CRUD API using NodeJS, ExpressJS, MongoDB

# Dependencies:

1. Node.js
2. Express.js
3. Nodemon
4. Insomnia
5. Postman
6. ThunderClient in VSCode

# Steps:

1. Setting and Installing Dependencies
2. Entering code in a js file like firstapi.js
3. Adding tests in scripts in package.json like

"serve": "node firstapi.js",

"dev": "nodemon firstapi.js"

1. Creating collection and folder in Insomnia and pasting the localhost link
2. Setting up MongoDB
   1. Creating Database and Cluster
   2. Setting connection string
   3. Setting with npm install mongodb and mongoose
3. Connect with the db using mongoose.connect(“connection string”);
4. Create Models in models folder like product.model.js
   1. Add Product Schema
   2. const Product=mongoose.model("Product",ProductSchema);

module.exports=Product;

1. Create a post in firstapi.js
2. Give json objects in the body while using thunderclient or insomnia
   1. But you cant get output since it cant get json object
   2. To make it get we have to add the line app.use(express.json());

## Add API:

1. Type the below code in firstapi.js

app.post('/api/products',async (req,res) => {

    try {

        const product = await Product.create(req.body);

        res.status(200).json(product);

    } catch (error) {

        res.status(500).json({message: error.message});

    }

});

1. Check the connection and add items with respect to the model

## Read API:

1. Using find() function
2. Paste the following code

app.get('/api/products',async(req,res) => {

    try {

        const products = await Product.find({});

        res.status(200).json(products);

    } catch (error) {

        res.status(500).json({message: error.message});

    }

});

1. Using findById can get particular product using id

app.get('/api/product/:id',async(req,res)=>{

    try {

        const {id}=req.params;

        const product=await Product.findById(id);

        res.status(200).json(product);

    } catch (error) {

        res.status(500).json({message:error.message});

    }

});

## Update API:

Similarly do for update

## Delete API:

Similarly do for delete

## Routes and Controllers:

1. Create folders controllers and routes
2. Create product.controller.js and product.route.js respectively
3. Cut and paste the code of CRUD and assign it in separate variables and export it in controller
4. Import the exported variables from controller to route and route for all urls