Prince Bhatt

Freelancer/Full Stack Developer

Freelancing Services & Pricing Overview

Client's Name: Softox Technologies

Client's Requirement: Mentioned in the Document

LinkedIn: **https://www.linkedin.com/company/softox/**

Total Project Cost: To be Mentioned

The payment for the project will be divided into three phases as follows:

(Will Provide Later)

Client's Declaration;

- I, the client, agree to pay an advance amount before the commencement of the project.
- In case of project cancellation from my side after the start, no Payment will refund.
- I agree to cooperate with the Freelancer during development by responding to queries and providing feedback promptly.

Client's Signature:

……………………

Freelancer's Declaration;

- I, Prince Bhatt, commit to delivering the project professionally and timely, as per the discussed requirements.
- I will keep the client informed about each phase of the development and involve the review and feedback.
- The project will be delivered on time, ensuring adherence to the agreed time line and deadlines.
- All source code, credentials, and documentation will be handed over to the client upon final payment.
- I ensure that the delivered project will be optimized, responsive, and built with clean code practices.

Freelancer's Signature;

……………………..

Contact Information

- **LinkedIn**: https://www.linkedin.com/in/prince-bhatt-0958a725a/
- **GitHub**: https://github.com/princebhatt03
- **Portfolio**: https://princebhatt03.github.io/Portfolio
- **Email**: princebhatt316@gmail.com
- **WhatsApp**: +91-6266307739

Feel free to reach out for collaborations, custom web solutions, or queries regarding project development.

Why Choose Me?

- Proven experience with multiple success fully delivered freelance projects
- Strong skills in MERN stack for full-stack web development
- Reusable backend system for faster and consistent development
- On-time delivery with post-completion support

Complaint Management System (CMS) - Project Proposal

1. Introduction

The Complaint Management System (CMS) is a web-based platform designed to streamline the process of lodging, tracking, and resolving complaints for electronic products such as TVs. The system will be built using the MERN stack and will provide a smooth communication channel between customers, service providers (admins), and the super admin (owner).

2. Objectives

- Allow customers to easily register complaints (tickets/tokens).
- Provide service providers (admins) with a dashboard to view and resolve tickets.
- Ensure complaint resolution is verified via OTP confirmation.
- Enable super admins to monitor all users, admins, tickets, and payments.
- Create a scalable, secure, and user-friendly system.

3. User Roles & Responsibilities

3.1 Customer (User)

- Register a complaint (ticket/token) with product and issue details.
- Receive a unique complaint ID.
- Track complaint status (Pending, In Progress, Resolved, Closed).
- Provide OTP verification upon resolution.
- Receive notifications (SMS/Email).

3.2 Admin (Service Provider/Electrician)

- Login to the admin dashboard.
- View assigned complaints.
- Update ticket status (In Progress → Resolved).
- Trigger OTP verification.
- Mark ticket as Closed after successful OTP validation.

3.3 Super Admin (Owner)

- Full access to all users, admins, and tickets.
- Assign tickets to admins.
- Manage users and admins (create, edit, delete).
- Monitor all complaints and their status.
- Access payment reports and analytics.

4. Key Features

- Secure user authentication (JWT based).
- Complaint registration and tracking.
- Role-based dashboards for User, Admin, and Super Admin.
- OTP-based resolution verification.
- Notifications via SMS/Email.
- Reports and analytics for Super Admin.
- Responsive UI/UX for all devices.

5. Technology Stack

- Frontend: React.js, Tailwind CSS / Bootstrap
- Backend: Node.js, Express.js
- Database: MongoDB (Atlas)
- Authentication: JWT, bcrypt
- OTP/Notifications: Twilio (SMS) / Nodemailer (Email)
- Deployment: Render (Backend), Vercel/Netlify (Frontend), MongoDB Atlas (Database)

6. Database Models (High-level)

1. User → { name, phone, email, password, role: 'user', tickets: [] }
2. Admin → { name, email, password, role: 'admin', assignedTickets: [] }
3. SuperAdmin → { name, email, password, role: 'superadmin' }
4. Ticket → { userId, issueType, description, address, status, assignedAdmin, otp, createdAt, resolvedAt }

7. Implementation Plan

Step 1: Project setup (MERN structure, environment setup).
Step 2: Authentication & role-based access.
Step 3: Ticket creation and management APIs.
Step 4: Ticket assignment (auto/manual).
Step 5: OTP verification flow.
Step 6: Super Admin dashboard.
Step 7: UI/UX implementation.
Step 8: Notifications system.
Step 9: Deployment & Testing.

8. Conclusion

This system will simplify the complaint management process, improve transparency between users and service providers, and provide the super admin with complete control and insights. The MERN stack ensures scalability, performance, and security for future growth and feature enhancements.