User.js(model)

const mongoose = require('mongoose')

const userSchema = new mongoose.Schema({

    name: {

        type: String,

        required: true

    },

    tech: {

        type: String,

        required: true

    },

    sub: {

        type: Boolean,

        required: true,

        default: false

    }

})

module.exports = mongoose.model('employeedbs', userSchema)

app.js

const express = require('express')

const mongoose = require('mongoose')

const url = 'mongodb://localhost:27017/emp'

const app = express()

mongoose.connect(url, { useNewUrlParser: true }, { useUnifiedTopology: true })

const con = mongoose.connection

con.on('open', () => {

    console.log('Connected')

})

app.use(express.json())

const userRouter = require('./routes/users')

app.use('/users', userRouter)

app.listen(9000, () => {

    console.log('Server started')

})

Users.js

const express = require('express')

const router = express.Router()

const User = require('../models/user')

router.get('/', async(req, res) => {

    try {

        const var1 = await User.find()

        res.json(var1)

    } catch (err) {

        res.send('Error' + err)

    }

    res.send('Get Request')

})

router.get('/:id', async(req, res) => {

    try {

        const var2 = await User.findById(req.params.id)

        res.json(var2)

    } catch (err) {

        res.send('Error' + err)

    }

    res.send('Get Request')

})

router.post('/', async(req, res) => {

    const data = new User({

        name: req.body.name,

        tech: req.body.tech,

        sub: req.body.sub

    })

    try {

        const p = await data.save()

        res.json(p)

    } catch (err) {

        res.send('Error')

    }

})

router.patch('/:id', async(req, res) => {

    try {

        const data1 = await User.findById(req.params.id)

        data1.name= req.body.name

        data1.tech= req.body.tech

        data1.sub = req.body.sub

        const a1 = await data1.updateOne()

        res.json(a1);

    } catch (err) {

        res.send('Error')

    }

})

router.delete('/:id', async(req, res) => {

    try {

        const data2 = await User.findById(req.params.id)

        //data2.sub = req.body.sub

        const d = await data2.remove()

        res.json(d);

    } catch (err) {

        res.send('Error')

    }

})

module.exports = router