**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 28 June 2025 |
| Team ID | LTVIP2025TMID57433 |
| Project Name | Online Complaint Registration and Management System |
| Maximum Marks | 4 Marks |

🧠 **Solution Architecture: ComplaintCare**

The solution architecture for the **ComplaintCare** project is designed to provide a smart, centralized platform where users can register complaints, track their resolution, and communicate with support agents in real-time. The system ensures transparency, accountability, and faster response times through a clean, role-based interface.

🔹 **Architecture Objectives**

* To allow users to easily register complaints and track their status online.
* To enable admins to assign complaints to agents based on category and availability.
* To support real-time communication between users and agents through an integrated messaging system.
* To provide a user-friendly, responsive interface for users, agents, and admins.
* To ensure secure, scalable, and efficient complaint management using modern web technologies.

| **Component** | **Description** |
| --- | --- |
| **User Interface** | **A web-based UI (using React or HTML/CSS/JS) for users to register, track, and communicate about complaints** |
| **Complaint Submission Module** | **Allows users to file complaints by entering relevant details such as category, description, and optional attachments** |
| **Messaging System** | **Real-time chat system enabling communication between users and assigned agents ( Firebase)** |
| **Status Tracking System** | **Maintains real-time status updates of each complaint (e.g., Pending, In Progress, Resolved).** |
| **User & Complaint Database** | **Stores user profiles, complaint records, agent assignments, and resolution history (using MongoDB/Mongoose).** |
| **Notification System** | **Sends updates to users on complaint status changes or agent responses via email or in-app alerts.** |

🔁 **Architecture Flow**

* User registers or logs in through the frontend UI.
* User submits a complaint via a structured form.
* Complaint data is sent to the backend server and stored in the database.
* Admin reviews complaints and assigns them to available agents.
* Agents update status and communicate with users via chat.
* Users receive real-time updates and can track complaint resolution.

🛠 **Technology Stack**

* **Frontend:** HTML, CSS, Bootstrap, React.js
* **Backend:** Node.js / Express.js
* **Database:** MongoDB (Mongoose)
* **Authentication & Realtime Chat:** JWT, Socket.io or Firebase
* **Deployment:** Vercel / Heroku / Netlify / Render

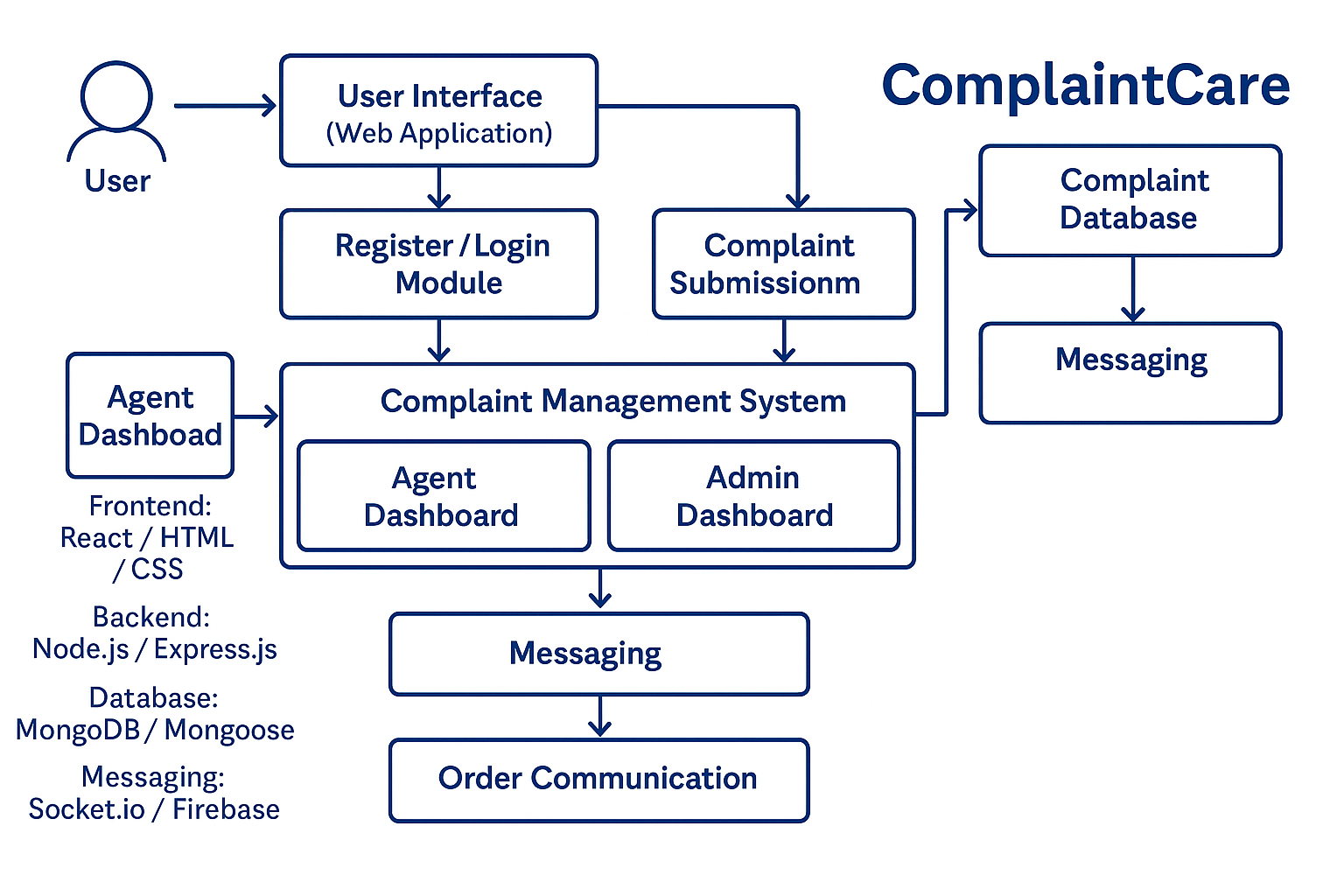
🚀 **Development Phases**

1. User Authentication & Role-based Dashboard Setup
2. Complaint Submission & Storage Integration
3. Admin Panel with Agent Assignment Features
4. Agent Dashboard with Status Update & Chat Support
5. Final UI Polishing, Testing, and Deployment

📈 **Scalability Considerations**

* Modular design allows support for more complaint categories and user roles.
* Real-time chat and complaint tracking can scale with cloud infrastructure.
* Multi-location support possible for large institutions or governments.
* Easily extendable to a mobile app or multilingual interface.

**Solution Architecture Diagram:**

****