**Node JS**

**Creating Rest Full Web Service using Express JS Module**

Get method : Get the resource like employee, customer, order, products

In json format

Get method : search data or records using pk property

1. Query param
2. Path param

Post method : create the resource like store or insert the data in file system or database system.

Put or Patch methods : these method is use to update the existing resource property. Put is update to update whole resource. Patch is use to update partial property.

Delete method : it is use to delete the resource using pk

MVC Mode View Controller

MVT

db -🡪 folder

dbConfig.js : data base configuration

model -🡪

LoginModel.js -🡪 which connect to database table

repository -🡪

LoginRepository.js -🡪 database code to do SignIn and SignUp functionality

Service -🡪

LoginService.js -🡪 you can write service conditions.

Controller -🡪

LoginController.js

--🡪 responsible to receive the value from client and pass the value to service layer and base upon response redirect to particular view or send data or response.

Routing -🡪 folder

It contains all routing details.

App.js

Main entry of the application.

Database : it is use to store the data in property format.

DBMS : Database Management System

RDBMS : Relational database management system

Oracle :

Oracle database not available.

**create database databasename;**

**use databasename;**

**show databases;**

**show tables;**

MERN Stack or MEAN Stack

Mongo DB/MySQL Express JS React / Angular Node JS

Asp.net

SQL Server 2018, 2022 etc

IBM Java with Db2

Oracle/MySQL Java technologies.

Python with Django : SQLite

Mongo Db or MySQL

Limitation of SQL Database.

Schema base

Employee

Id Name Salary

Number varchar float phonenumber city

1 Steven 45000 null null

2 Lex 34000 8448575757 null

3 Lena 23000 null Bangalore

Sid SName Age technologies

1 Steven 34 C,C++,Python,Java

No SQL Database.

\_id,name,age,project:

Prompt message for github copilot for generating login\_app.

Provide new workspace for login application with emailid,password and type of user as admin and customer. Need client and server folder separately. With frontend as react js, backend express js with following model layer, controller layer, service layer, routing and database is mongo db with JWT token. Admin can add product and view product. Customer can view only product using rest api through react js.