Backend

note :

1. to count the no of item in collection in monogo db :

const productcount=await Product.countdoucument();

2.we can make a method for the function that we define by using

function\_name.methods.method\_name=function()

2.To make user auntication

step 1:

i) install package :npm i bcryptjs jsonwebtoken validator nodemailer cooki-parser body-parser

bcryptjs: use to encrypt user pasword

validator : use to validate e-mail

nodemailer: if any one forget password then to reset password we can send email or opt

cooki-parse: is used to store a token genrated by jsonwebtoken

step2 :

i)in the model folder of backend create a file with the name usermodel , in that define your module

ii) in the controller folder create a file with the name usercontroller in that import user module

iii) in the router folder create a file with the name userroutes in that import the create route and export the route in the in app.js

iv)check wether it is working or not

Step 3:

encrypte the password

step 4 :

create token

step 5 : creating login user

i) create a function in user controller name exports.loginuser=

ii)put validation on every point

iii) create a router for that in userrouter file

iv)check it

step 6 :creating cookie for jwttoken and replacing it in the usercontroller

i)create a file in utiles with the name jwttokken.js

ii) write your code therir and export it in the usercontroller and use it

iii)import cookie-parser in App.js and write app.use(cooki-prser) in App.js

step 7:To create a file for user authentication

i)in middleware file create authintication.js and write your function upto console .log(token)

ii)import authintication.js in the router/productrouter/ and uder it use it in the .get route

iii)run postman of getproduct and chek wether we are getting token or not

iv)creating cases if token find and if not find

v) then run it

step 8. creating logout in controller/user/contoller

i)create logout and create a logout router

ii) run logout on postman

step 9 : use isauthincate user in updateproduct , deleteproduct,getproductdetails

step 10 : in middleware/authication/

create authroizedUser

i) export it in the router/productrouter

and use it as second argument beside authinticate user

step 11: creating reset password functon

i)create a method in modles/usermodels restpassword token

ii) in controller/usercontroller/ create forget password

3.During making a model if we want the input should be unique then we can use : unique:true

  email: {

    type: *String*,

    required: [true, "Please enter your email"],

    unique: true,

    validator: [validator.isEmail, "Please enter valid email"],

  },

5.During find method if we donot want anyy field to get selected then we can use :

select:false

password: {

    type: *String*,

    required: [true, "Please enter your password"],

    minLength: [8, "Your password should have minimum length 8"],

    maxLength: [32, "Max length of password should not exceed 32 chracter"],

    select:false

  },

5.11 . during the find method if we want to select the filed whome we have give select:false , we can do that by useing .select

*let* user=await User.find({email}).select("+password");

5.encrypt the password :

i)import mdoule: bcryptjs in usermodule

userSchmas.pre("save",async *function*(){

    if(!this.isModified("password")){

      next();

    }

    else{

      this.password=await Bcryptjs.hash(this.password,10);

    }

})

6.To genrate a token

i)import moule : jwt=require(‘jsonwebtken’);

ii)int the user module file create a a method with a function

userSchmas.methods.getJWTToken=*function*(){

  return jwt.sign({id:this.\_id},process.env.JWT\_SECRET,{

    expiresIn:process.env.JWT\_EXPIRE,

  })

}

iii) use the above made functon in usercontroller inside the post method