Day 12 → Express & MongoDB

Express → It is a backend server.

```
npm i express
```

To continously run node server

```
npm i nodemon  → Indirectly kickstart node index.js/server.js
```

① Creating Server.

```
const express = require('express')

const app = express()

app.listen(3001, () => {

    cla("Server Started")
```

37

(2) Testing the Server

(1) Enter the url in webpage with the port number given

localhost:3001

(2) Go to postman

↳ select get request

↳ Enter the url

↳ Click send.

MongoDB.

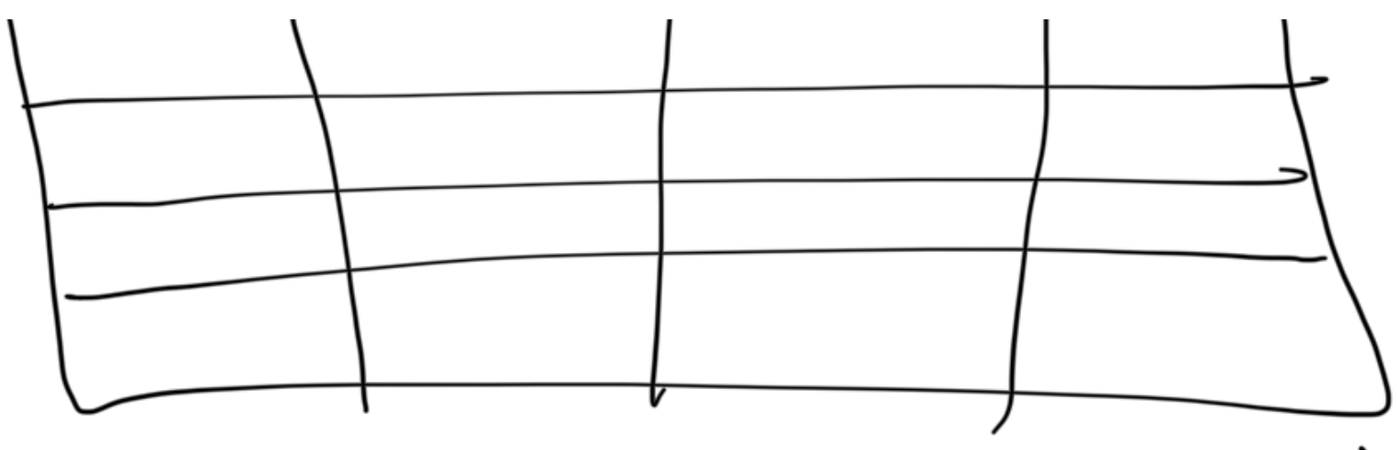
→ what kind of document? JSON

why Mongo DB: document based data

student DB:

S.No	Reg No	Name	Roll No

→ SQL → Structure Query Language



Once the table is designed it can be changed without Alter operation

NoSQL → Non Structure Query Language.

↳ ORM → Object Relational Mapping.

There is no joins concept.

Over NoSQL tables are called as collections.

```
{
  firstName : "Rollo",
  lastName : "Rollo",
  college : "sjit",
  batch : "2022"
}
```

* Select * from Table1; → Read → fetch all the data from table.

To work with Database we have operations ⇒ CRUD

C - Create
R - Read / Retriere
U → Update
D → Delete.

Database Creation :

use <databasename> ; ⇒ Switched to <db name>

Collection creation :

db.createCollection ("signup"); ⇒ { ok : 1 }

Inserting in collection :

1) insertOne

db.signup.insertOne ({"firstName": "SSIT"})

db. Signup. insert...

2) insertMany

db. Signup. insertMany ([{ }, { }{ }])

Display the records: → R

① findOne

db. signup. findOne ({ }); → condition

↓

It is will show it's 1st record no matter of how many records are there.

② find

db. signup. find({ }) → condition

↓

It will display all the records.

Updating :- → U

→ where condition

... $set:

① updateOne

↳ db.Signup.updateOne({fName: "Hello"}, {fName: "Hello, SJIT"})

Even though it matches the condition it will update the 1st record only.

② updateMany. db.Signup.updateMany({fName: "Hi"}, {$set: {fName: "Hello"}})

↓

It will update the entire records matching condition.

Delete:

↳ deleteOne: db.Signup.deleteOne({fName: "Hello"})

↓

It will delete only the 1st Hello

↳ deleteMany: db.Signup.deleteMany({fName: "Hello"})

↓

It will delete the entire records.

It will delete all ...

X

db completed.