Project Report

Project Title: ComplaintCare - Online Complaint Management System

Submitted by: [Add Your Team Members Here]

Date: June 2025

# Table of Contents

1. Introduction  
2. Project Overview  
3. Architecture  
4. Setup Instructions  
5. Folder Structure  
6. Running the Application  
7. API Documentation  
8. Authentication  
9. User Interface  
10. Testing  
11. Screenshots or Demo  
12. Known Issues  
13. Future Enhancements

• Project Title: ComplaintCare - Online Complaint Management System

# • Team Members: - Team Leader: Shekh Basir - Team Member: Chitrada Bhargavi - Team Member: Lanka Durga Prasad - Team Member: Gudapati Posukumar • Team ID: LTVIP2025TMID54893

• Project Title: ComplaintCare - Online Complaint Management System

• Team Members: [Add your team member names here]

# 2. Project Overview

• Purpose:

This platform enables users to register complaints, and allows administrators and agents to manage and resolve them efficiently. It is designed to streamline communication between users and the service team.

• Features:

- User registration & login with secure authentication  
- Role-based dashboard (Admin, Agent, User)  
- Complaint submission, tracking, and management  
- Email notifications  
- Real-time status updates

# 3. Architecture

• Frontend: Built with React.js, using Bootstrap and Material UI. Axios is used for API integration.

• Backend: Node.js and Express.js, exposing RESTful APIs for complaint and user management.

• Database: MongoDB with Mongoose for schema modeling.

# 4. Setup Instructions

• Prerequisites: Node.js (v14+), MongoDB, Git

• Installation:

```bash  
git clone https://github.com/yourusername/complaint-system.git  
cd complaint-system  
  
cd client  
npm install  
  
cd ../server  
npm install  
  
# Create .env in server:  
MONGO\_URI=your\_mongodb\_connection\_string  
JWT\_SECRET=your\_jwt\_secret  
PORT=5000  
```

# 5. Folder Structure

• Client:  
/client  
 ├── src  
 │ ├── components/  
 │ ├── pages/  
 │ ├── App.js  
 │ └── index.js

• Server:  
/server  
 ├── controllers/  
 ├── models/  
 ├── routes/  
 ├── middleware/  
 ├── server.js

# 6. Running the Application

• Frontend:  
```bash  
cd client  
npm start  
```

• Backend:  
```bash  
cd server  
npm start  
```

# 7. API Documentation

Sample Endpoints:

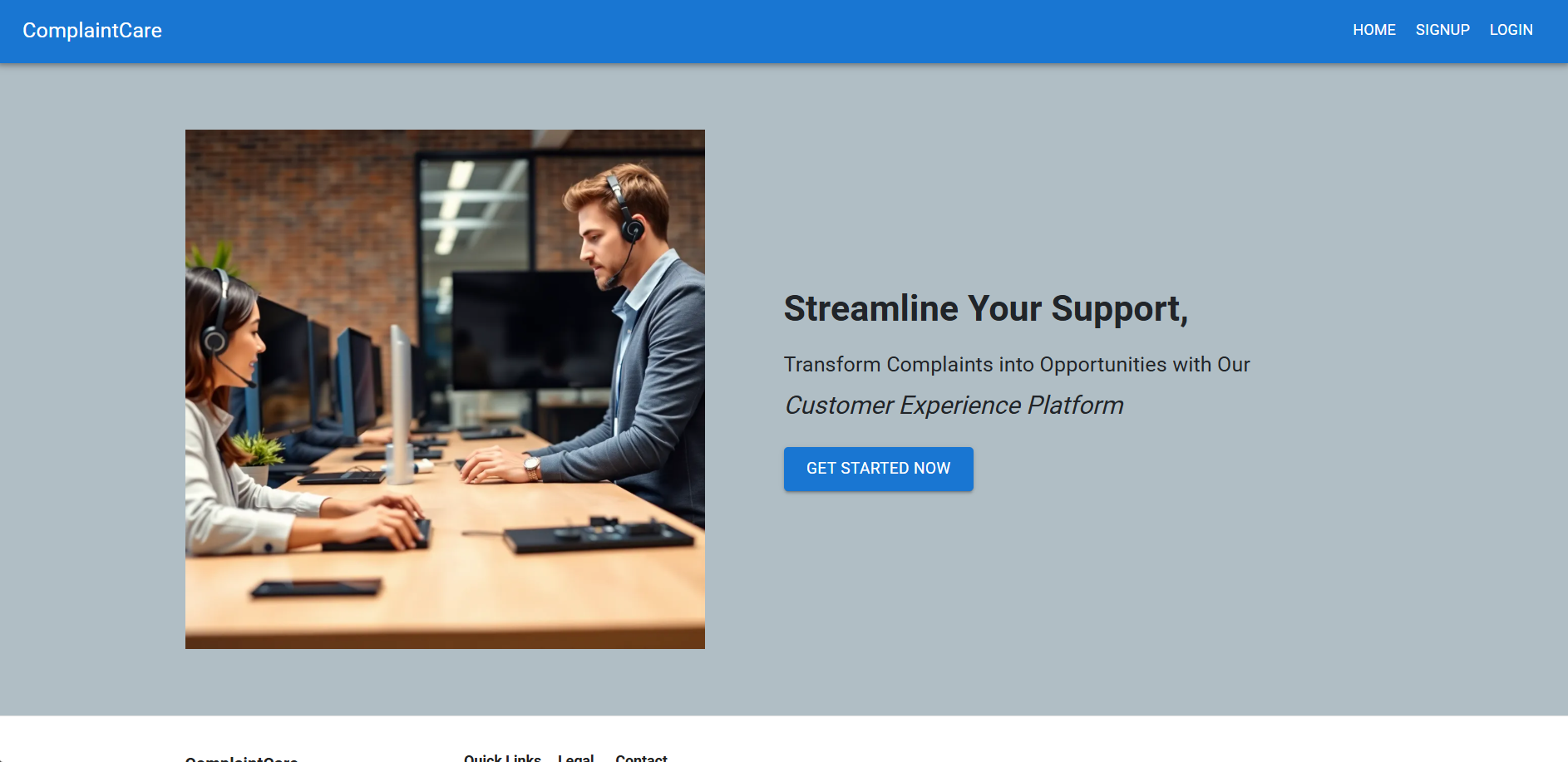
POST /api/users/register - Register a user  
POST /api/users/login - Login  
GET /api/complaints - Fetch complaints  
POST /api/complaints - Submit complaint  
PUT /api/complaints/:id/status - Update status

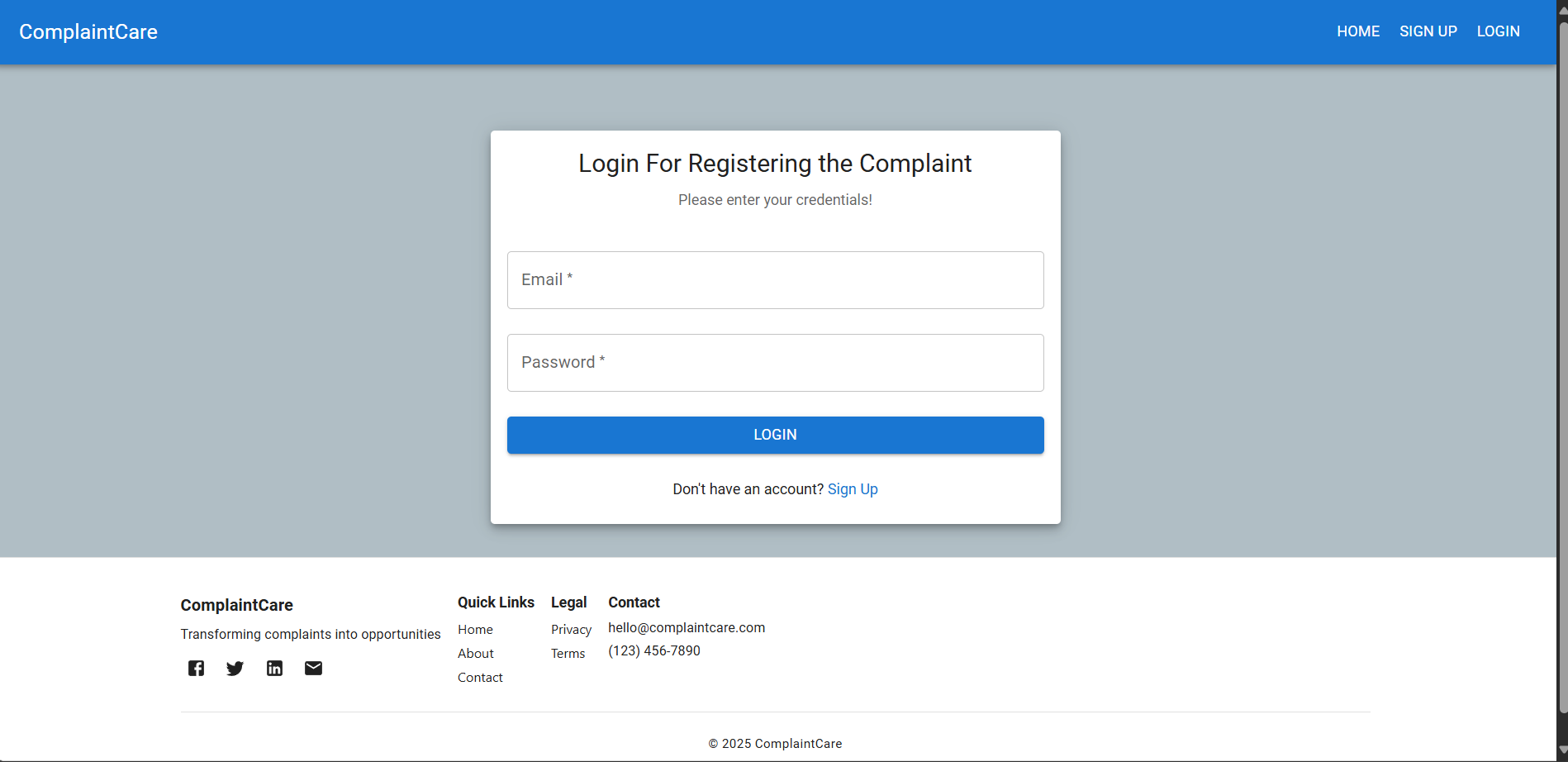
# 8. Authentication

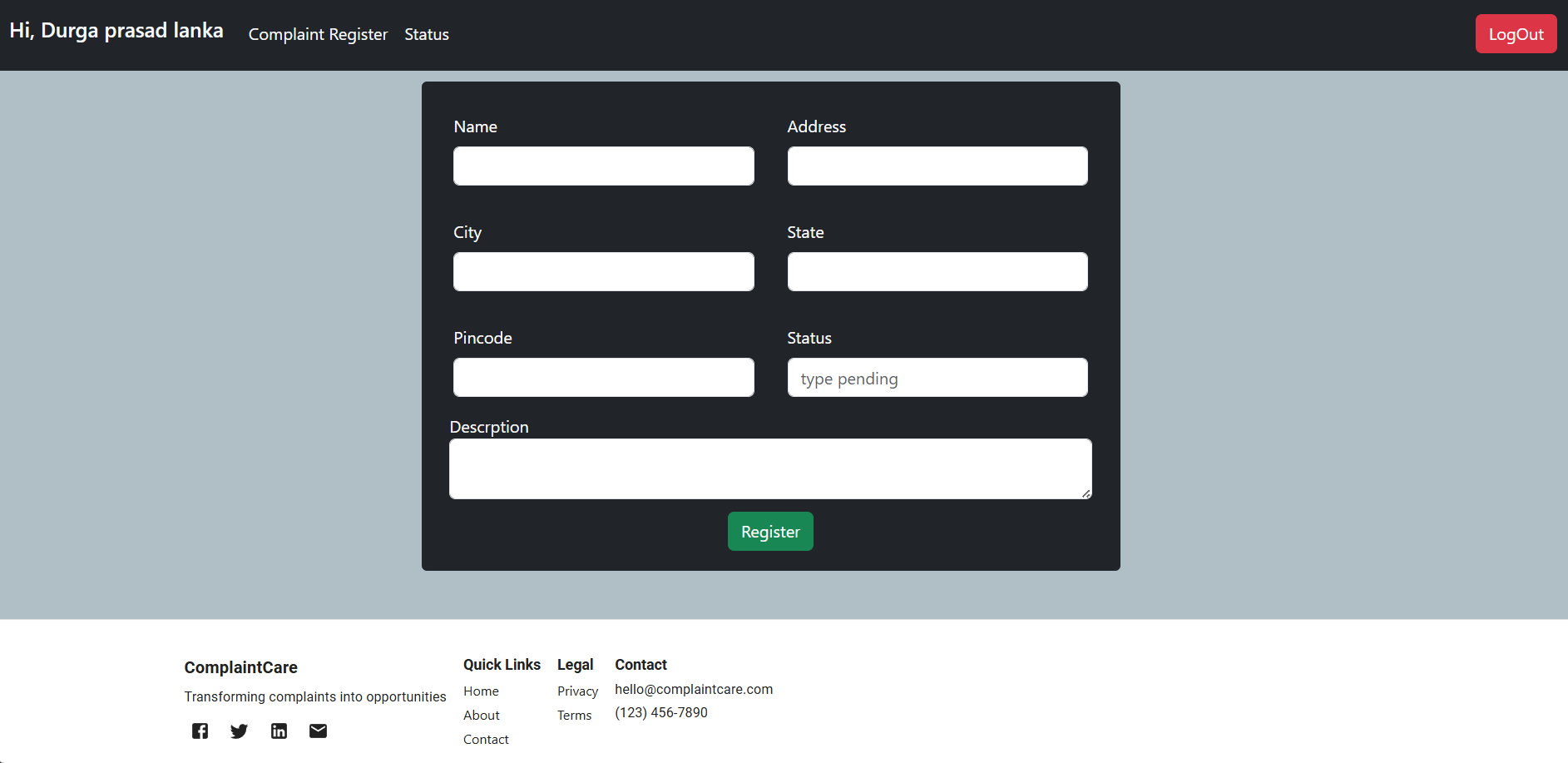
Authentication is handled using JWT tokens. Middleware protects routes and validates user sessions.

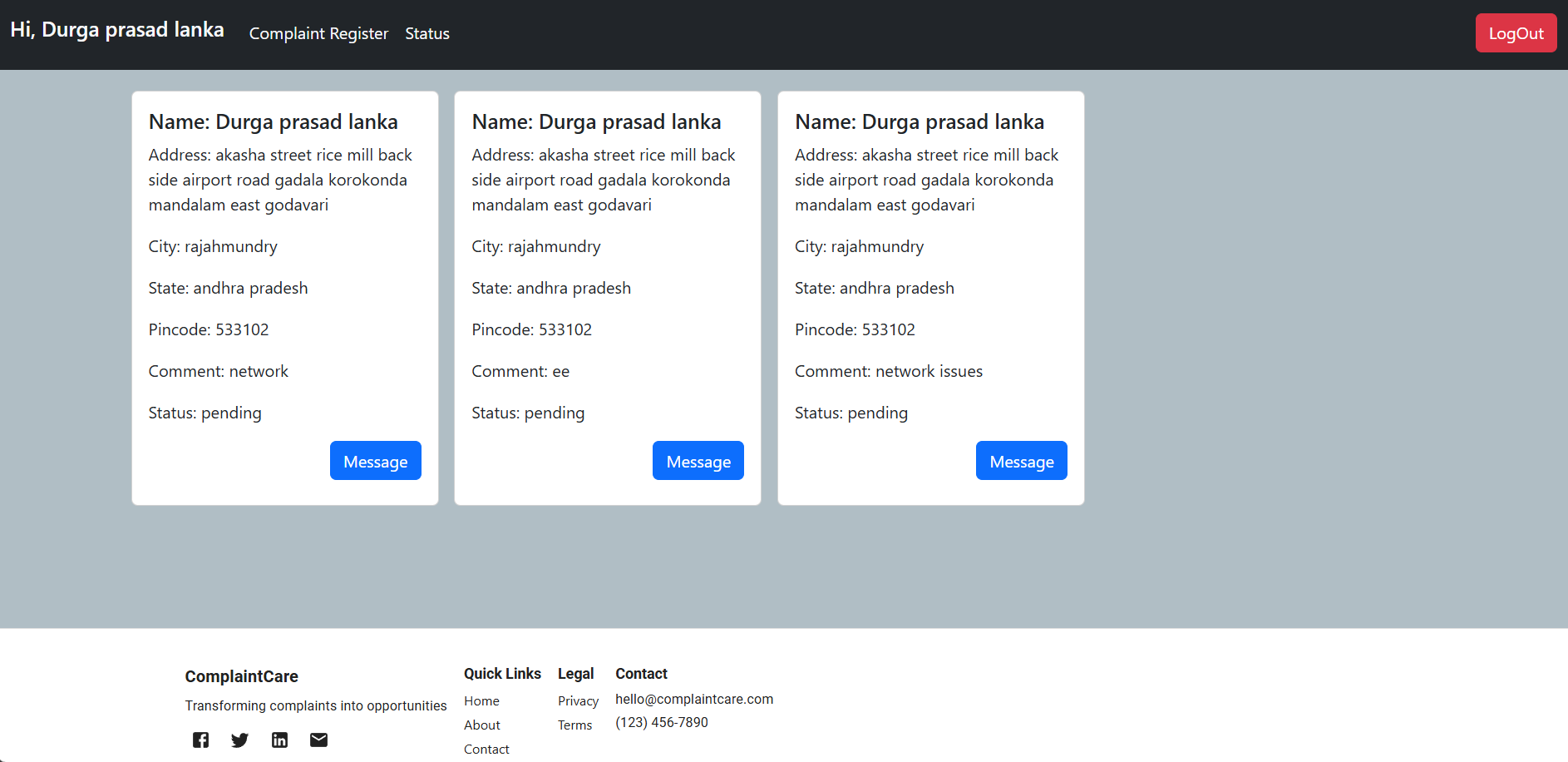
# 9. User Interface

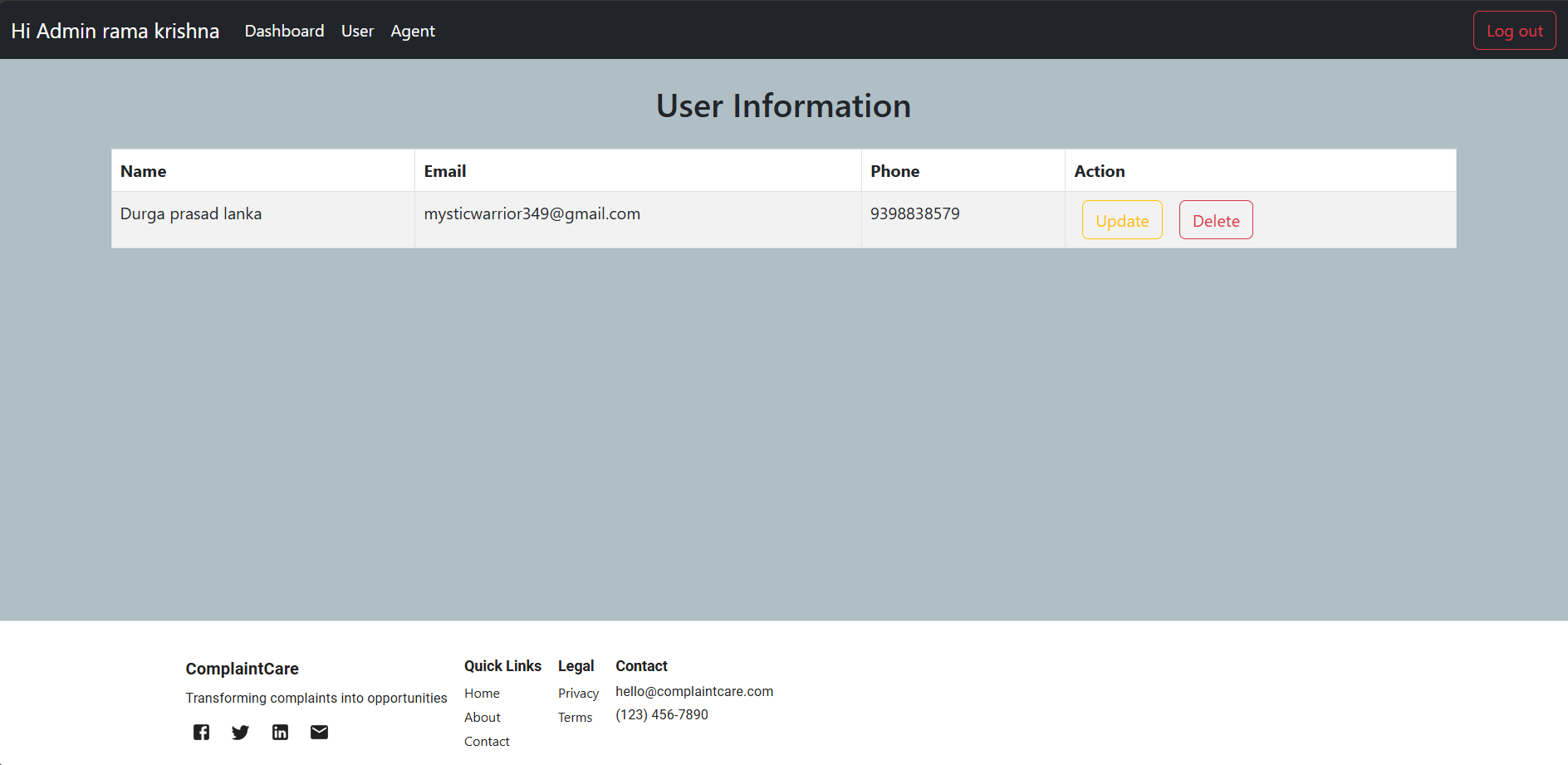
Screenshots of major UI pages:











# 10. Testing

Testing was done manually and using tools like Postman for API testing.

# 11. Screenshots or Demo

Demo link: [Add if hosted]

# 12. Known Issues

- Form reset issues  
- Missing pagination on list view  
- Occasional email delay

# 13. Future Enhancements

- Add file upload support  
- Enable real-time chat  
- Role-based access improvements  
- Push notification integration