CREATING A CRM WITH NODEJS

NodeJS as Backend

- Asynchronous, Event-Driven, Non-Blocking Programming language.
- Ideal for building scalable web applications.
- Uses **View Engines** to render dynamic content onto a web page.
- Offers speed in data processing, client-server interaction, development and progress.
- Has the ability to run JavaScript code on the server and to generate dynamic web page content before the page is sent to the user's web browser.

EJS View Engine as Frontend

- EJS is a powerful and flexible templating engine that enhances your NodeJS applications by allowing you **to generate dynamic HTML content with ease**.
- Its simplicity and integration with Express make it an ideal choice for developers looking to build server-rendered web applications quickly.
- **How it Works:** Using EJS, we can define **HTML pages in the EJS syntax** and you specify where various data will go in the page. Then, your app combines data with the template and "renders" a complete HTML page where EJS takes your data and inserts it into the web page according to how you've defined the template.

MongoDB as Database

- **Represents data as a Collection of documents rather than Tables as in MySQL.**
- **Top Choice for NodeJS application development**, as NodeJS seeking a powerful NoSQL database solution i.e. MongoDB.
- NoSQL - non-relational databases that **uses a non-tabular format to store data**, rather than in rule-based, relational tables like relational databases.
- Stores data in BSON format (similar to **JSON format**). NodeJS also process datas in JSON format. Hence, they work well together.
- MongoDB's document-oriented structure, built on a flexible schema design, allows developers to store data in JSON-like documents.

- **Mongoose** - a JavaScript object-oriented programming library that **creates a connection between MongoDB and the NodeJS JavaScript runtime environment**. It provides a straightforward, schema-based solution to model application data.
- **Databases can be download** using **mongoexport** Tool. This tool will allow exporting data from a MongoDB Collection **in JSON Format.**

MySQL as Database

- Can be used with NodeJS.
- **Node-mysql** is the npm library developed to use MySQL with NodeJS.
- Can be an opt solution if we are in need to get data from multiple tables (relational data)
- if we are using MySQL, then we may also have to face some problems in fronted development as we might have to convert relational data into json in node.js. So, it is best is keep it simple with json support database like MongoDB