Project Design Phase
Solution Architecture

Date	23 February 2026
Team ID	LTVIP2026TMIDS90304
Project Name	ONLINE COMPLAINT REGISTRATION AND MANAGEMENT SYSTEM
Maximum Marks	4 Marks

Solution Architecture Overview

The solution architecture for the Online Complaint Registration and Management System is designed to bridge the gap between fragmented manual reporting and a streamlined digital resolution process. It defines the structure, behavior, and requirements of the platform to ensure efficient delivery for all stakeholders (Users, Agents, and Admins).
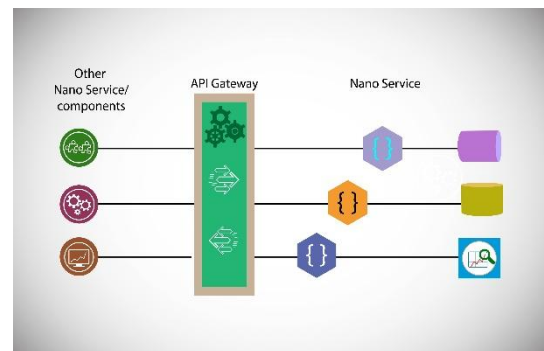
Solution Architecture Diagram

The following diagram illustrates the high-level technical flow and data exchange within the MERN stack environment:

Data Flow & Component Interaction

The architecture facilitates a seamless transition from problem identification to resolution through the following sub-processes:

- **Client Layer (React.js)**: John (the User) interacts with the system through a responsive UI to register, login, and submit complaint details with evidence .
- **Communication Layer (Axios & Socket.io)**: RESTful APIs handle standard data exchanges, while Socket.io enables real-time bidirectional messaging between John and Agent Sarah for prompt issue discussion
- **Server Layer (Node.js/Express.js)**: The backend processes business logic, including authentication via JWT/sessions and the intelligent routing of complaints to appropriate agents
- **Data Persistence Layer (MongoDB/Mongoose)**: Complaint data, user profiles, and chat histories are stored in scalable NoSQL collections for reliable and quick retrieval
- **Source**: https://encrypted-tbn2.gstatic.com/licensed-image?q=tbn:ANd9GcTdZIpkVrm2w7VoMG9t8wNffv333V79Q1LHL73j9IQeKNDxkZWchI-ZbFRawQGJUJBfIq4zvi4hbfbD3zCoaQnfjToQ7uZfD3-vvKGO4_j2L6rxKmY

Solution Goals Met

- **Best Tech Solution**: Utilizing the MERN stack provides a scalable, modern environment to solve the business problem of inefficient complaint handling.
- **Feature Definition**: The architecture clearly defines role-based access, automated notifications, and built-in messaging as core solution requirements .
- **Delivery Specification**: Provides the roadmap for development milestones, from database schema design to final project implementation