



CRM System

Overview

The **Customer Relationship Management (CRM) system** centralizes **customer interactions, leads, and support tickets**, providing role-based access for **Admin, Sales Rep, Support Agent, and Customer**.

It uses **JWT authentication** for security and **Swagger** for API documentation.

Scope & Roles

- **Admin:** Manage teams, users, customers, and generate reports
- **Sales Rep:** Manage leads, schedule follow-ups, track progress
- **Support Agent:** Handle support tickets and update status
- **Customer:** Raise tickets and track updates

Tech Stack

- **Frontend:** React + Bootstrap (dashboards & UI)
- **Backend:** Spring Boot REST APIs
- **Database:** MySQL (relational data for users, customers, leads, tickets)
- **Authentication:** JWT Security
- **API Documentation:** Swagger

Key Features

- Centralized database for customers, leads, and tickets
- Role-based access control
- Lead and follow-up management for Sales Reps
- Ticket management for Support Agents
- Sales and ticket reporting
- Responsive dashboards and analytics

Sample APIs

Module	Endpoint	Method	Description	Role
Authentication	/api/auth/register	POST	Register new user	All
Authentication	/api/auth/login	POST	Login & generate JWT	All
Customers	/api/customers	GET	List all customers	Admin
Customers	/api/customers/{id}	GET	Get customer details	Admin/Sales
Customers	/api/customers	POST	Add new customer	Admin
Leads	/api/leads	POST	Add new lead	Sales Rep
Leads	/api/leads/{id}	PUT	Update lead status	Sales Rep
Leads	/api/leads	GET	List all leads	Sales Rep/ Admin
Tickets	/api/tickets	POST	Create support ticket	Customer
Tickets	/api/tickets/{id}	PUT	Update ticket status	Support
Tickets	/api/tickets	GET	List all tickets	Support/Admin
Reports	/api/reports/sales	GET	View sales report	Admin
Reports	/api/reports/tickets	GET	View ticket metrics	Admin

Security

- **JWT tokens** for authentication
- **Role-based access control** for API endpoints
- Only authorized roles can access respective modules

Frontend

- **Admin Dashboard:** Users, customers, tickets, reports
- **Sales Rep Dashboard:** Leads, follow-ups, analytics
- **Support Dashboard:** Tickets queue and status updates
- **Customer Dashboard:** Raise tickets and track status
- **Responsive UI:** Built with React + Bootstrap

Database Design (MySQL)

- **Users Table:** Stores user info and role
- **Customers Table:** Customers linked to sales reps
- **Leads Table:** Leads linked to customers and sales reps
- **Tickets Table:** Tickets linked to customers and support agents
- **Reports Table:** Optional for sales/performance metrics

Swagger Integration

- Swagger provides **interactive API documentation**
- Accessible via `http://localhost:8080/swagger-ui.html`
- Allows testing all endpoints without code

Folder Structure

Backend (Spring Boot + JWT + Swagger)

```

crm-backend/
├── controller/           # REST API controllers
                           (UserController, LeadController, TicketController,
                           ReportController)
├── service/              # Business logic classes
├── repository/           # JPA repositories (UserRepository,
                           CustomerRepository, LeadRepository, TicketRepository)
├── model/                # Entity classes (User, Customer,
                           Lead, Ticket, Report)
├── security/             # JWT configuration, filters
├── exception/            # Global exception handling
├── resources/
│   └── application.yml   # Spring Boot configurations

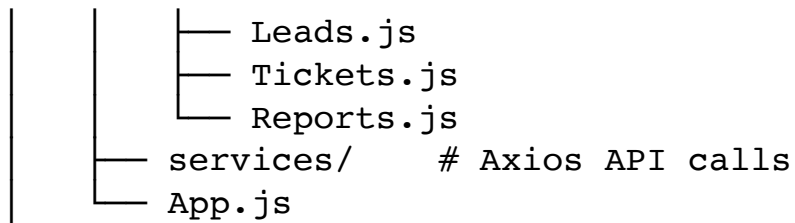
```

Frontend (React + Bootstrap)

```

crm-frontend/
├── src/
│   ├── components/
│   │   ├── Login.js
│   │   ├── Dashboard.js
│   │   └── Customers.js

```



Benefits

- Centralized CRM platform for all roles
- Secure JWT authentication and role-based access
- Structured relational data using MySQL
- Responsive dashboards for Admin, Sales, Support, and Customer
- Real-time tracking of leads, tickets, and reports
- Swagger documentation for easy API testing