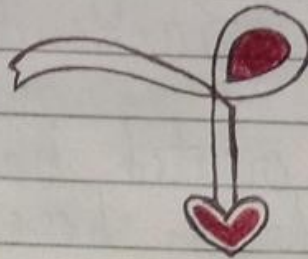


Hello Everyone,

Date 24/09/21 Page no. 1.

I'm **A**run Chaudhary

Let's, Start

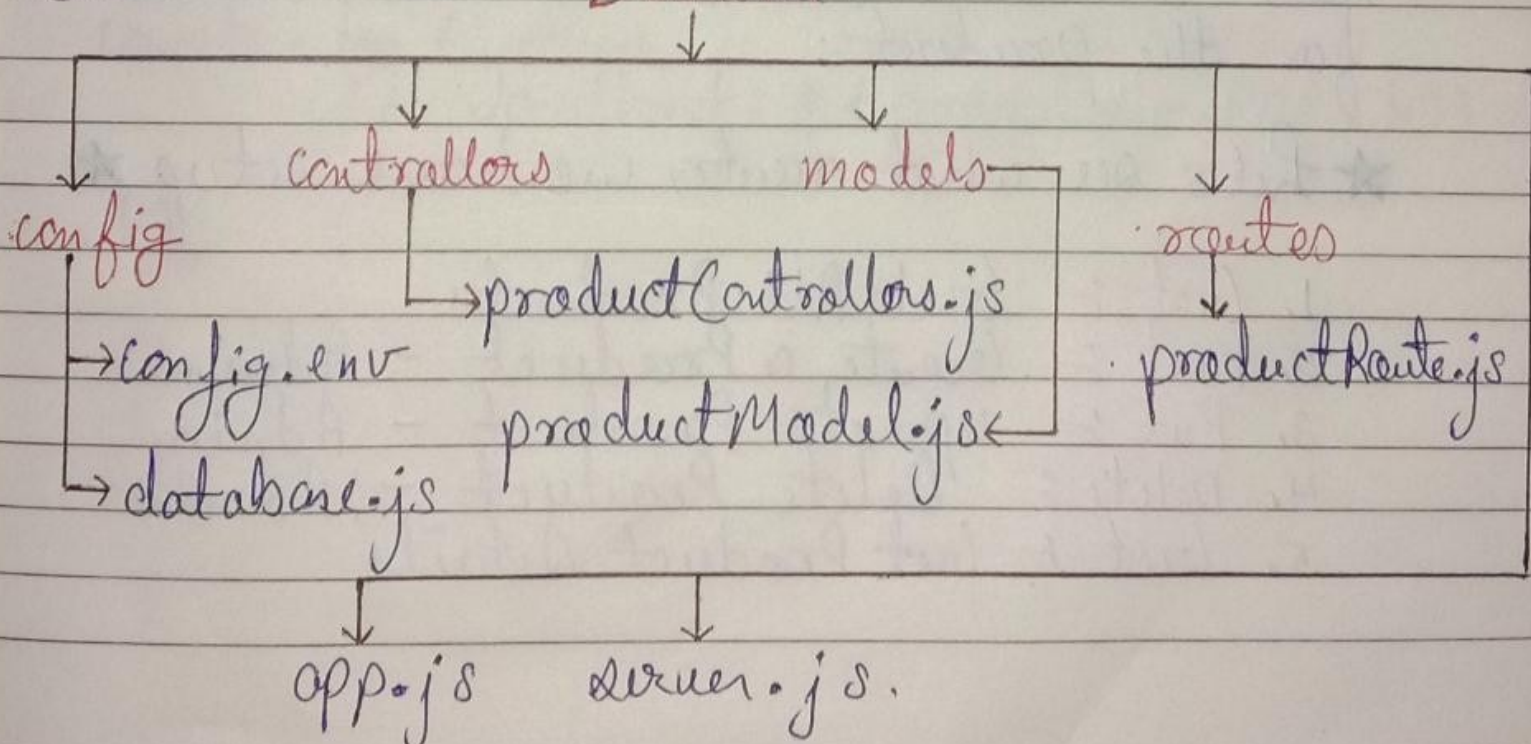


Q1. Do you know that with the help of node, express, and postman. We can create the Basic environment in this way? &

How we connect the database to the server? & we will use Mongo DB for database?

Ans =>

Backend.



Notes:- I will keep nothing what I'm using to build an Ecommerce website.

Setting Up Backend.

1. First of all we have created basic environment and run it and from here our server started working.
2. Now I have to use postman so that we can test whether our CRUD Method is working well or not.
3. Now we will connect to database and for database we will use MySQL.
4. In the second point we ~~we~~ created a route, now we will create all the routes for the product.

★ Let's see what routes we have setup ★

1. **Get** :- Get All Products
2. **Post** :- Create a Product - Admin
3. **Put** :- Update Product - Admin
4. **Delete** :- Delete Product - Admin
5. **Get** :- Get Product Details

Server.js

```
const app = require("./app");
```

```
const dotenv = require("dotenv");
```

```
const connectDatabase = require("./config/database");
```

```
// Config  
dotenv.config({ path: "backend/config/config.env" });
```

```
// Connecting to database  
connectDatabase();
```

```
// PORT
```

```
app.listen(process.env.PORT, () => {
```

```
  console.log(`Server is working on http://  
localhost: ${process.env.PORT}`)
```

```
});
```


app.js

```
const express = require("express");  
const app = express();
```

```
app.use(express.json());
```

// Import the Route

```
const product = require("./routes/productRoute");
```

```
app.use("/api/v1", product);
```

```
module.exports = app;
```

config.env

```
PORT = 4000
```

```
DB_URI = "mongodb://localhost:27017/Ecommerce"
```

database.js

```
const mongoose = require("mongoose");

const connectDatabase = () => {
  mongoose.connect(process.env.DB_URI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
    // useCreateIndex: true,
  }).then((data) => {
    console.log(`Mongoose connected with  
server: ${data.connection.host}`);
  });
};

module.exports = connectDatabase;
```


productModel.js

```
const mongoose = require("mongoose");
```

```
const productSchema = mongoose.Schema ({
```

```
  name: {
```

```
    type: String,
```

```
    required: [true, "Please Enter Product Name"],
```

```
    trim: true
```

```
  },
```

```
  description: {
```

```
    type: String,
```

```
    required: [true, "Please Enter Product description"],
```

```
  },
```

```
  price: {
```

```
    type: Number,
```

```
    required: [true, "Please Enter Product Price"],
```

```
    maxLength: [8, "Price cannot exceed 8 characters"],
```

```
  },
```

rating: {
 type: Number,
 default: 0,
 },

images: [

public_id: {
 type: String,
 required: true,
 },

url: {
 type: String,
 required: true,
 },

],

category: {
 type: String,
 required: [40, "Stock cannot
 exceed 4 characters"],
 default: 1
 ↓
 required: [true, "Please Enter
 Product Category"],

stock: {

type: String,
required: [true, "stock cannot exceed 4
characters"],

default: 1

},

numOfReviews: {

type: Number,
default: 0,

},

reviews: [

user: {

type: mongoose.Schema.ObjectId,
required: true,

},

name: {

type: String,
required: true,

},

rating: {

type: Number,
required: true,

},

comment: {

type: String,
required: true,

{,

{,

,

createdAt: {

type: Date,
default: Date.now,

{,

});

module.exports = mongoose.model("Product",
productSchema);

productController.js

const Product = require("../models/productModel");

// create product -- only admin do this work
exports.createProduct = async (req, res, next) => {

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

res.status(200).json({

success: true,

product,

});

};

// Get All Product

exports.getAllProducts = async (req, res) => {

const products = await Product.find();

res.status(200).json({

success: true,

products,

});

};

11 Get Product Details

exports.getProductDetails = async (req, res, next) => {

const product = await Product.findById(req.params.id);

if (!product) {
 return res.status(500).json({
 success: false;
 message: "Product not found"
 });
};

res.status(200).json({
 success: true,
 product
});

};

11 Update Product -- Only Admin do this work

exports.updateProduct = async (req, res, next) => {

let product = await Product.findById(req.params.id);

```
if (!product) {  
    return res.status(500).json({  
        success: false,  
        message: "Product not found",  
    });  
};
```

```
product = await Product.findByIdAnd-  
    Update(req.params.id, req.body,  
    {  
        new: true,  
        runValidators: true,  
        useFindAndModify: false,  
    });
```

```
res.status(200).json({  
    success: true,  
    product  
});  
};
```



```
// Delete Product
exports.deleteProduct = async (req, res, next) => {
  const product = await product.
    findById (req.params.id)

  if (!product) {
    return res.status(500).json({
      success: false,
      message: "Product not found"
    });
  }

  await product.remove();
  res.status(200).json({
    success: true,
    message: "Product Delete Successfully"
  });
};
```

productRoute.js

```
const express = require("express");
```

```
const { getAllProducts, createProduct,  
    updateProduct, deleteProduct,  
    getProductDetails } = require  
    ("../controllers/productController");
```

```
const router = express.Router();
```

```
router.route("/products").get(getAllProducts);
```

```
router.route("/product/new").post(createProduct);
```

```
router.route("/product/:id").put(updateProduct)  
    .delete(deleteProduct)  
    .get(getAllProducts  
        getProductDetails);
```

```
module.exports = router;
```

So far we have created our database and we also connected the database to our server and -

The work of set the route and control of the product is completed.