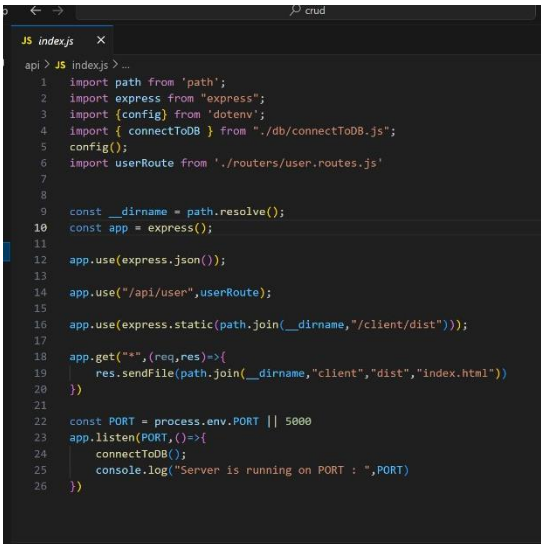
Name: KARRI JYOTHI SREE

Phone: 8106148809

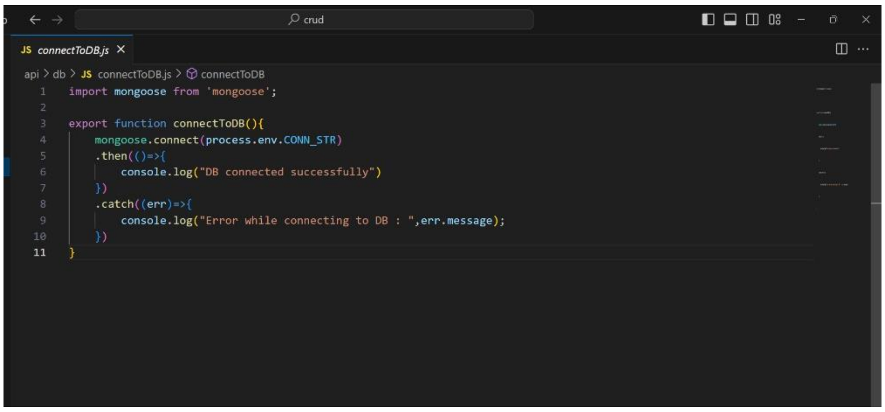
Email: karrijyothisree@gmail.com

Roll Num: 20NM1A1221

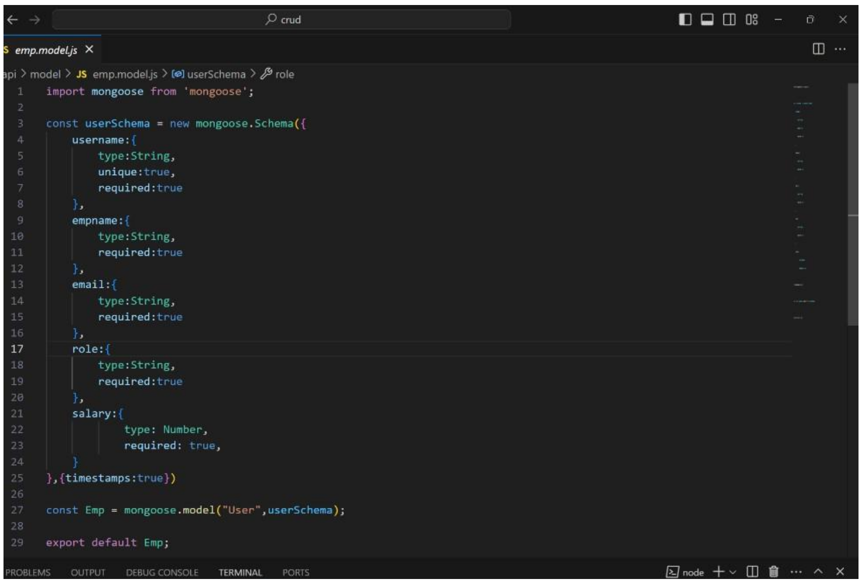
College: VIGNAN’S INSTITUTE OF ENGINEERING FOR WOMEN



```javascript
import path from 'path';
import express from "express";
import {config} from 'dotenv';
import { connectToDB } from "./db/connectToDB.js";
config();
import userRoute from './routers/user.routes.js'


const __dirname = path.resolve();
const app = express();

app.use(express.json());

app.use("/api/user",userRoute);

app.use(express.static(path.join(__dirname,"/client/dist")));

app.get("*",(req,res)=>{
    res.sendFile(path.join(__dirname,"client","dist","index.html"))
})

const PORT = process.env.PORT || 5000
app.listen(PORT,()=>{
    connectToDB();
    console.log("Server is running on PORT : ",PORT)
})
```



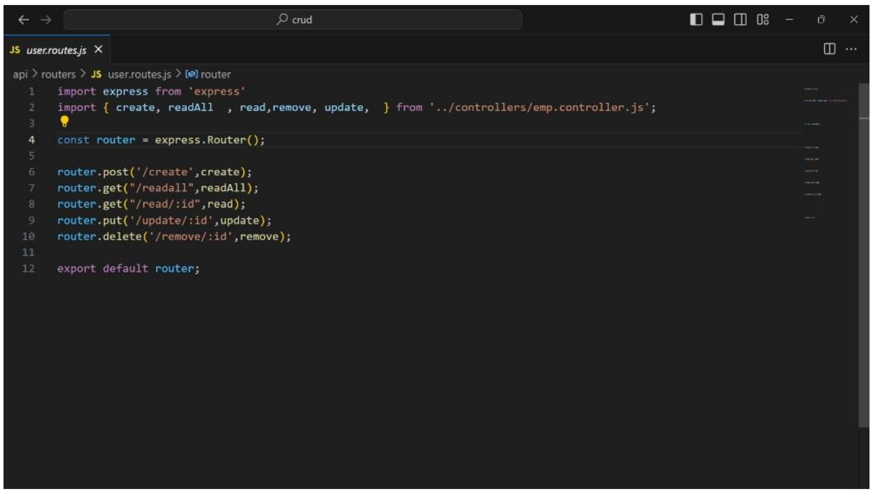
```javascript
import mongoose from 'mongoose';

export function connectToDB(){
    mongoose.connect(process.env.CONN_STR)
    .then(()=>{
        console.log("DB connected successfully")
    })
    .catch((err)=>{
        console.log("Error while connecting to DB : ",err.message);
    })
}
```



emp.model.js

api > model > emp.model.js > userSchema > role

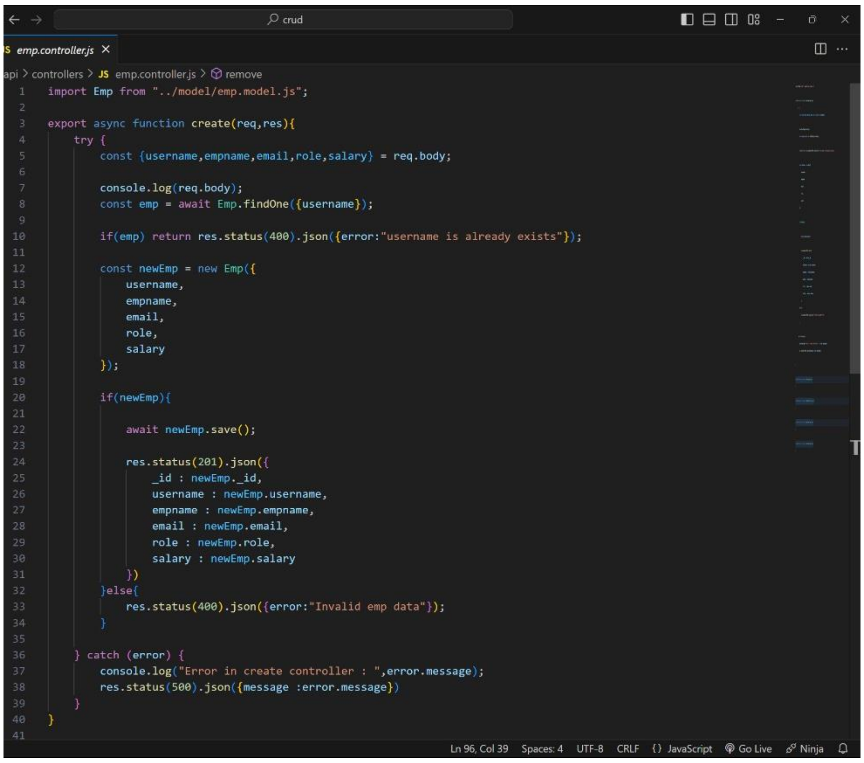
```javascript
import mongoose from 'mongoose';

const userSchema = new mongoose.Schema({
    username:{
        type:String,
        unique:true,
        required:true
    },
    empname:{
        type:String,
        required:true
    },
    email:{
        type:String,
        required:true
    },
    role:{
        type:String,
        required:true
    },
    salary:{
            type: Number,
            required: true,
    }
},{timestamps:true})

const Emp = mongoose.model("User",userSchema);

export default Emp;
```



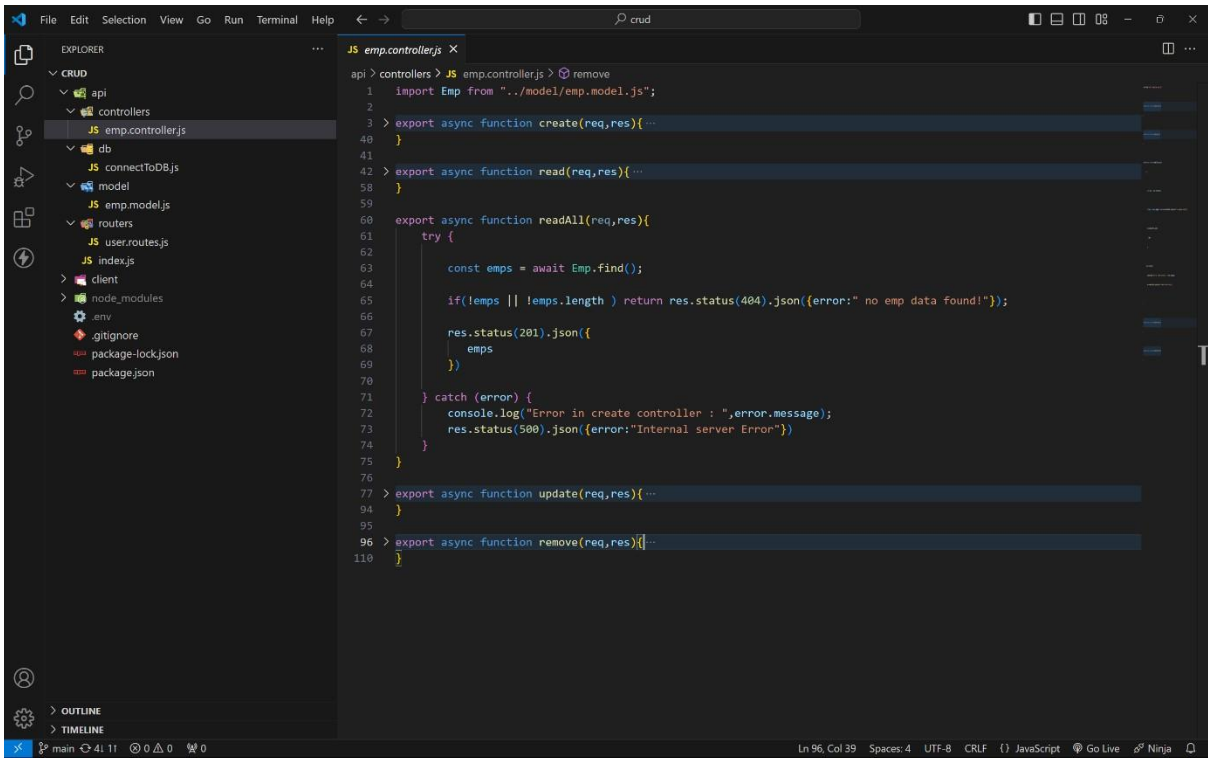
user.routes.js

api > routers > user.routes.js > router

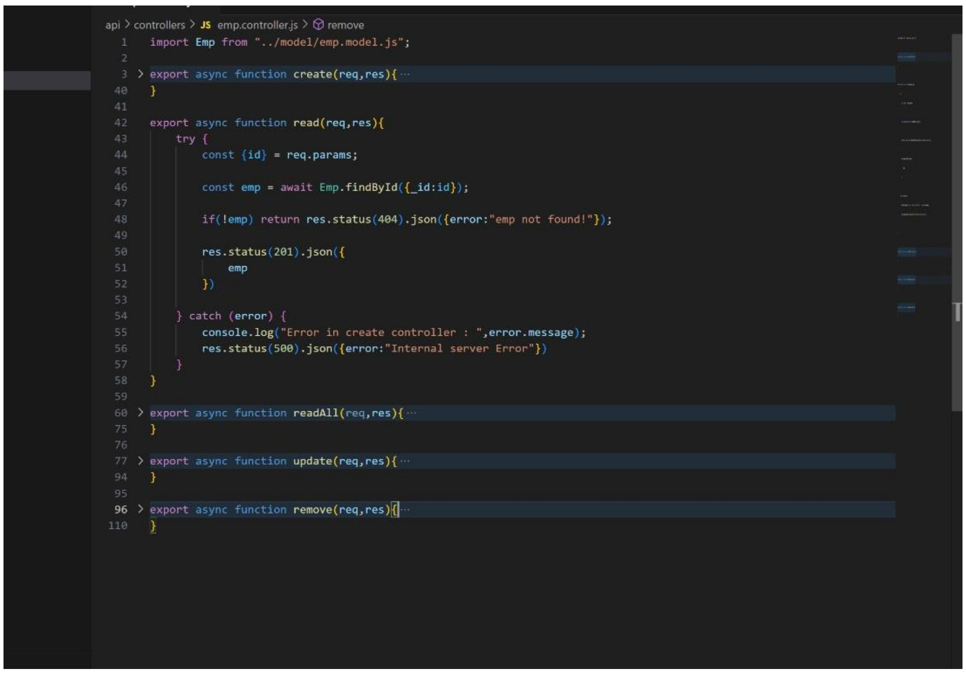
```javascript
import express from 'express'
import { create, readAll  , read,remove, update,  } from '../controllers/emp.controller.js';

const router = express.Router();

router.post('/create',create);
router.get("/readall",readAll);
router.get("/read/:id",read);
router.put('/update/:id',update);
router.delete('/remove/:id',remove);

export default router;
```



```
import Emp from "../model/emp.model.js";

export async function create(req,res){
    try {
        const {username,empname,email,role,salary} = req.body;

        console.log(req.body);
        const emp = await Emp.findOne({username});

        if(emp) return res.status(400).json({error:"username is already exists"});

        const newEmp = new Emp({
            username,
            empname,
            email,
            role,
            salary
        });

        if(newEmp){

            await newEmp.save();

            res.status(201).json({
                _id : newEmp._id,
                username : newEmp.username,
                empname : newEmp.empname,
                email : newEmp.email,
                role : newEmp.role,
                salary : newEmp.salary
            })
        }else{
            res.status(400).json({error:"Invalid emp data"});
        }

    } catch (error) {
        console.log("Error in create controller : ",error.message);
        res.status(500).json({message :error.message})
    }
}

```

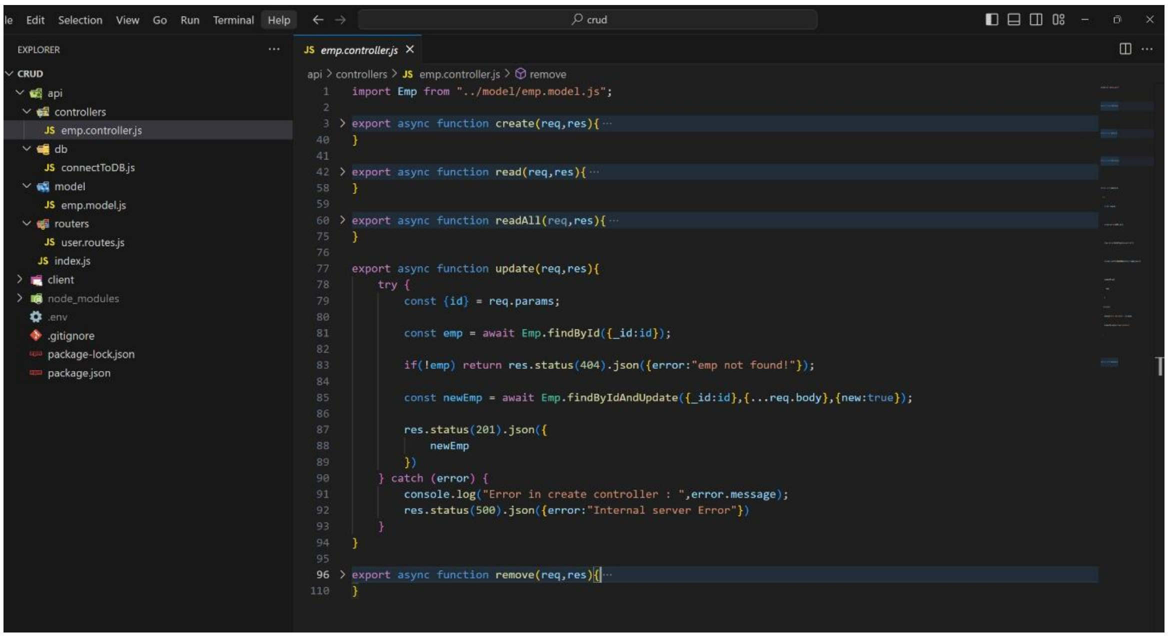


```
import Emp from "../model/emp.model.js";

export async function create(req,res){ ···
}

export async function read(req,res){ ···
}

export async function readAll(req,res){
    try {

        const emps = await Emp.find();

        if(!emps || !emps.length ) return res.status(404).json({error:" no emp data found!"});

        res.status(201).json({
           emps
        })

    } catch (error) {
        console.log("Error in create controller : ",error.message);
        res.status(500).json({error:"Internal server Error"})
    }
}

export async function update(req,res){ ···
}

export async function remove(req,res){ ···
}
```



api > controllers > JS emp.controller.js > remove

```
import Emp from "../model/emp.model.js";

export async function create(req,res){···
}

export async function read(req,res){
    try {
        const {id} = req.params;

        const emp = await Emp.findById({_id:id});

        if(!emp) return res.status(404).json({error:"emp not found!"});

        res.status(201).json({
            emp
        })

    } catch (error) {
        console.log("Error in create controller : ",error.message);
        res.status(500).json({error:"Internal server Error"})
    }
}

export async function readAll(req,res){···
}

export async function update(req,res){···
}

export async function remove(req,res){···
}
```



le Edit Selection View Go Run Terminal Help

crud

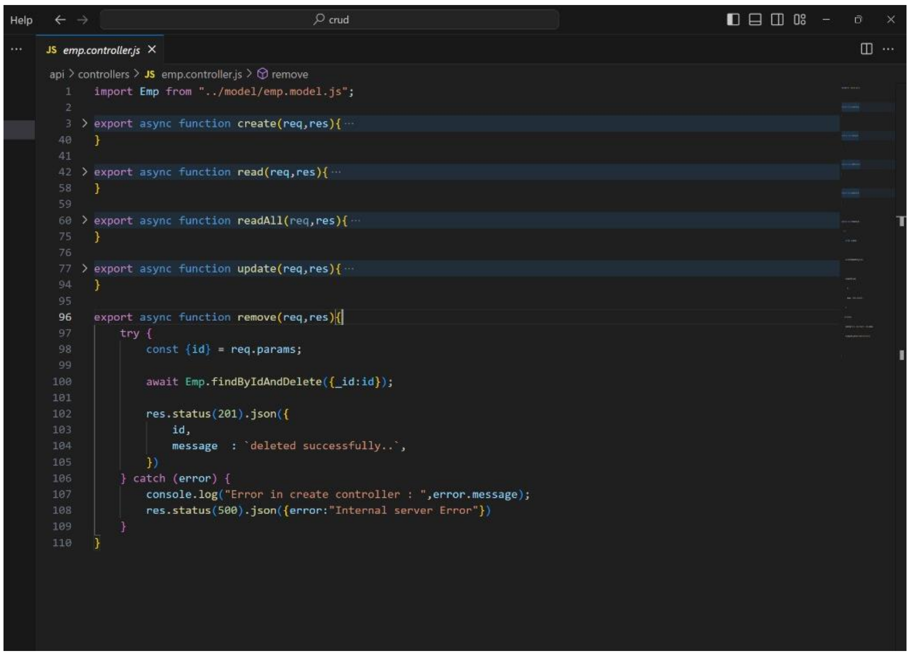
EXPLORER

```
CRUD
  api
    controllers
      emp.controller.js
    db
      connectToDB.js
    model
      emp.model.js
    routers
      user.routes.js
    index.js
  client
  node_modules
  .env
  .gitignore
  package-lock.json
  package.json
```

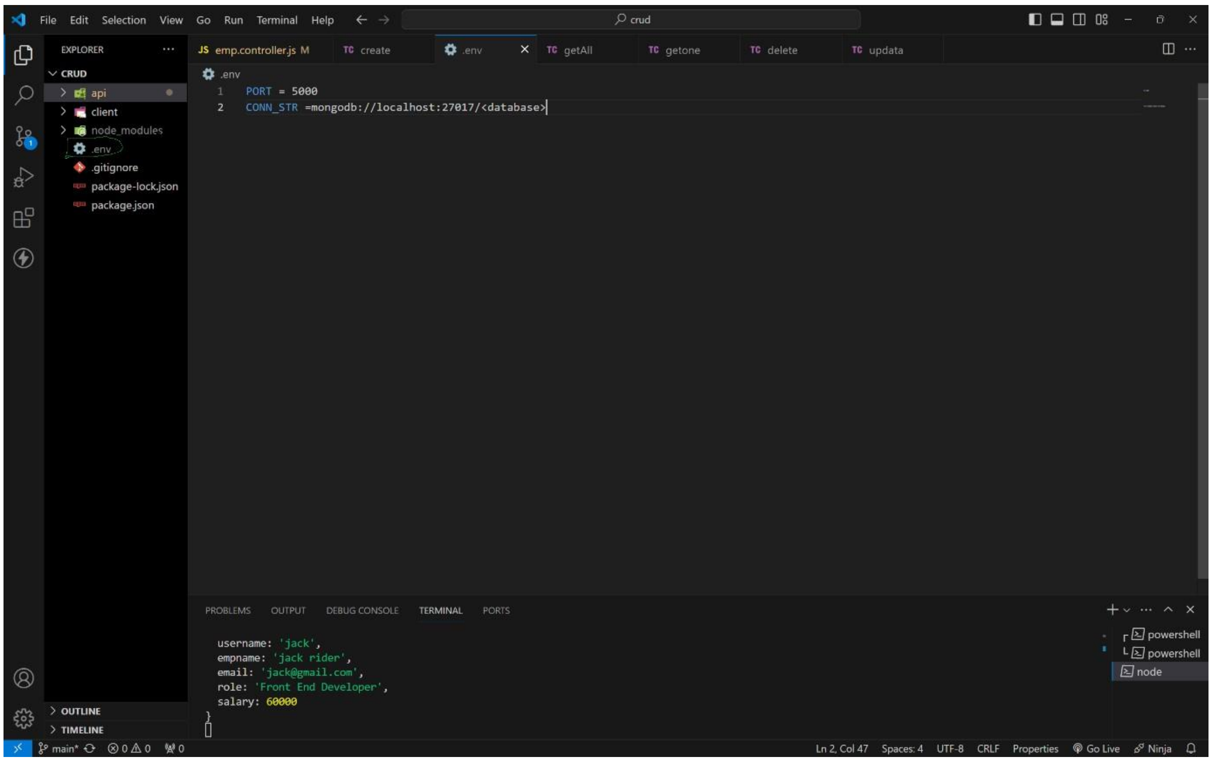
emp.controller.js

api > controllers > JS emp.controller.js > remove

```
import Emp from "../model/emp.model.js";

export async function create(req,res){···
}

export async function read(req,res){···
}

export async function readAll(req,res){···
}

export async function update(req,res){
    try {
        const {id} = req.params;

        const emp = await Emp.findById({_id:id});

        if(!emp) return res.status(404).json({error:"emp not found!"});

        const newEmp = await Emp.findByIdAndUpdate({_id:id},{...req.body},{new:true});

        res.status(201).json({
            newEmp
        })
    } catch (error) {
        console.log("Error in create controller : ",error.message);
        res.status(500).json({error:"Internal server Error"})
    }
}

export async function remove(req,res){···
}
```

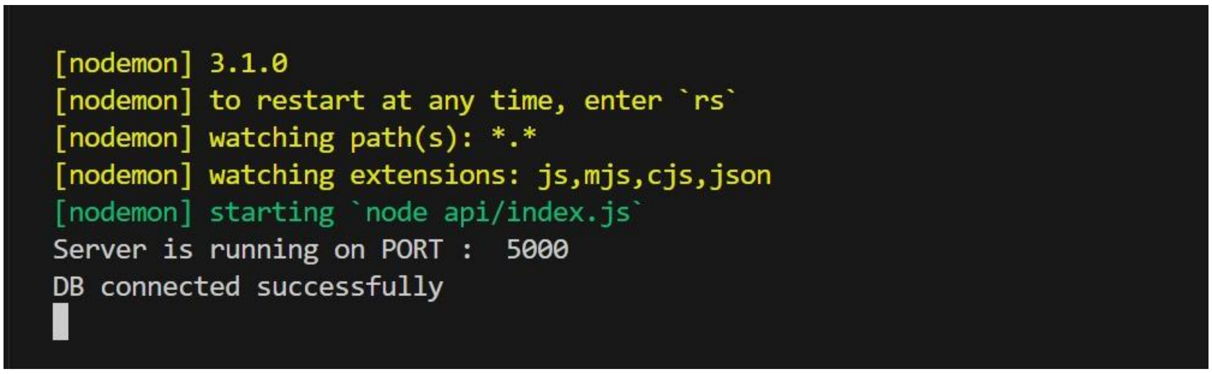


```javascript
import Emp from "../model/emp.model.js";

export async function create(req,res){ ···
}

export async function read(req,res){ ···
}

export async function readAll(req,res){ ···
}

export async function update(req,res){ ···
}

export async function remove(req,res){
    try {
        const {id} = req.params;

        await Emp.findByIdAndDelete({_id:id});

        res.status(201).json({
            id,
            message  : `deleted successfully..`,
        })
    } catch (error) {
        console.log("Error in create controller : ",error.message);
        res.status(500).json({error:"Internal server Error"})
    }
}
```



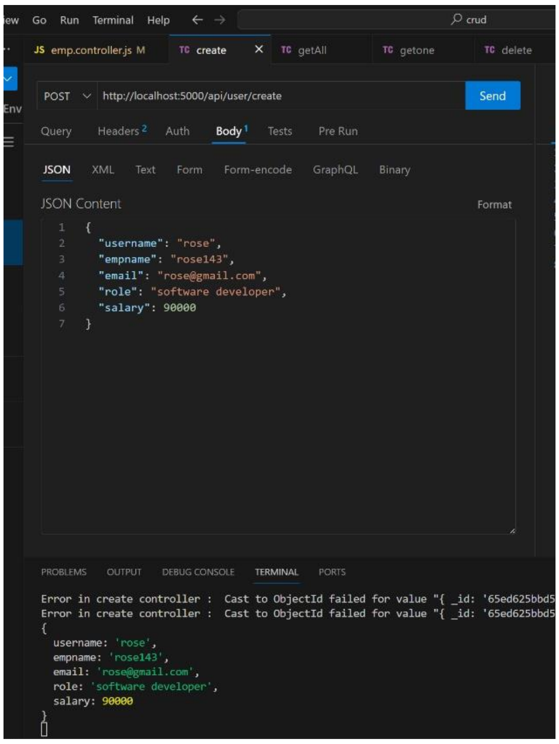
```
PORT = 5000
CONN_STR =mongodb://localhost:27017/<database>
```

```
  username: 'jack',
  empname: 'jack rider',
  email: 'jack@gmail.com',
  role: 'Front End Developer',
  salary: 60000
}
```



```
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node api/index.js`
Server is running on PORT :  5000
DB connected successfully
```

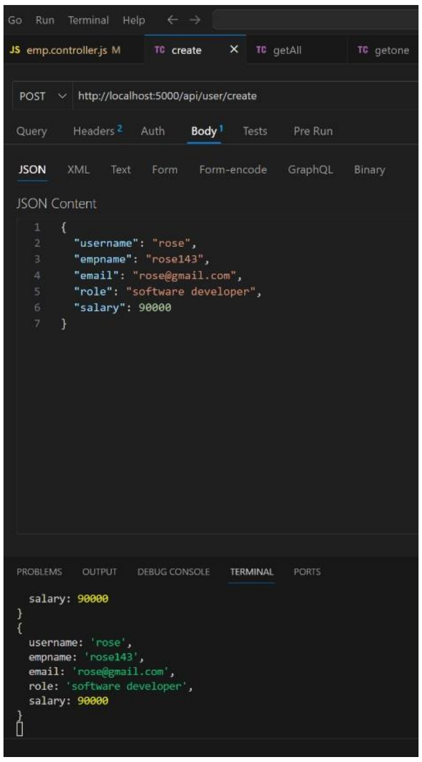
Output :


iew Go Run Terminal Help
crud
emp.controller.js M
create
getAll
getone
delete
Send
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
JSON Content
Format
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

POST http://localhost:5000/api/user/create

```
{
  "username": "rose",
  "empname": "rose143",
  "email": "rose@gmail.com",
  "role": "software developer",
  "salary": 90000
}
```

```
Error in create controller :  Cast to ObjectId failed for value "{ _id: '65ed625bbd5
Error in create controller :  Cast to ObjectId failed for value "{ _id: '65ed625bbd5
{
  username: 'rose',
  empname: 'rose143',
  email: 'rose@gmail.com',
  role: 'software developer',
  salary: 90000
}
```

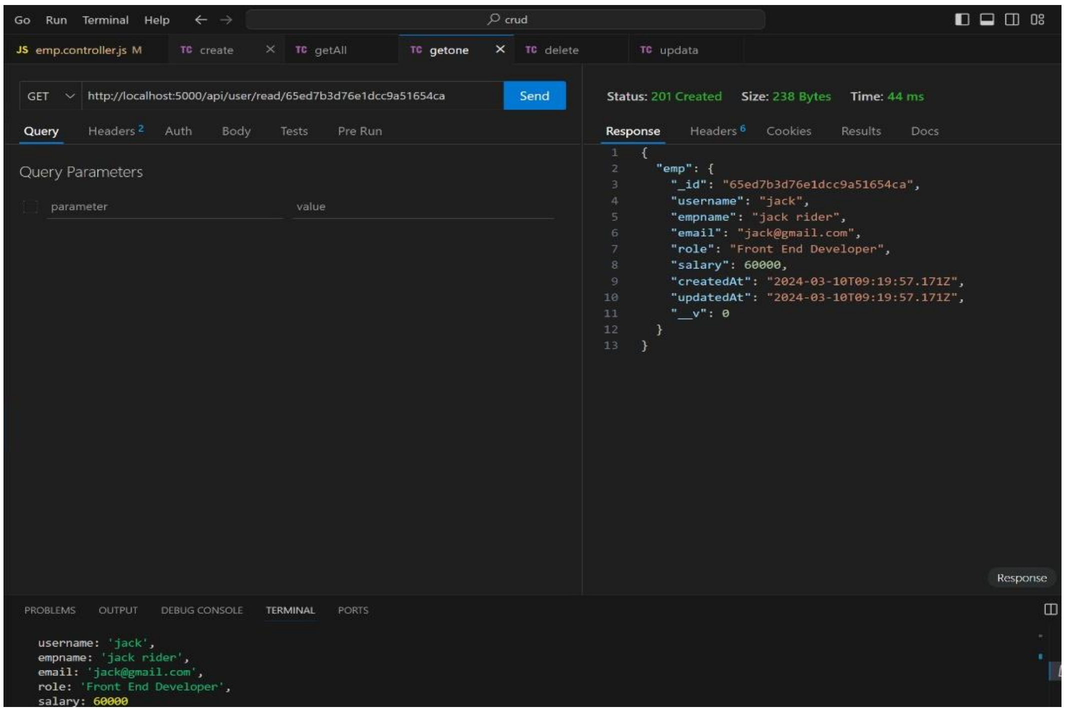

Go Run Terminal Help
emp.controller.js M
create
getAll
getone
POST
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

POST http://localhost:5000/api/user/create

JSON Content

```
{
  "username": "rose",
  "empname": "rose143",
  "email": "rose@gmail.com",
  "role": "software developer",
  "salary": 90000
}
```

```
  salary: 90000
}
{
  username: 'rose',
  empname: 'rose143',
  email: 'rose@gmail.com',
  role: 'software developer',
  salary: 90000
}
```


Go Run Terminal Help
crud
emp.controller.js M
create
getAll
getone
delete
updata
GET
Send
Query Headers 2 Auth Body Tests Pre Run
Response Headers 6 Cookies Results Docs
parameter
value
Response
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

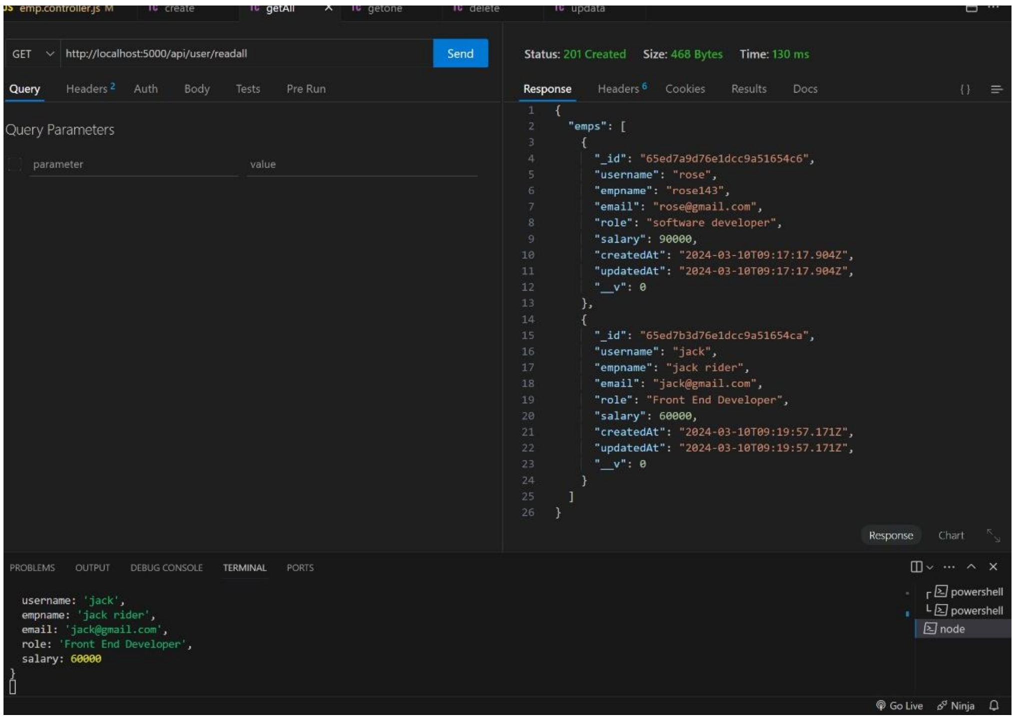
GET http://localhost:5000/api/user/read/65ed7b3d76e1dcc9a51654ca

Status: 201 Created Size: 238 Bytes Time: 44 ms

Query Parameters

```
{
  "emp": {
    "_id": "65ed7b3d76e1dcc9a51654ca",
    "username": "jack",
    "empname": "jack rider",
    "email": "jack@gmail.com",
    "role": "Front End Developer",
    "salary": 60000,
    "createdAt": "2024-03-10T09:19:57.171Z",
    "updatedAt": "2024-03-10T09:19:57.171Z",
    "__v": 0
  }
}
```

```
  username: 'jack',
  empname: 'jack rider',
  email: 'jack@gmail.com',
  role: 'Front End Developer',
  salary: 60000
```



```
GET http://localhost:5000/api/user/readall
```

Status: 201 Created Size: 468 Bytes Time: 130 ms

```
{
  "emps": [
    {
      "_id": "65ed7a9d76e1dcc9a51654c6",
      "username": "rose",
      "empname": "rose143",
      "email": "rose@gmail.com",
      "role": "software developer",
      "salary": 90000,
      "createdAt": "2024-03-10T09:17:17.904Z",
      "updatedAt": "2024-03-10T09:17:17.904Z",
      "__v": 0
    },
    {
      "_id": "65ed7b3d76e1dcc9a51654ca",
      "username": "jack",
      "empname": "jack rider",
      "email": "jack@gmail.com",
      "role": "Front End Developer",
      "salary": 60000,
      "createdAt": "2024-03-10T09:19:57.171Z",
      "updatedAt": "2024-03-10T09:19:57.171Z",
      "__v": 0
    }
  ]
}
```

```
  username: 'jack',
  empname: 'jack rider',
  email: 'jack@gmail.com',
  role: 'Front End Developer',
  salary: 60000
}
```