Project Design Phase
Solution Architecture

Date	June 2025
Team ID	LTVIP2025TMID57442
Project Name	Learning Online platform
Maximum Marks	4 Marks

Solution Architecture Overview:

The ResolveNow system follows a client-server-based architecture using the MERN stack (MongoDB, Express.js, React.js, Node.js). The architecture bridges user interface requirements with backend logic and database storage, ensuring real-time communication, efficient complaint tracking, and modular code management.

Architecture Layers:

1. Frontend (Client Layer):
   - Technology: React.js with Bootstrap & Material UI
   - Role:
     - User-facing interface for complaint registration and tracking o Role-based access (Admin, User, Agent) o Axios used for API calls o Real-time dark/light theme toggle o Chat UI for interaction with agents
2. Backend (Application Layer):
   - Technology: Node.js with Express.js □ Role:
     - Handles REST API endpoints o Authentication and session management o Complaint routing logic o Connects frontend to database
3. Database (Storage Layer):
   - Technology: MongoDB Atlas (Cloud-hosted NoSQL DB)
   - Role:
     - Stores user profiles, complaints, status updates, chat history, and agent/admin data o Document-based structure for flexibility and scalability
4. Optional Integration:

- Socket.io (For Real-Time Messaging)
- Email/SMS Gateway (For notifications)

Data Flow Overview:

1. User Sign Up/Login: User credentials are sent from frontend to backend, verified, and stored in MongoDB.
2. Complaint Submission: User submits a complaint, which is stored in the database and visible to admins.
3. Assignment: Admin assigns complaint to agent based on workload.
4. Chat: User and agent communicate in real-time (via chat module).
5. Tracking: Complaint status is updated by agents/admins and reflected on user dashboard.

Architecture Diagram:

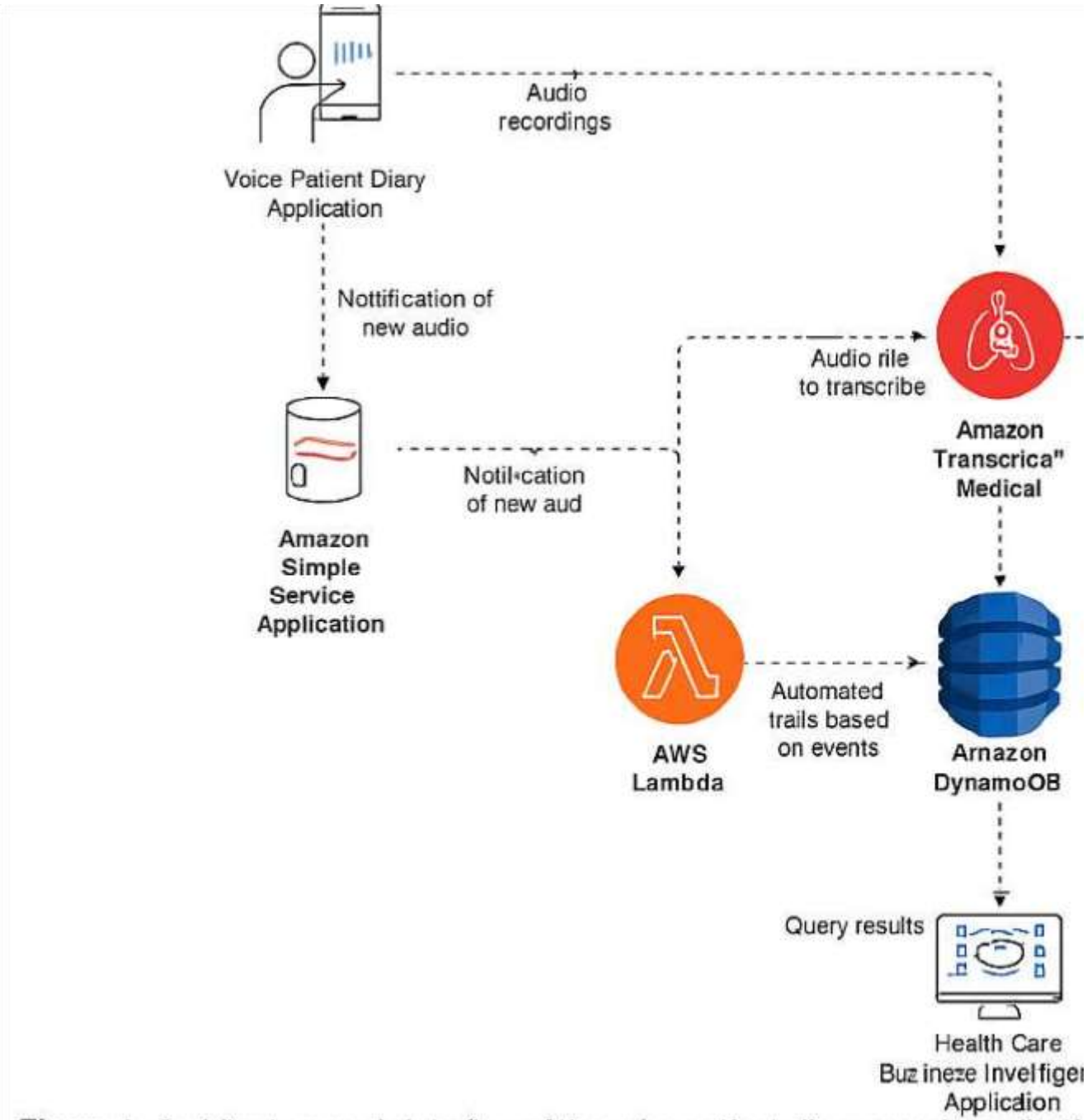

Figure 1: Architecture and data flow of the voice patient-diary sample applicati