Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**12**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | **Create a WebApi project in nodejs that contains the complete RESTFUL URLs for Product Management.** |
|  |  |
|  |  |

**Task 01: Create a WebApi project in nodejs that contains the complete RESTFUL URLs for Product Management.**

**Solution:**const express=require("express")

const cookieParser=require("cookie-parser")

const bcrypt=require("bcryptjs")

const mongoose=require("mongoose")

const Register=require("./models/reg")

const Collect=require("./models/collectRent")

const Pay=require("./models/payRent")

require("./db/conn")

const auth=require("./middleware/auth")

const hbs=require("hbs")

const path=require("path")

const template\_path=path.join(\_\_dirname,"./templates/views")

const partial\_path=path.join(\_\_dirname,"./templates/partials")

const port=8000

const app=express()

app.use(express.urlencoded({extended:false}))

app.use(cookieParser())

app.set("view engine","hbs")

app.set("views",template\_path)

hbs.registerPartials(partial\_path)

app.get("/",(req,res)=>{

    res.render("sign")

})

app.get("/signin",(req,res)=>{

    res.render("signin")

})

app.post("/signup",async(req,res)=>{

    try

    {

    console.log(req.body.name)

    const pass=req.body.password

    console.log("Passsword: "+pass);

    const useremail=await Register.findOne({email:req.body.email})

    if(!req.body.name){

        return res.json({message:"name"})

    }

    if(!req.body.password){

        return res.json({message:"password"})

    }

    if(!req.body.email){

        return res.json({message:"empEmail"})

    }

    if(useremail){

        return res.json({message:"exist"})

    }

        // res.send("pass match")

        const regEmployee=new Register({

            name:req.body.name,

            email:req.body.email,

            passsword:req.body.password

        })

        console.log(regEmployee);

        const regis=await regEmployee.save()

        console.log("Hey Amjad REgister Successfully");

    res.render("sign")

}

    catch (error) {

    console.log(error);

    return res.status(422).json({message:"User register failed"})

}

})

app.post("/login",async(req,res)=>{

try {

    const email=req.body.email

    console.log("Password from server: "+req.body.password);

    const useremail=await Register.findOne({email:email})

    if(!email){

        return res.json({message:"email"})

    }

    useremail.tokens=[]

    const pass=req.body.password

        if(useremail.passsword===pass){

            console.log("matched successfully");

            res.render("profile",{email})

        }

        else{

            console.log("Not Matched");

            res.status(400).json({message:"User signin Failed"})

        }

} catch (e) {

    console.log(e);

    return res.status(422).json({message:"User failed"})

}

})

app.post("/payRent",async(req,res)=>{

    const item=req.body.item

    const price=req.body.price

    if(!customer){

        return res.json({message:"customer"})

    }

    if(!item){

        return res.json({message:"item"})

    }

    if(!price){

        return res.json({message:"price"})

    }

    const useremail=await Pay.findOne({customer:customer})

    if(useremail){

        return res.json({message:"exist"})

    }

    const payItem=new Pay({

        customer:req.body.customer,

        item:req.body.name,

        price:req.body.price,

        status:"Not Paid"

    })

    const regis=await payItem.save()

})

app.post("/collectRent",async(req,res)=>{

    const item=req.body.item

    const price=req.body.price

    const customer=req.body.customer

    if(!customer){

        return res.json({message:"customer"})

    }

    if(!item){

        return res.json({message:"item"})

    }

    if(!price){

        return res.json({message:"price"})

    }

    const useremail=await Collect.findOne({customer:customer})

    if(useremail){

        return res.json({message:"exist"})

    }

    const collectItem=new Collect({

        customer:req.body.customer,

        item:req.body.name,

        price:req.body.price,

        status:"Not Collected"

    })

    const regis=await collectItem.save()

})

app.get("/getPayRent",async(req,res)=>{

    try {

        const items = await Pay.find()

        res.json(items)

      } catch (error) {

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

      }

})

app.get("/getCollectRent",async(req,res)=>{

    try {

        const items = await Collect.find()

        res.json(items)

      } catch (error) {

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

      }

})

app.put("/payingRent",async(req,res)=>{

    cname=req.body.name

    if(!cname){

        return res.json({message:"cname"})

    }

    const useremail=await Pay.findOne({customer:cname})

    if(!useremail){

        return res.json({message:"notFound"})

    }

    useremail.status="Paid"

    await useremail.save()

})

app.put("/collectingRent",async(req,res)=>{

    cname=req.body.name

    if(!cname){

        return res.json({message:"cname"})

    }

    const useremail=await Pay.findOne({customer:cname})

    if(!useremail){

        return res.json({message:"notFound"})

    }

    useremail.status="Collected"

    await useremail.save()

})

app.delete("/deletingAccount",async(req,res)=>{

    try {

        const item = await Register.findOneAndDelete({ email: req.body.email })

        if (!item) {

          return res.json({message:"incorrect"})

        }

        res.json({ message: 'Deleted successfully' })

      } catch (error) {

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

      }

})

app.listen(port,()=>{

    console.log(`Listening to port ${port}`);

})