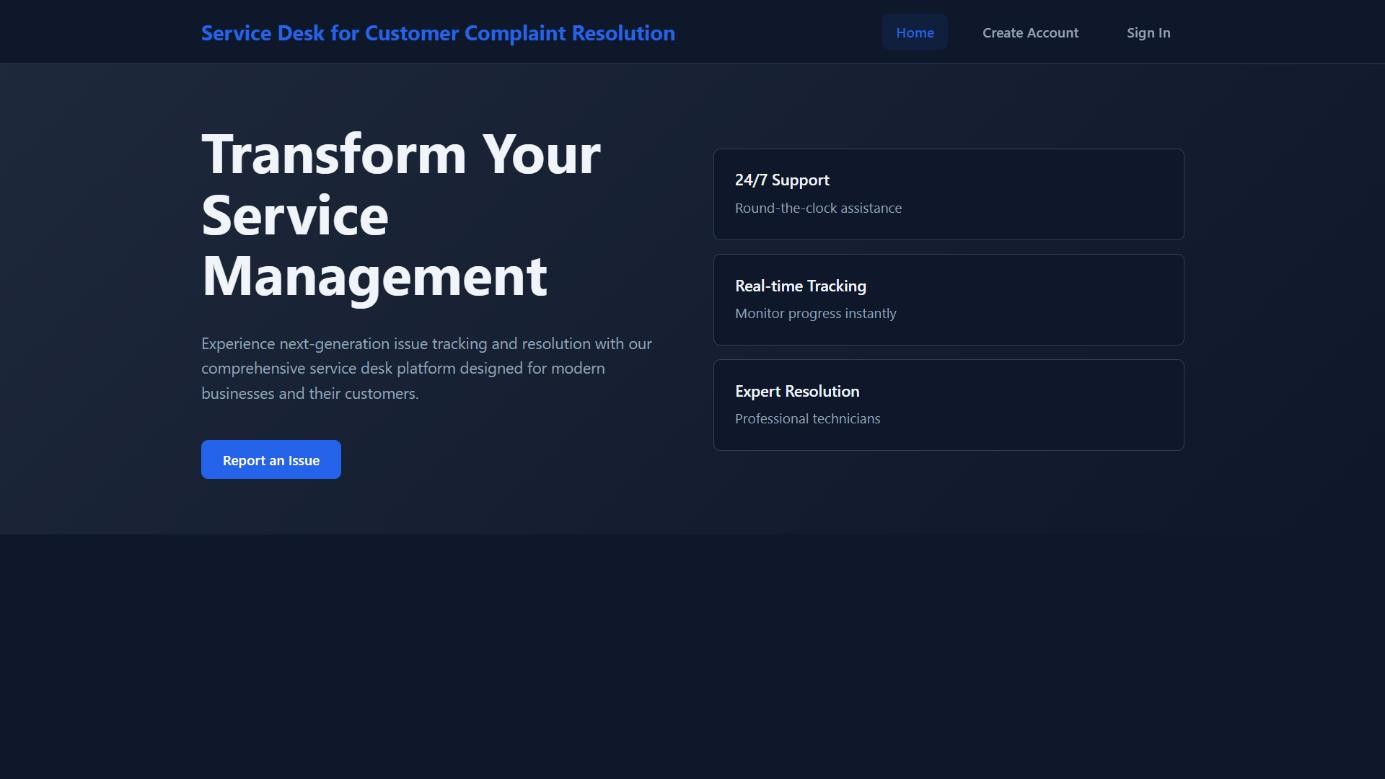
## Performance Testing

* **Tool Used:** Postman (API), Chrome Dev Tools
* **Test Case:** Login response time
* **Observation:** Occasional delay due to OTP (bug ID: BG-001)
* **Recommendation:** Optimize async operations, retry mechanism

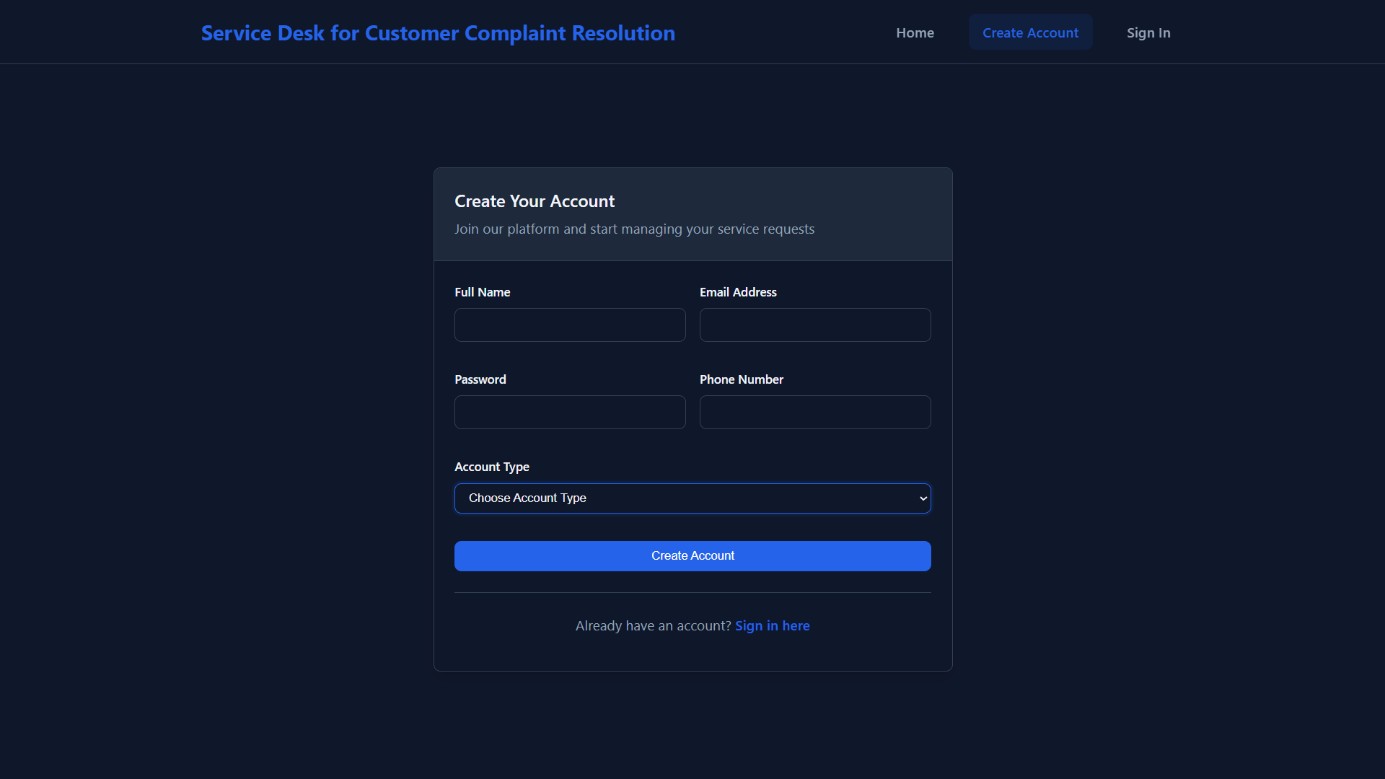
# RESULTS

## Output Screenshots

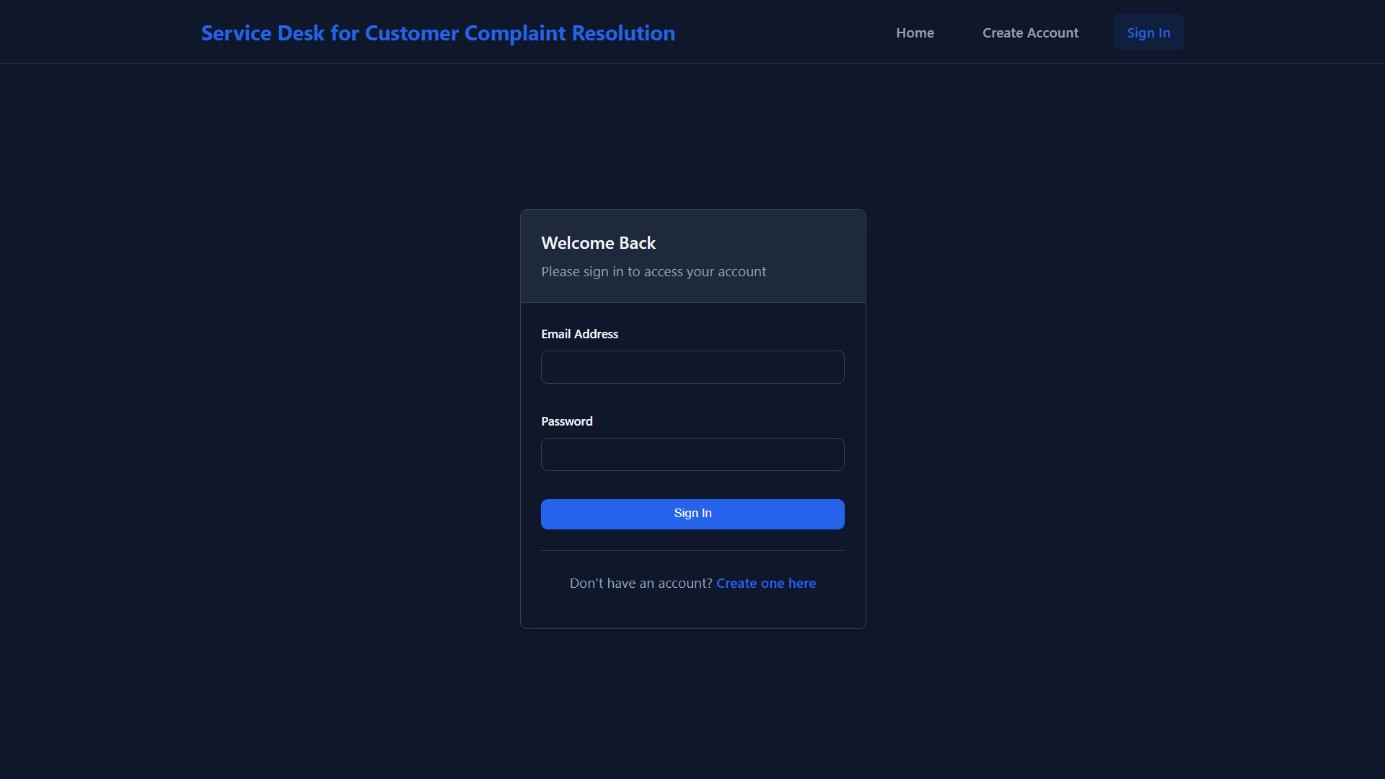
* Landing Page



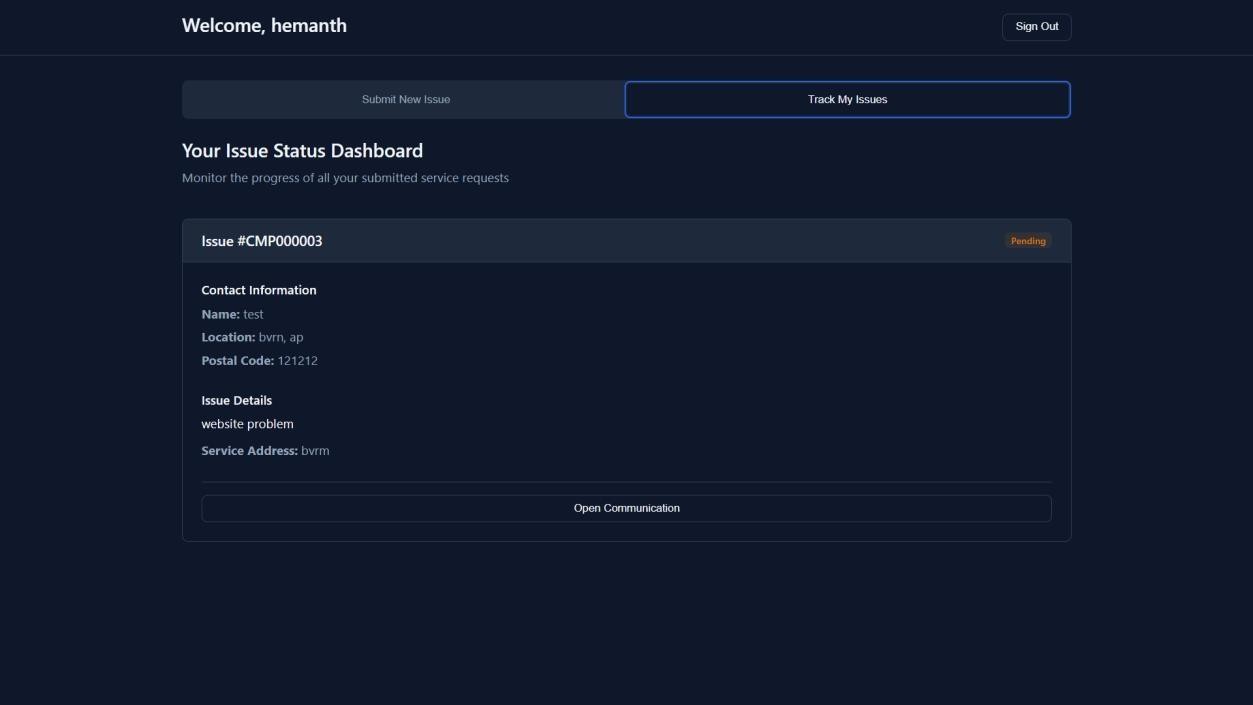
* Signup Page

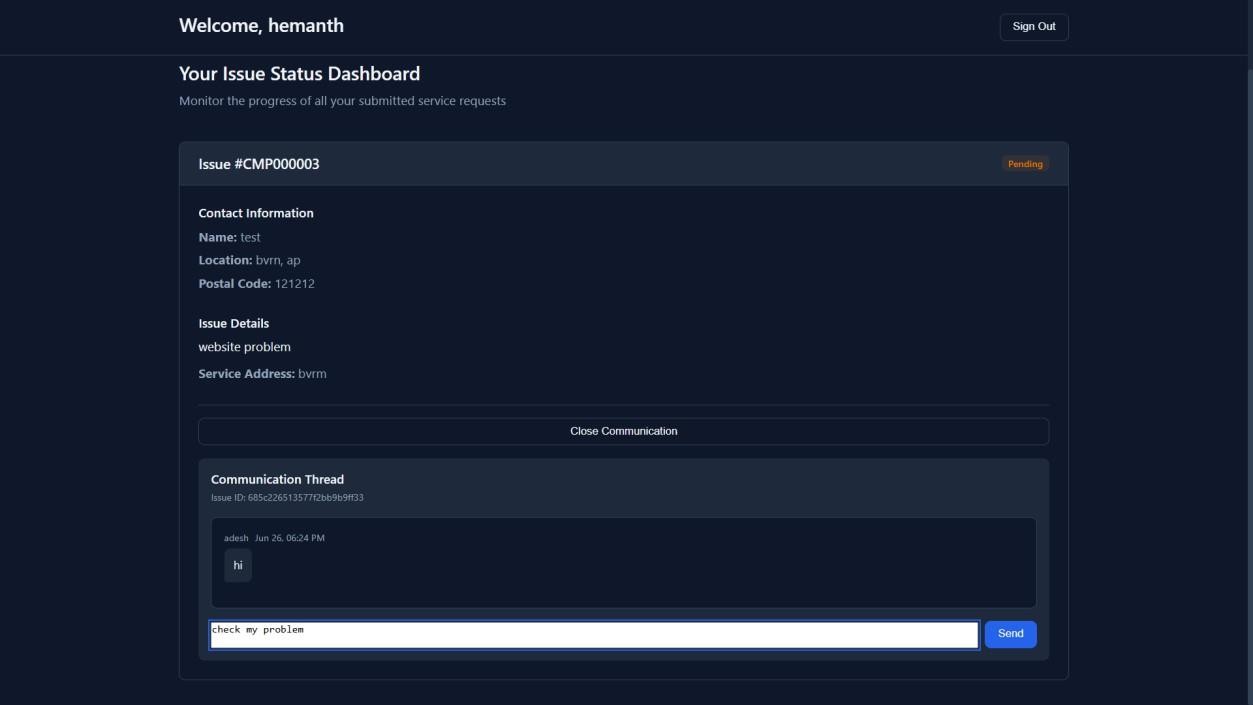


* Login Page

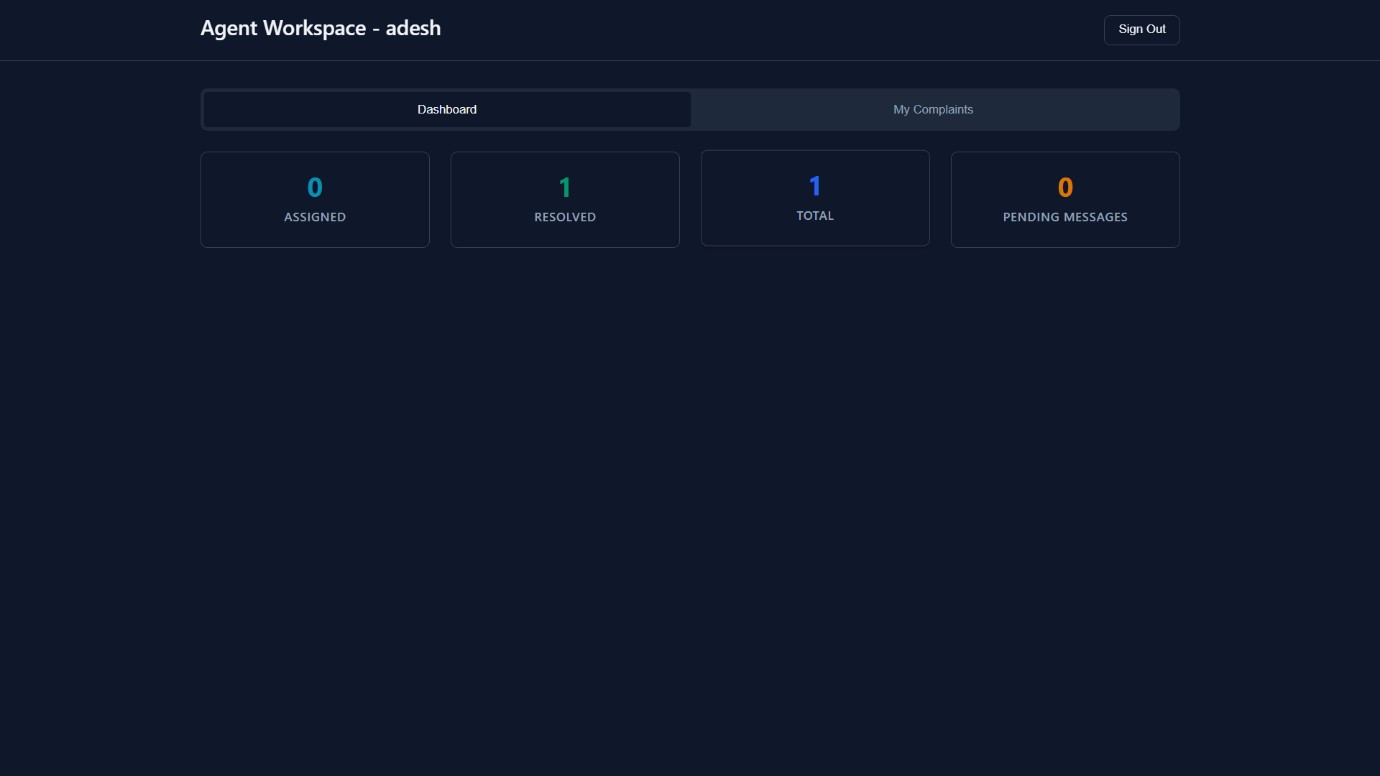


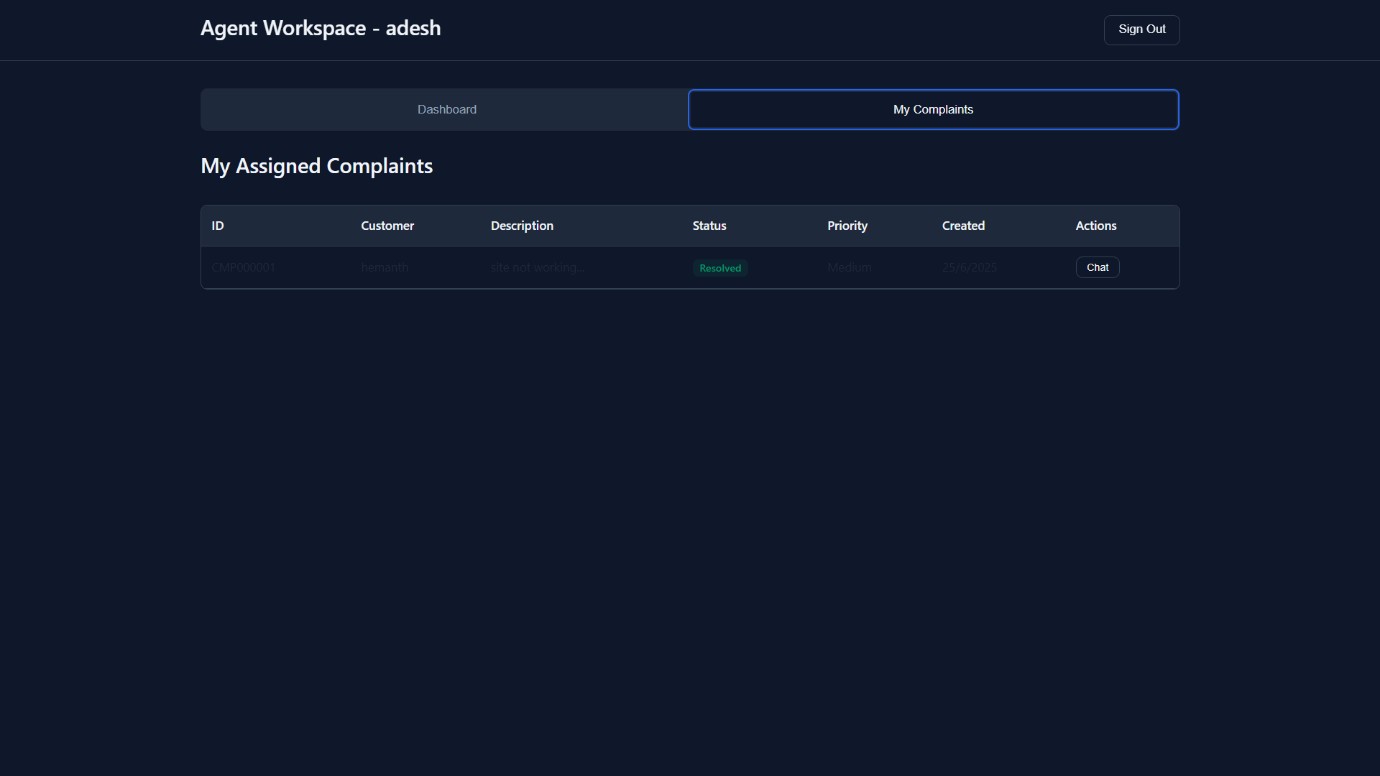
* User Complaint Dashboard

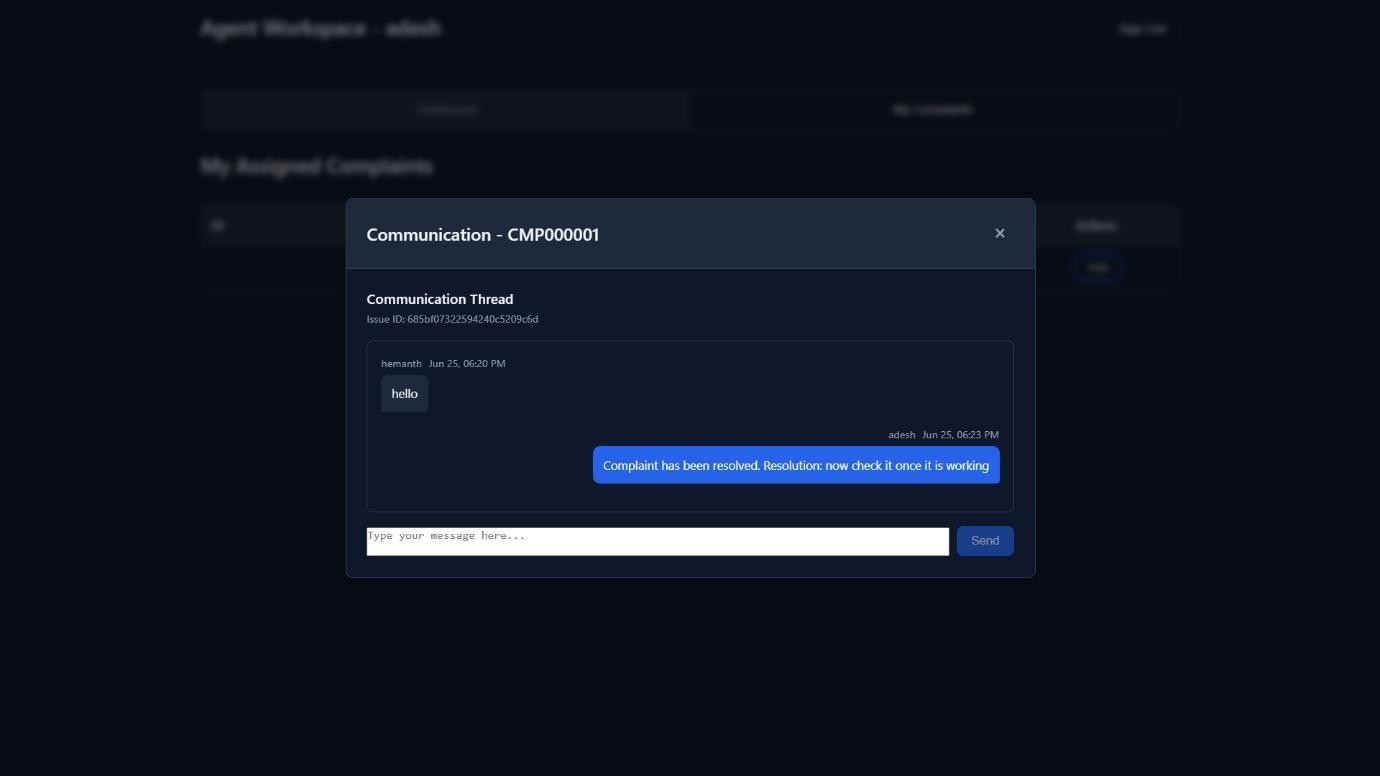




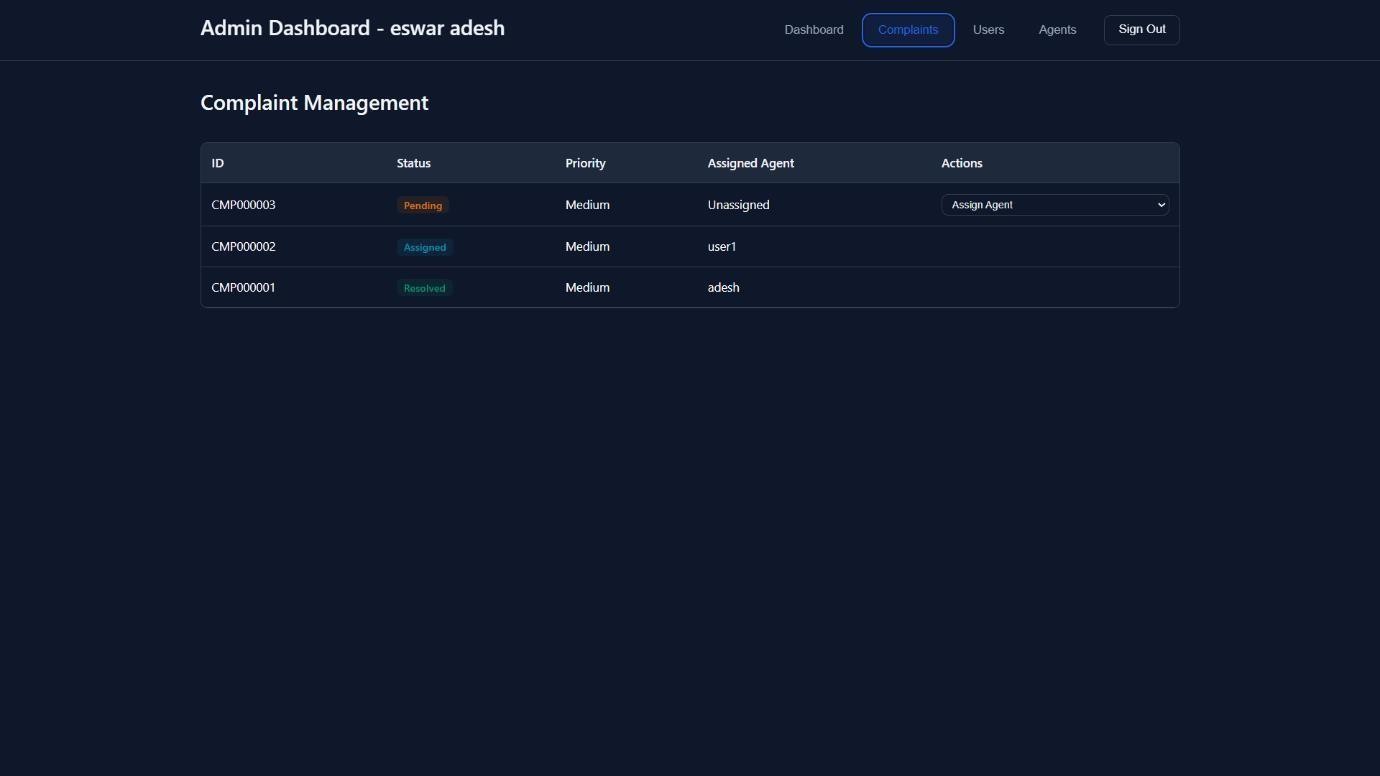
* Agent Complaint View

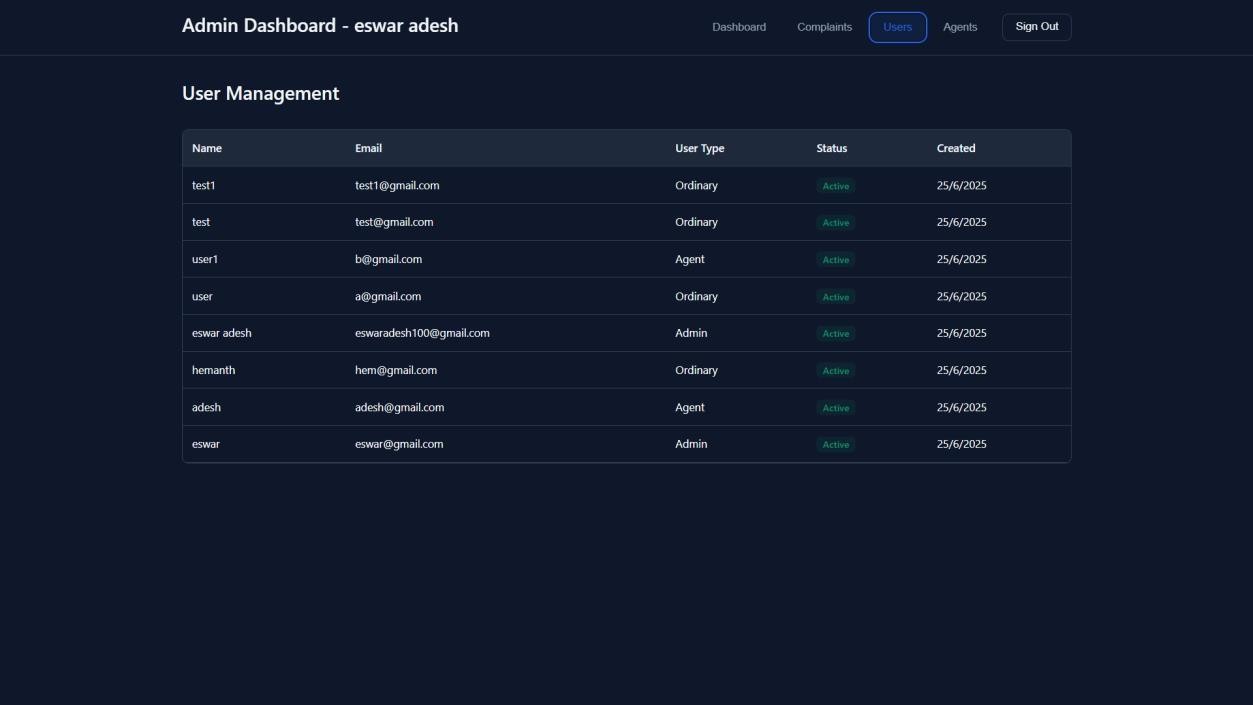






* Admin Dashboard





# ADVANTAGES & DISADVANTAGES

**Advantages:**

* Real-time communication  Complaint transparency
* Role-based access control
* Scalable tech stack

**Disadvantages:**

* Requires stable internet
* May need mobile optimization
* Admin overload if not automated

# CONCLUSION

The Service Desk application meets the objective of offering a structured way for users to raise and track complaints. With added real-time interaction and analytics, it ensures better responsiveness and accountability.

# FUTURE SCOPE

* AI-based complaint classification
* Auto-assignment of agents
* Multilingual support
* Mobile app integration
* Push notifications

# APPENDIX

* **Source Code GitHub Link : https://github.com/ch-santosh/smartbrige-mernstack.git**[](https://github.com/EswarAdeshCh/Service_Desk)

**Demo Video Link:**https://youtu.be/Wx6UCJNLP3w?si=6zZh4w0klMdYFJqd