   "bcrypt": "^5.0.1",

    "dotenv": "^16.0.1",

    "ejs": "^3.1.8",

    "express": "^4.18.1",

    "jsonwebtoken": "^8.5.1",

    "mongoose": "^6.3.4"

User.js (Model)

import mongoose from "mongoose"

const userSchema = mongoose.Schema({

   name :{

     type: String,

     required:true

   },

   email :{

    type: String,

    required:true,

    unique: true

  },

  password :{

    type: String,

  required:true

  },

  date :

  {

    type:Date,

    default:Date.now

  }

})

const userModel = mongoose.model("user", userSchema);

export default userModel;

Routes

import express  from "express";

const authrouter = express.Router();

import userController

 from "../controllers/userController.js";

 authrouter.post("/register", userController.

 userRegistration)

 authrouter.post("/login", userController.

 userLogin)

 export default authrouter;

mideleware

import jwt from "jsonwebtoken"

import userModel from "../model/user.js"

var authenticateUser = async(req,res,next)=>{

  let token

  const {authorization} = req.headers

  if(authorization && authorization.startsWith("Bearer"))

  {

     console.log("has bearer")

    try{

 token  = authorization.split(' ')[1]

 console.log(token)

//  const data = {

//    user : {

//      email : req.body.email

//    }

//  }

const key  = process.env.JWT\_SECRETKEY

    console.log(key)

    const {email} =   jwt.verify(token,key)

  req.user = await userModel.findOne({email:email})

next()

    }

    catch(error)

    {

   res.status(401).send({"status":"failed","message":"Not ver"})

    }

  }

  if(!token)

  {

    res.status(401).send({"status":"failed","message":"There is no token"})

  }

}

export default authenticateUser

userController.js

import userModel from "../model/user.js"

import bcrypt from "bcrypt"

import jwt from "jsonwebtoken"

class userController{

  static userRegistration = async(req,res)=>

  {

     const{name,email,password,date}  = req.body

    console.log(req.body)

     const user = await userModel.findOne({email : email})

     console.log(user)

     if(user)

     {

       res.send({"status" :"failed", "message":"Email already exist"})

     }

     else if(name && password && email)

     {

       // Encrypt Password

       const salt = await bcrypt.genSalt(10);

       const secPassword = await bcrypt.hash(req.body.password, salt)

        const document = new userModel(

  {

    name : name ,

    email : email,

    password : secPassword,

    date : date

  }

)

 const result = await document.save();

 res.status(201).send({"status" :"succes", "message" :"User is registered"})

}

else

{

  res.send({"status" :"failed", "message":"All fields are required"})

}

  }

static userLogin = async (req,res)=>

{

  try{

    const {email, password} = req.body

    console.log(email + " " + password)

    if(email && password)

    {

      const user = await userModel.findOne({email:email})

      if(user!=null)

      {

       // Decrypt Password

        const isMatch = await bcrypt.compare(password, user.password)

      console.log("P"+ password + " " + user.password)

      if((user.email===email) && isMatch)

      {

      const data =

      {

         user :

         {

            email : user.email

         }

      }

      const key  = process.env.JWT\_SECRETKEY

      const token = jwt.sign(data,key,

        {expiresIn:'5d'})

console.log(token)

res.send({"status":"success","message":"User exist","token":token})

      }

      else

      {

        console.log("not match")

        res.send({"status":"failed","message":"User do not exist"})

      }

     }

    else

    {

      res.send({"status":"failed","message":"Not reg"})

    }

  }

  else

  {

    res.send({"status":"failed","message":"Provide both email & password"})

  }

  }

  catch(error)

  {

    res.send(error)

  }

}

}

export default userController;

app.js

import express from "express";

import connectDb from './connectDb.js';

import {join} from "path";

import web from './routes/web.js';

import authroute from "./routes/auth.js";

import dotenv from 'dotenv';

import authenticateUser from './middleware/auth-middleware.js'

//const port = 3000;

var app = express();

dotenv.config();

const port = process.env.PORT

// console.log(port)

app.use(express.json());

// const databaseUrl ="mongodb+srv://user:password123567@cluster0.ulbwxcb.mongodb.net/practiceDb?retryWrites=true&w=majority";

app.use("/api",  web);

//Public Route

app.use("/api/user", authroute)

// console.log(process.env.DATABASE\_URL)

const databaseUrl = process.env.DATABASE\_URL

// console.log(databaseUrl)

// const databaseUrl ="mongodb://localhost:27017";

// app.set("views" ,"/views");

app.set("view engine" ,"ejs");

app.use(express.static(join(process.cwd(), "public")));

connectDb(databaseUrl);

app.listen(port, ()=>

{

  console.log(`Connected at ${port}`);

})