Customer Relationship Management (CRM) System

JIDHIN SYAM K

A CRM system is designed to help sales representatives their relationships with customers. It typically involves managing customer data, tracking interactions, managing sales opportunities, and generating dashboards to give insights into customer behaviour and sales performance.

**Features and Functionalities**

* **Customer Management:**
  + Add, edit, delete, and view customer records, including details like name, email, phone, address, and industry.
* **Opportunity Management:**
  + Track sales opportunities and manage their stages, including adding, editing, deleting, and viewing opportunities linked to specific customers.
* **Interaction Management:**
  + Log and track various customer interactions (calls, meetings, emails) and view the history of all interactions associated with specific customers.
* **Contact Management:**
  + Add and view contact details linked to specific customers, including communication preferences and notes.
* **Dashboard and Reporting:**
  + Display key metrics such as the number of customers, active opportunities, sales pipeline stages, recent interactions, and other analytics**.**

**System Architecture**

* **Frontend Layer**
  + Developed using React, the frontend provides a responsive and interactive user interface for managing customers, opportunities, interactions, and visualizing data through dashboards.
  + Components like Customers.js, Opportunities.js, Interactions.js, Dashboard.js, and Contacts.js handle various operations such as adding, editing, deleting, and viewing records.
  + The frontend communicates with the backend via RESTful APIs to fetch data and perform actions.
* **Backend Layer:**
  + Built with Node.js and Express.js, the backend serves as the application’s server, handling API requests from the frontend and interacting with the MongoDB database to perform CRUD operations.
  + Routes such as customers.js, opportunities.js, interactions.js, and contacts.js are designed to handle specific requests related to customers, opportunities, interactions, and contacts.
* **Database Layer:**
  + MongoDB serves as the database to store all data in a NoSQL format, providing flexibility in data schema and structure. Collections such as customers, opportunities, interactions, and contacts store respective data.

**The CRM system is built using the following technologies:**

* **MongoDB**: A NoSQL database used to store customer, interaction, and opportunity data in a flexible document format.
* **Express.js**: A lightweight web application framework for Node.js, used to build the backend API for the CRM system.
* **React**: A JavaScript library for building user interfaces, used to create the frontend of the CRM system.
* **Node.js**: A runtime environment that allows JavaScript to be run on the server side, used to build the backend of the CRM system.

**How to Run the CRM System**

**Backend Setup**

* Install Node.js: Make sure Node.js is installed on your machine.

Clone the Repository: Clone the CRM system repository from your version control system

* Navigate to the Backend Directory: Change into the backend directory.

cd crm-system/backend

* Install Dependencies: Install the required dependencies using npm.

npm install

* Start the Backend Server: Start the server using the following command.

npm start

**Frontend Setup**

* Navigate to the Frontend Directory: Change into the frontend directory.

cd crm-system/frontend

* Start the Frontend Server: Start the frontend server using the following command.

npm start

SCREENHOTS OF CRM SYSTEM









