CRM

Customer Relationship Management

### **Project Scope for Custom CRM System**

**Objective:** Develop a CRM platform to streamline client and project management across Sales, Support, and IT departments with role-based access and task visibility.

### **Key Features & Modules**

1. **User Management & Roles**
   * Roles: Sales Team, Support Team, IT Team, IT Manager, Admin.
   * Role-based permissions and data access.
   * Employee database with profiles, departments, and activity logs.
2. **Client Management**
   * Sales department can create and manage client records.
   * Fields for client details, requirements, contact information, and project scope.
   * Lead and opportunity tracking.
3. **Project & Work Management**
   * Support team reviews client requests, categorizes project types, and communicates updates with clients.
   * Ability to forward finalized project details to the IT department.
   * Discussion and comment section for client communication.
4. **Task Assignment & Tracking**
   * IT Manager creates tasks and assigns them to IT employees.
   * Task deadlines, status updates, and progress tracking.
   * Internal notes for technical teams.
5. **Role-Based Visibility**
   * Active projects visible to Support, IT Manager, and assigned IT employees.
   * Completed tasks are only visible to IT Manager and Support Team; hidden from other roles.
6. **Notifications & Updates**
   * In-app notifications for updates, task assignment, or client communications.
   * Real-time dashboard for each department.

**Functional Requirements**

## **Functional Requirements for CRM System**

### **1. User and Role Management**

* System must allow admin to create, update, and deactivate user accounts.
* Users must be assigned roles: Sales, Support, IT Employee, IT Manager, Admin.
* Each role must have access only to relevant modules and data.

### **2. Employee Database**

* System must store employee details: name, department, designation, contact, and login credentials.
* IT Manager must see a list of IT employees for task assignment.

### **3. Client Management (Sales Department)**

* Sales team must be able to add new clients with contact details and requirements.
* Sales team must edit and update client information until it’s forwarded to Support.
* Clients’ history and status must be visible to authorized users.

### **4. Project Categorization (Support Department)**

* Support team must view clients added by Sales.
* Support team must categorize the client’s work type or project (e.g.,web, GMB,SEO,Graphics,social, media).
* Support team must communicate with the client, record updates, and collect additional information.
* Support must forward finalized project details to IT department.

### **5. Task Management (IT Department)**

* IT Manager must assign them to IT employees.
* IT employees must update task status (e.g., To-Do, In Progress, Completed).
* IT Manager must monitor overall project progress.

### **6. Visibility & Access Control**

* Active tasks must be visible to Sales, Support, IT Manager, and the assigned IT employee.
* Completed tasks must be hidden from Sales and IT employees but still visible to Support and IT Manager.
* Admin must have full visibility across all data.

### **7. Notifications & Alerts**

* Notifications must be sent when:  
  + Sales adds a new client (to Support).
  + Support forwards project details to IT (to IT Manager).
  + IT Manager assigns tasks (to IT employee).
  + Task updates or completions occur (to relevant stakeholders). **.**

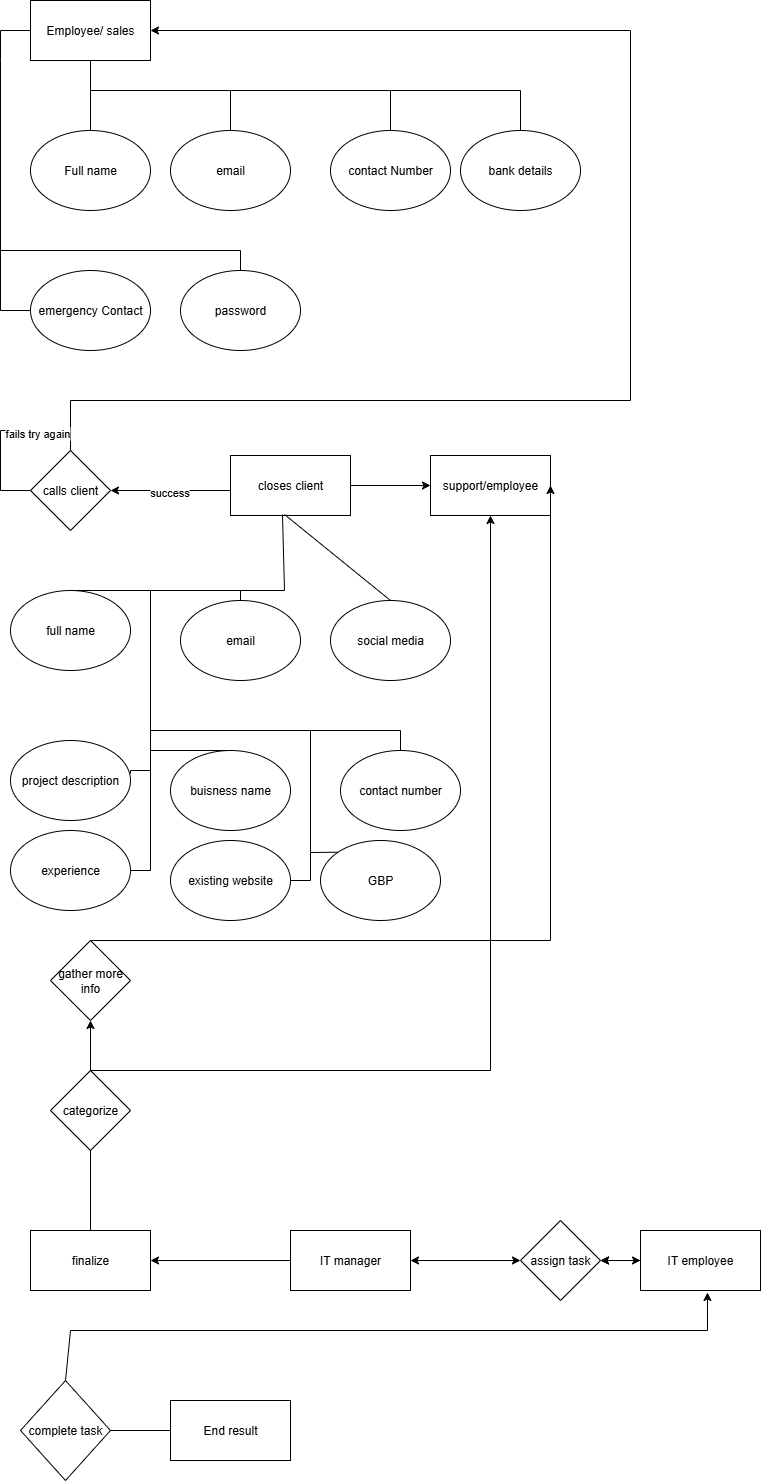
### **8. Reporting & Dashboard**

* System must provide department-wise dashboards:  
  + Sales: Client pipeline and new leads.
  + Support: Active projects, pending clarifications.
  + IT: Task load, deadlines, project statuses.
* Export reports (PDF, Excel) for management.

### **9. Security & Audit**

* All actions (add, edit, delete) must be logged with user ID and timestamp.
* Role-based access control must ensure data privacy.

**ERD:**

****

### **Tech Stack for CRM**

**1. Frontend (UI Layer)**

* Next.js – React-based framework for server-side rendering and routing.
* Tailwind CSS – Utility-first CSS framework for fast and responsive UI design.

**2. Backend (API & Business Logic)**

* Node.js – Runtime environment for executing JavaScript on the server.
* Next.js API Routes – Backend APIs handled within the Next.js app.

**3. Database**

* MongoDB – NoSQL document-oriented database for storing clients, employees, tasks, and logs.

**4. Authentication & Security**

* NextAuth.js – Authentication solution for Next.js with support for credentials

**5. File & Image Storage**

* Firebase Storage – Cloud storage for images and project-related files.