**Submitted By**

**Shreyas Sarjerao Hujare**

**Ankush Jalindar Parit**

**1. INTRODUCTION**

**1.1 Project Overview**

The Online Complaint Registration & Management System is a MERN stack-based web application designed to streamline the complaint handling process for institutions, companies, and organizations. The system allows users to submit complaints online, view the status of their complaints, and receive updates. Administrators can log in to a separate dashboard to view all complaints, change their status, and mark them as resolved.

The solution uses React.js for the frontend, Node.js + Express.js for the backend, and MongoDB for the database. Security is ensured through JWT-based authentication and role-based access. The system is accessible via both desktop and mobile devices.

**1.2 Purpose**

The main purpose of this project is to address inefficiencies in traditional, manual complaint handling methods by introducing a centralized, digital platform that:

Improves complaint tracking and resolution times.

Provides transparency between users and administrators.

Ensures secure, reliable, and scalable operations.

**2. IDEATION PHASE**

**2.1 Problem Statement**

Manual complaint management suffers from:

Delays in response time.

No transparency for the complainant.

Risk of data loss or duplication.

High administrative workload.

**2.2 Empathy Map Canvas**

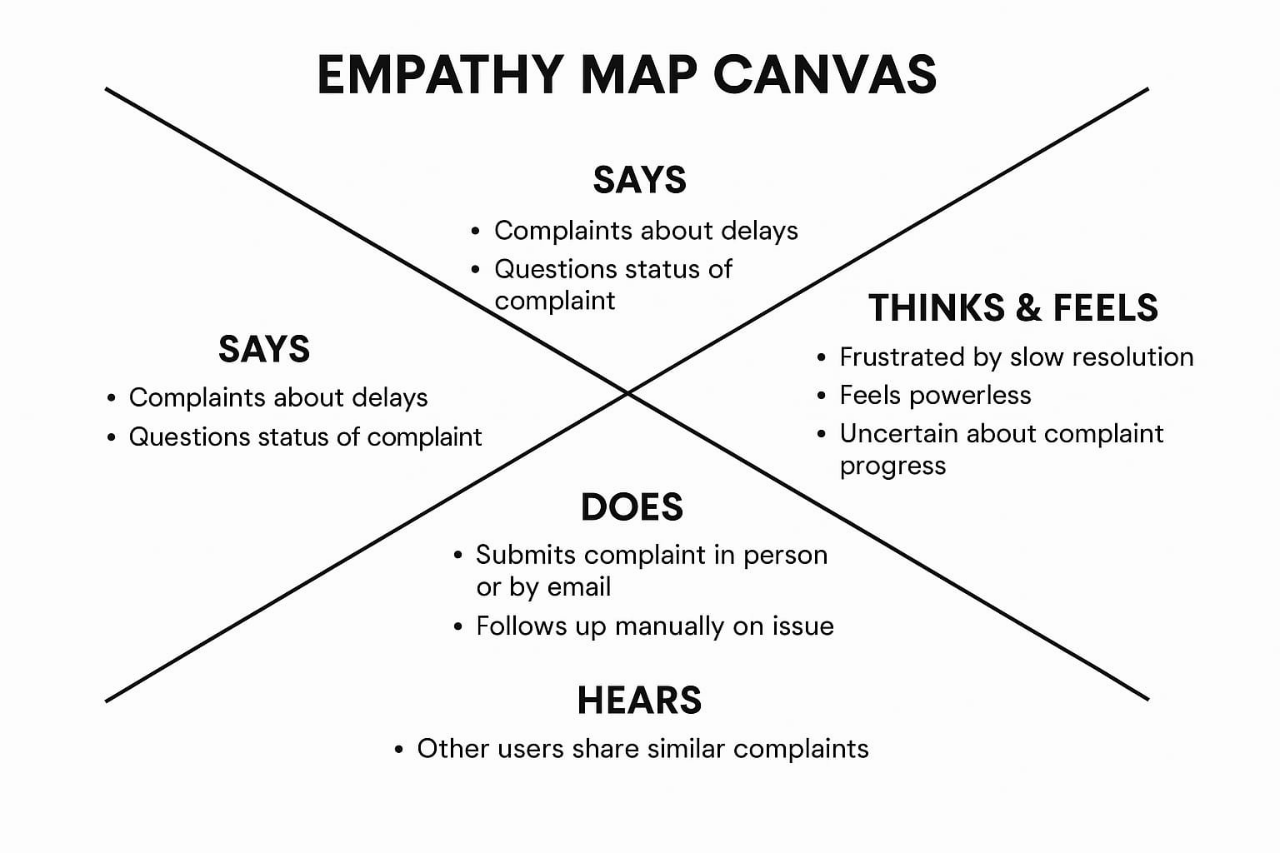
Who? Users (students, employees, citizens) and administrators (college staff, company managers).

What do they see? Slow resolution, no updates, long queues.

What do they hear? Complaints from others about delays.

What do they think/feel? Frustrated, powerless, uncertain about complaint status.

What do they do? Submit complaints in person or by email.



**2.3 Brainstorming**

Ideas generated included:

Mobile app for complaint handling.

Web portal with complaint tracking.

Email/SMS notifications.

Role-based dashboards for better management.

The web-based MERN solution was chosen for its cost-effectiveness and scalability.

---

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

1. User logs into the system.

2. User submits a complaint with details.

3. Admin receives the complaint in the dashboard.

4. Admin updates the complaint status.

5. User receives real-time status updates.

**3.2 Solution Requirement**

Functional Requirements:

User signup/login.

Complaint submission form.

Admin dashboard to manage complaints.

Non-Functional Requirements:

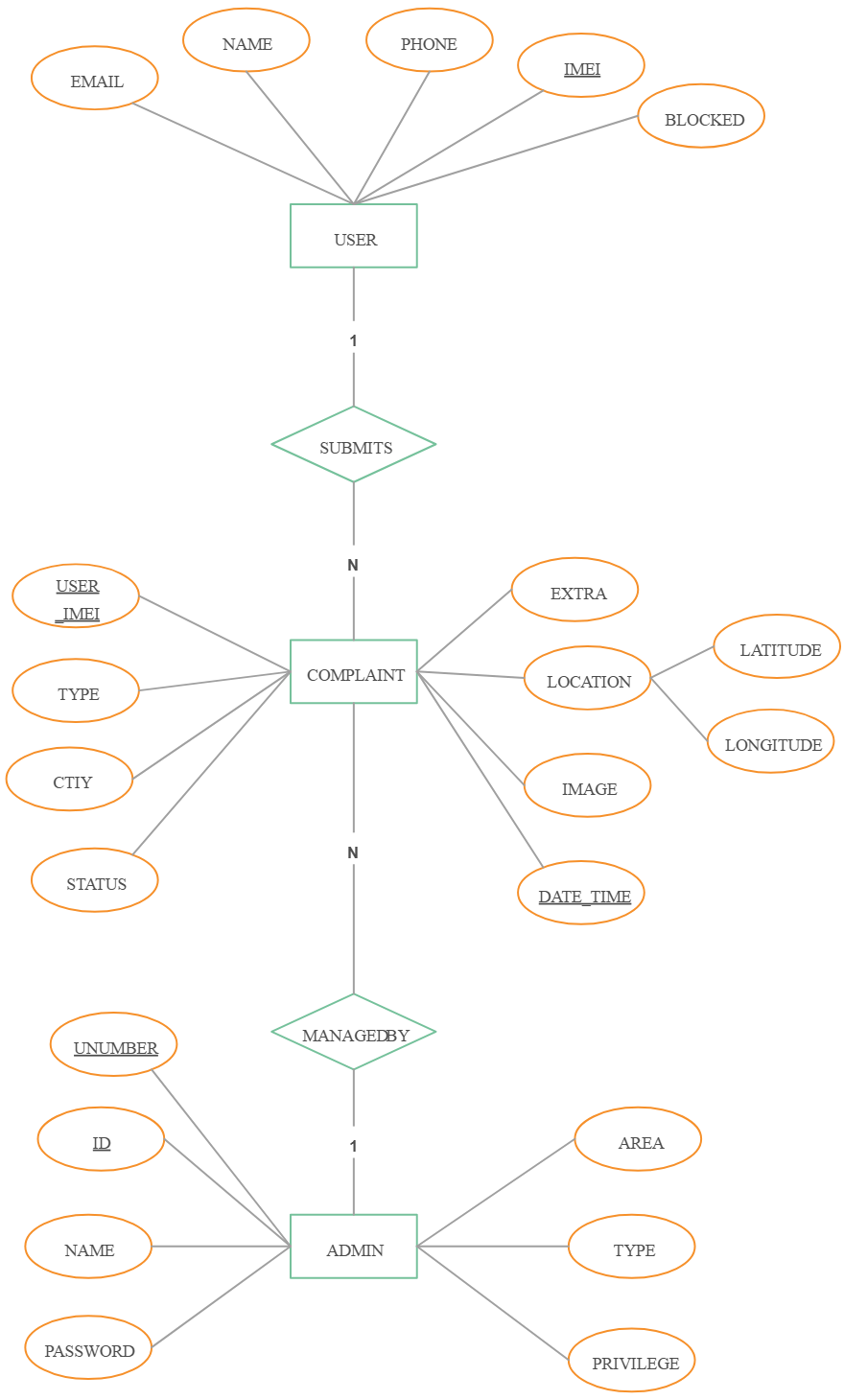
Secure authentication.

Responsive UI.

Fast server responses (<2 sec).

**3.3 Data Flow Diagram (DFD)**

Level 0: User → System → Admin → Resolution.



Level 1: Breaks into login, complaint submission, status update, resolution confirmation.

**3.4 Technology Stack**

Frontend: React.js

Backend: Node.js + Express.js

Database: MongoDB

Authentication: JWT

Styling: TailwindCSS

Hosting: Netlify (frontend), Render (backend)

---

**4. PROJECT DESIGN**

**4.1 Problem Solution Fit**

The chosen web-based solution directly addresses user needs for transparency, faster resolution, and easy access from anywhere.

**4.2 Proposed Solution**

Role-based access for users and admins.

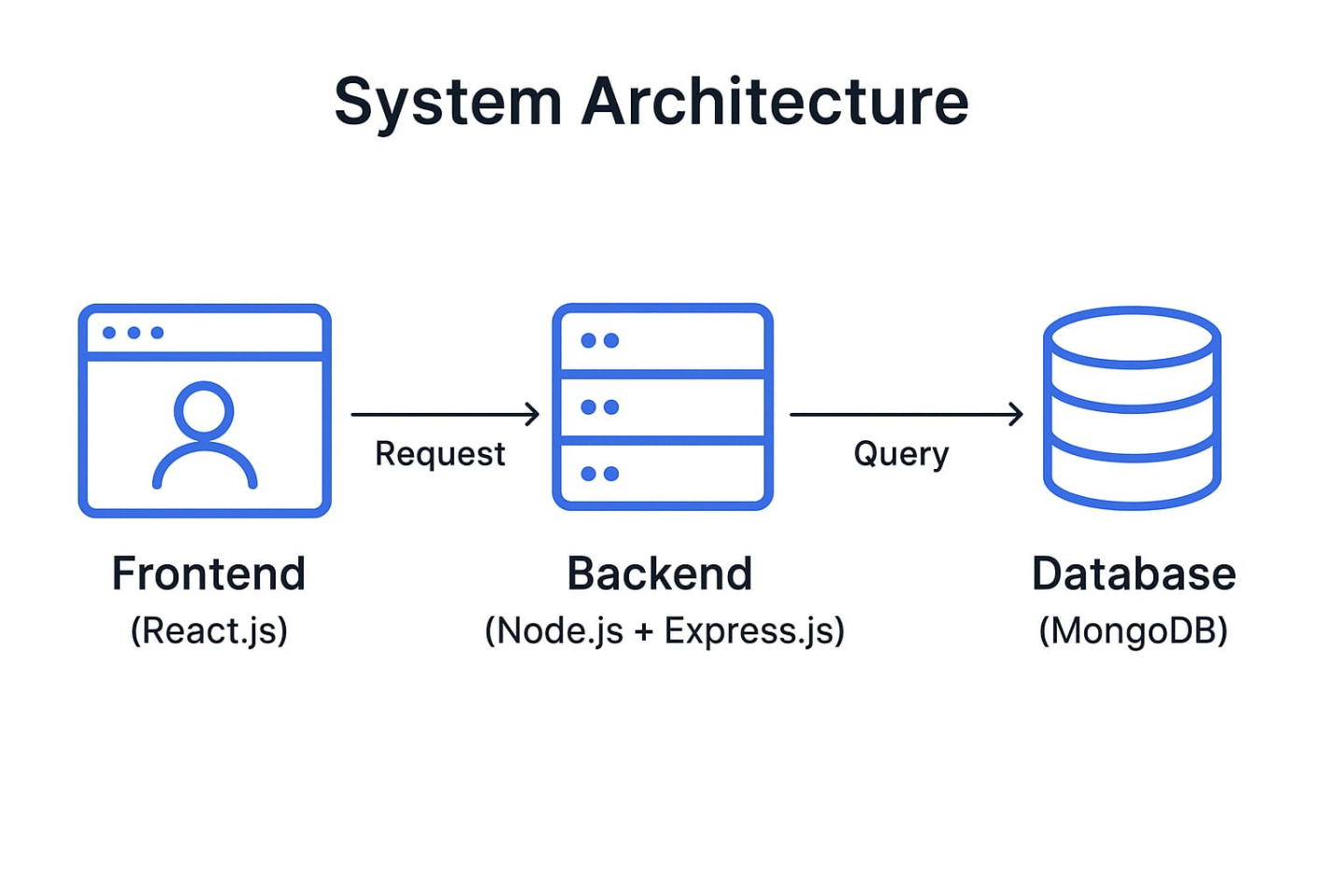
Centralized complaint database.

Real-time status updates.

**4.3 Solution Architecture**

Flow:

User Interface (React) → API Layer (Express) → Database (MongoDB) → Admin Interface.



---

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

Week 1–2: Requirement gathering and planning.

Week 3–4: Frontend and backend setup.

Week 5: Database integration.

Week 6: Authentication module.

Week 7: Admin dashboard.

Week 8: Testing and deployment.

---

**6. FUNCTIONAL AND PERFORMANCE TESTING**

**6.1 Performance Testing**

Tested API endpoints using Postman.

Checked page load times on multiple devices.

Database queries optimized for speed.

---

**7. RESULTS**

**7.1 Output Screenshots**

Login Page

Complaint Submission Form

Complaint List (User)

Admin Dashboard

Complaint Status Update Page

**[ This are Shown in the video]**

---

**8. ADVANTAGES & DISADVANTAGES**

**Advantages:**

Faster complaint resolution.

Transparency for users.

Secure authentication.

**Disadvantages:**

Requires internet access.

Initial setup cost for hosting.

---

**9. CONCLUSION**

The system successfully replaces manual complaint handling with an efficient, transparent, and secure online platform.

---

**10. FUTURE SCOPE**

Mobile app version.

AI-based complaint prioritization.

Integration with email/SMS alerts.

---

**11. APPENDIX**

Source Code: **https://github.com/shreyas2807-hp/COMPLAINT\_MANAGEMENT\_SYSTEM\_USING\_MERN.git**

Project Demo Link: **https://drive.google.com/file/d/1Rd7s9AnrT4kmAybly0PgodEh-\_lBmNCx/view?usp=drivesdk**

Thank You