Creating a RESTful API using express.js and creating a database and index in MongoDB.

NAME : KSHATRI AKHILESWARI BAI

EMAIL ID : 208X1a0537@khitguntur.ac.in

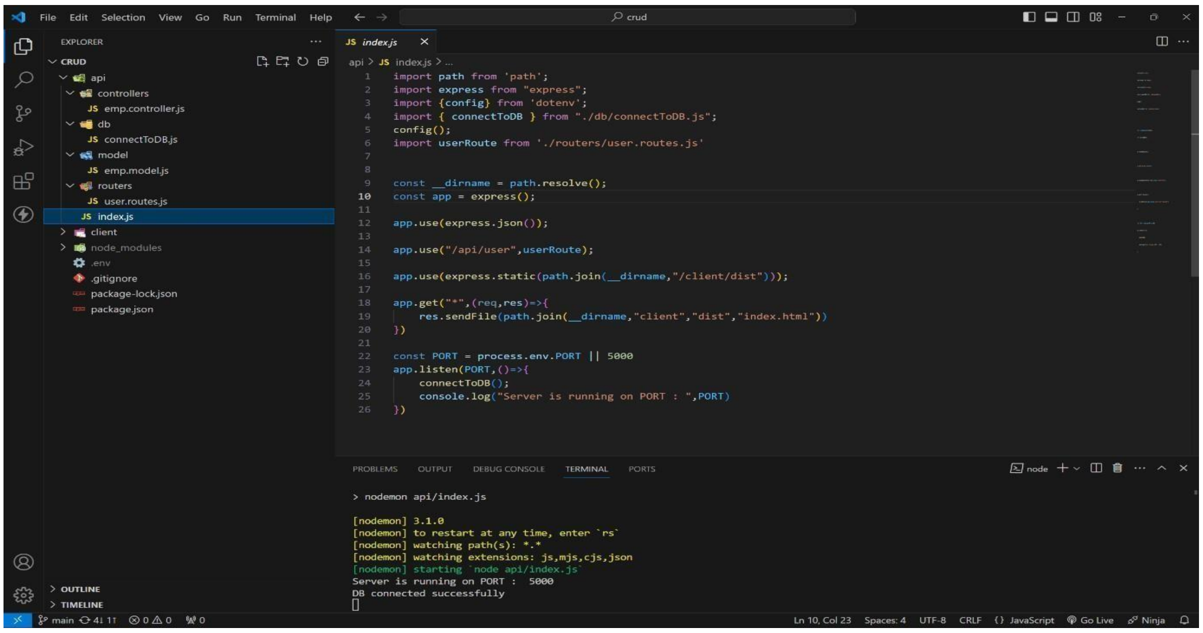
PHONE NO : 8712363393

ROLL NO : 208X1A0537

COLLEGE : KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY, GUNTUR

source code :

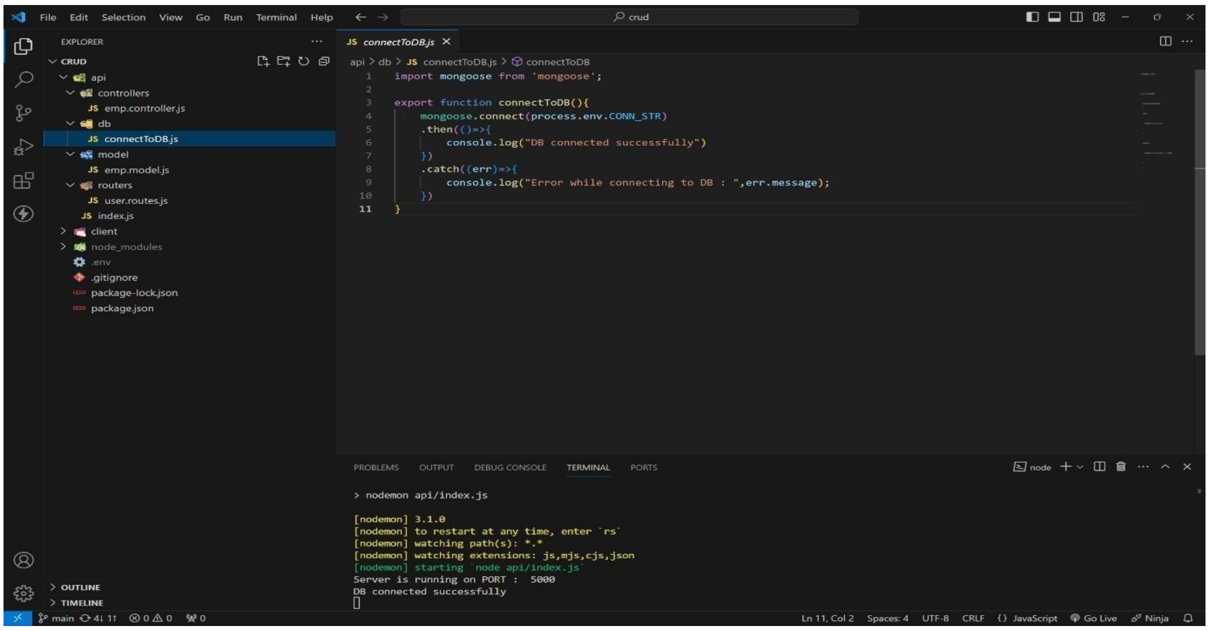
index.js file :



```
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)
})
```

```
> nodemon api/index.js

[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
```

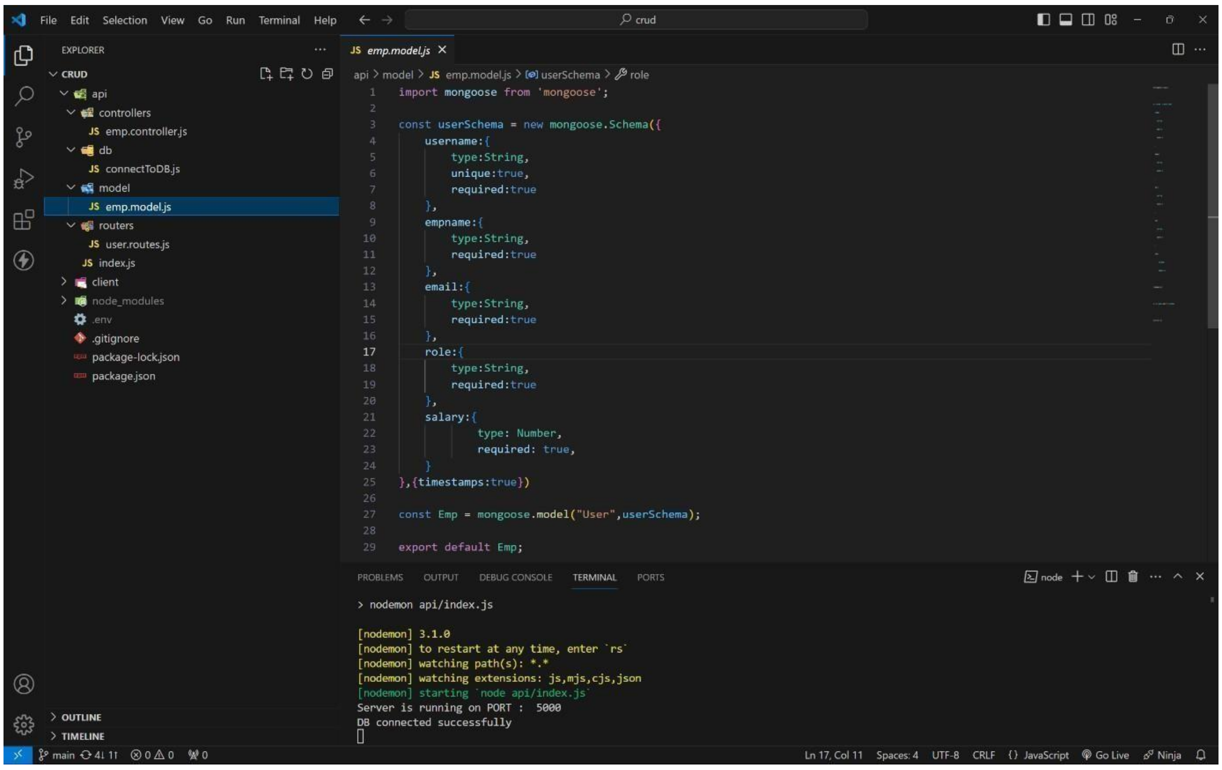
MONGODB CONNECTION :



```
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);
    })
}
```

```
> nodemon api/index.js

[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
```

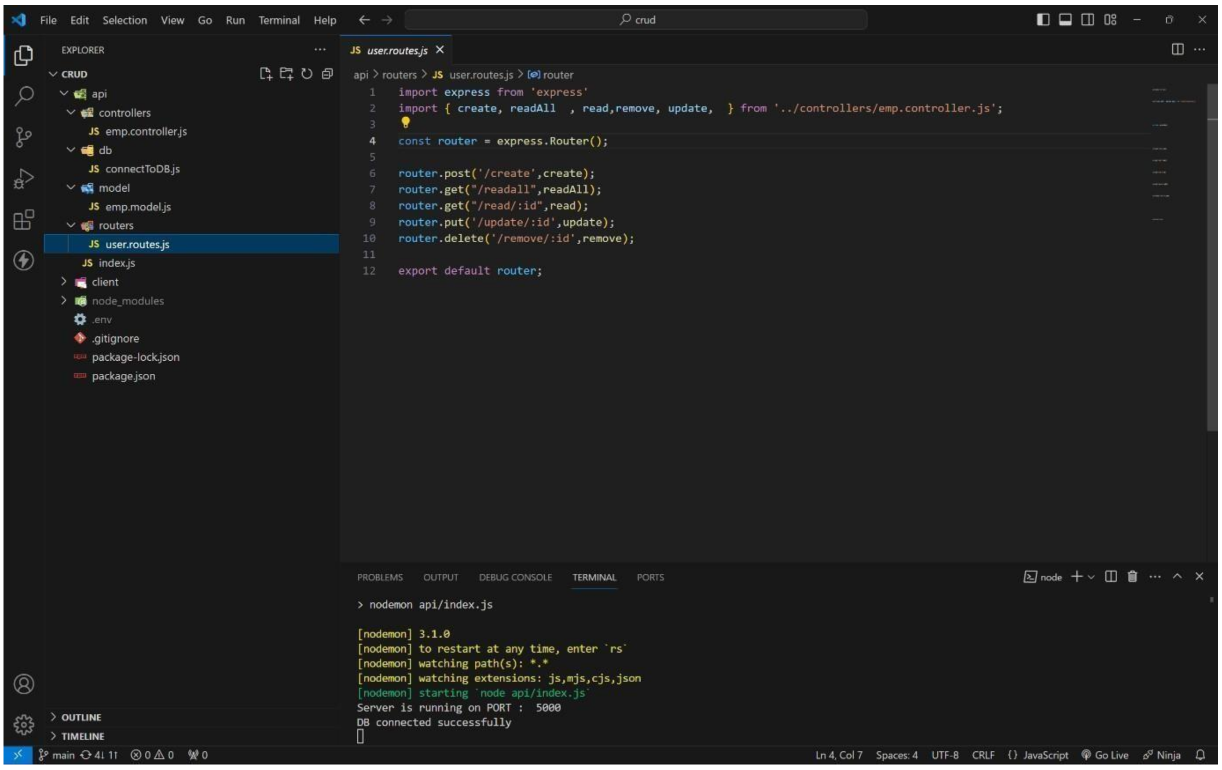
MODEL :



```
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;
```

```
> nodemon api/index.js

[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
```

ROUTES:

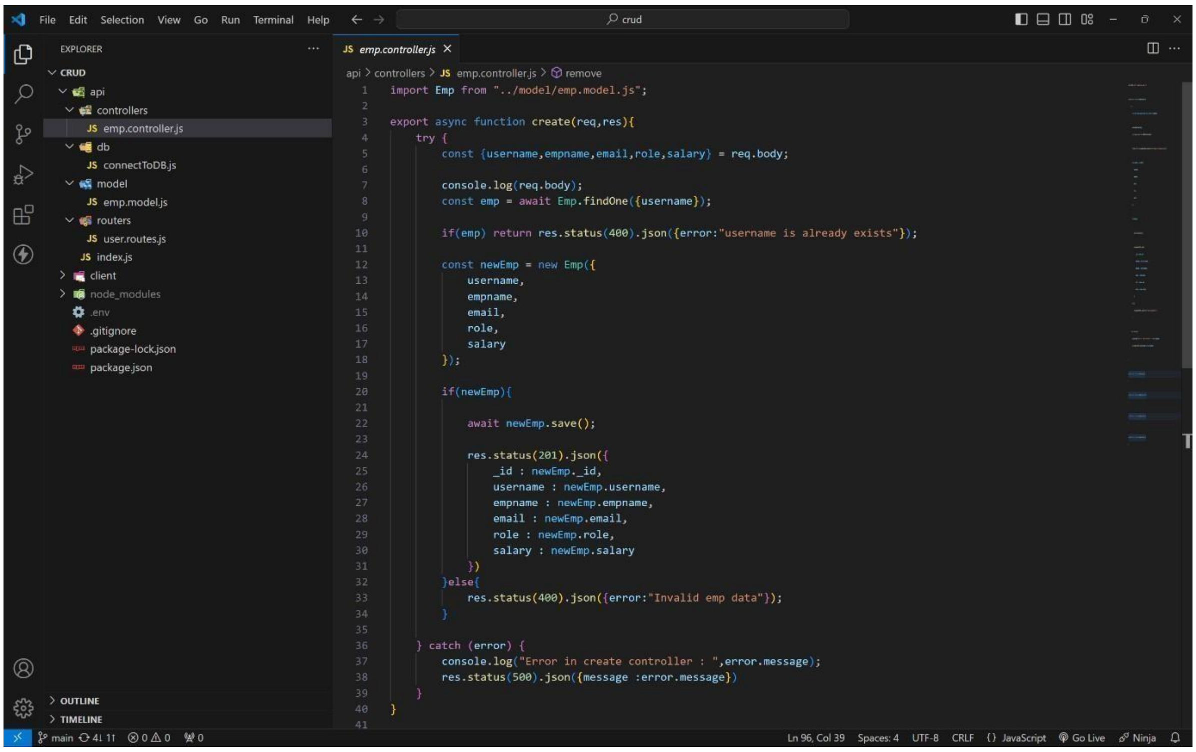


```
import express from 'express'
import { create, readAll  , read,remove, update,  } from '../controllers/emp.controller.js';

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;
```

```
> nodemon api/index.js

[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
```

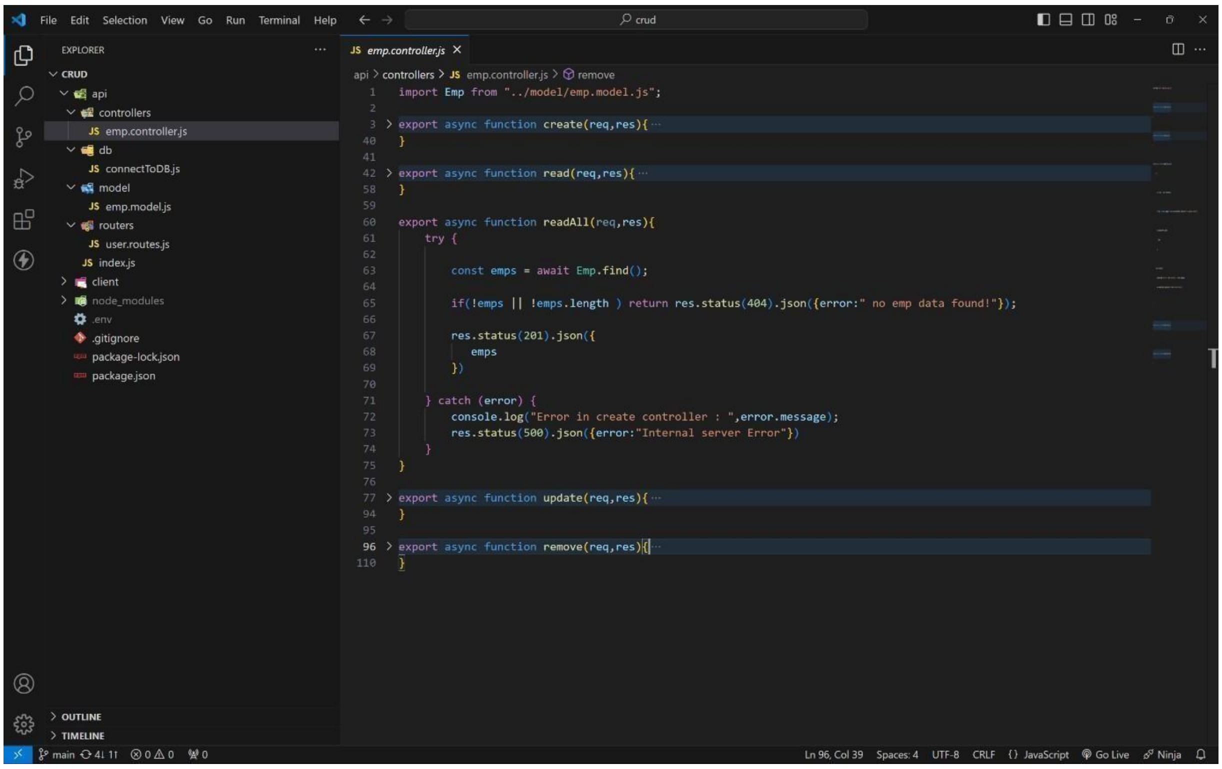
CONTROLLERS :

CREATE :



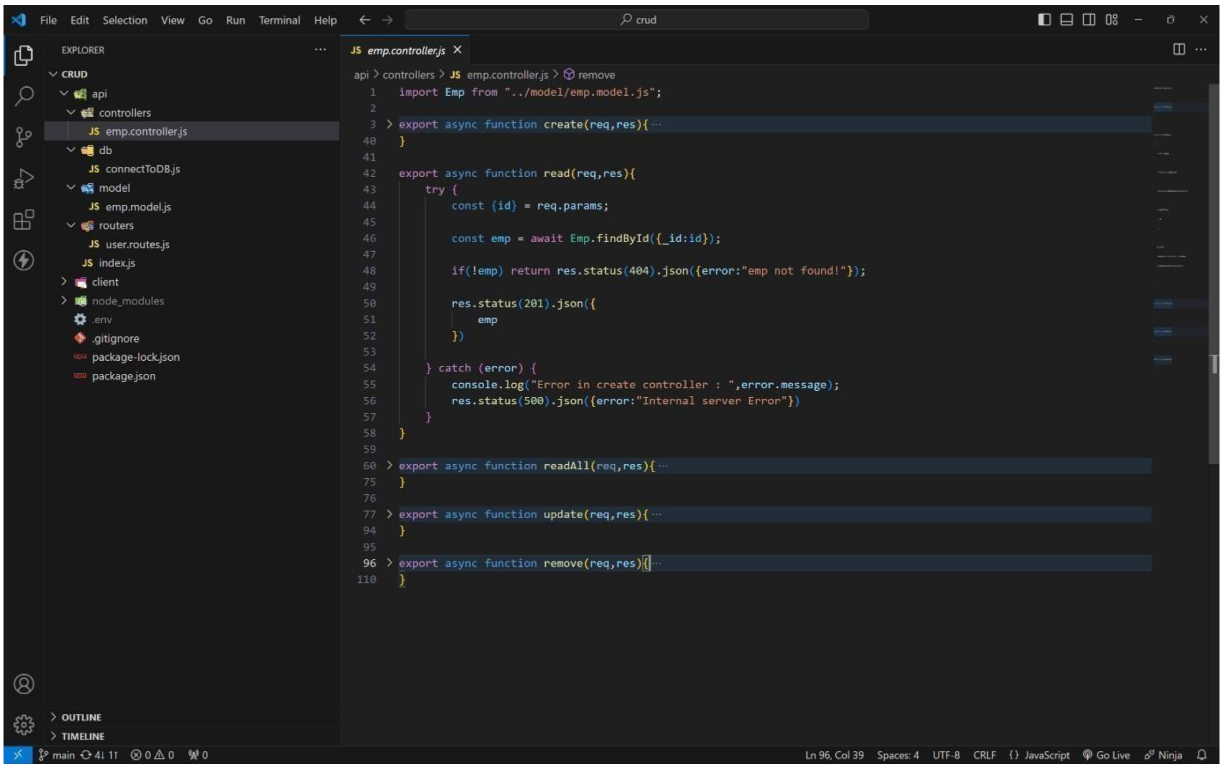
```javascript
import Emp from "../model/emp.model.js";

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})
    }
}

```

READALL:



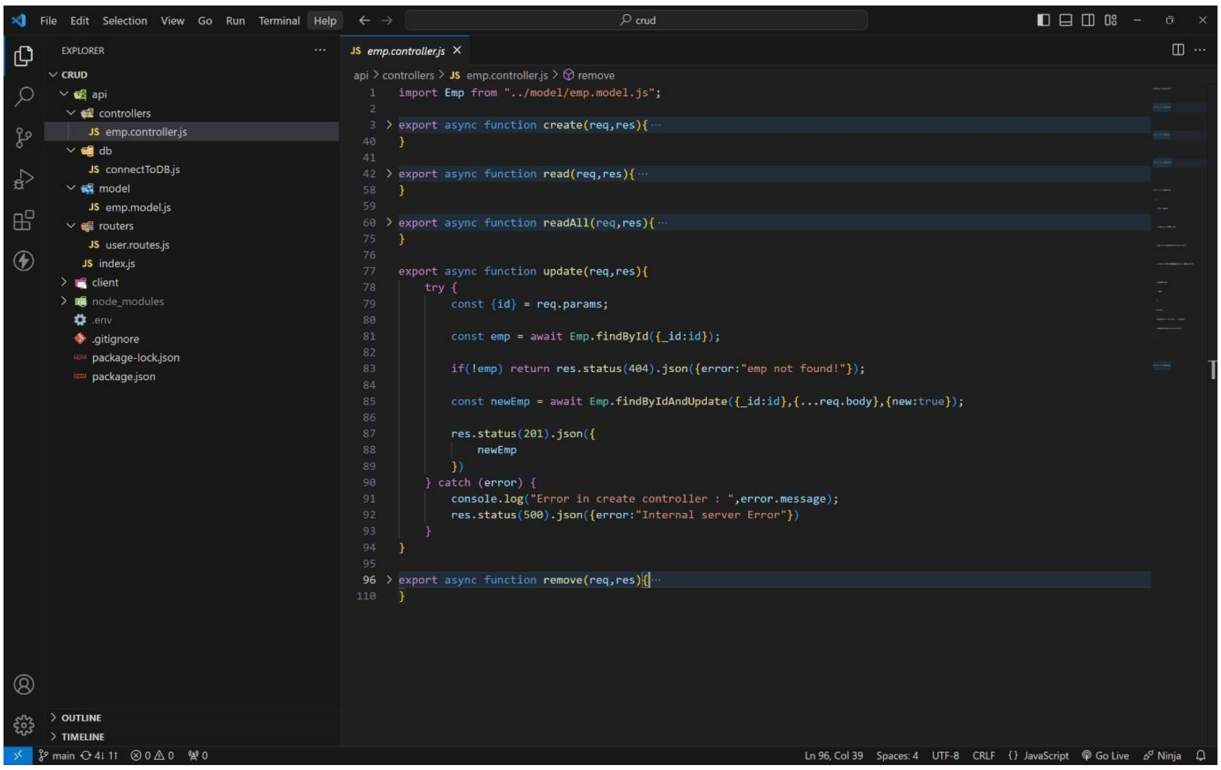
```javascript
import Emp from "../model/emp.model.js";

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

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

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

READONE :



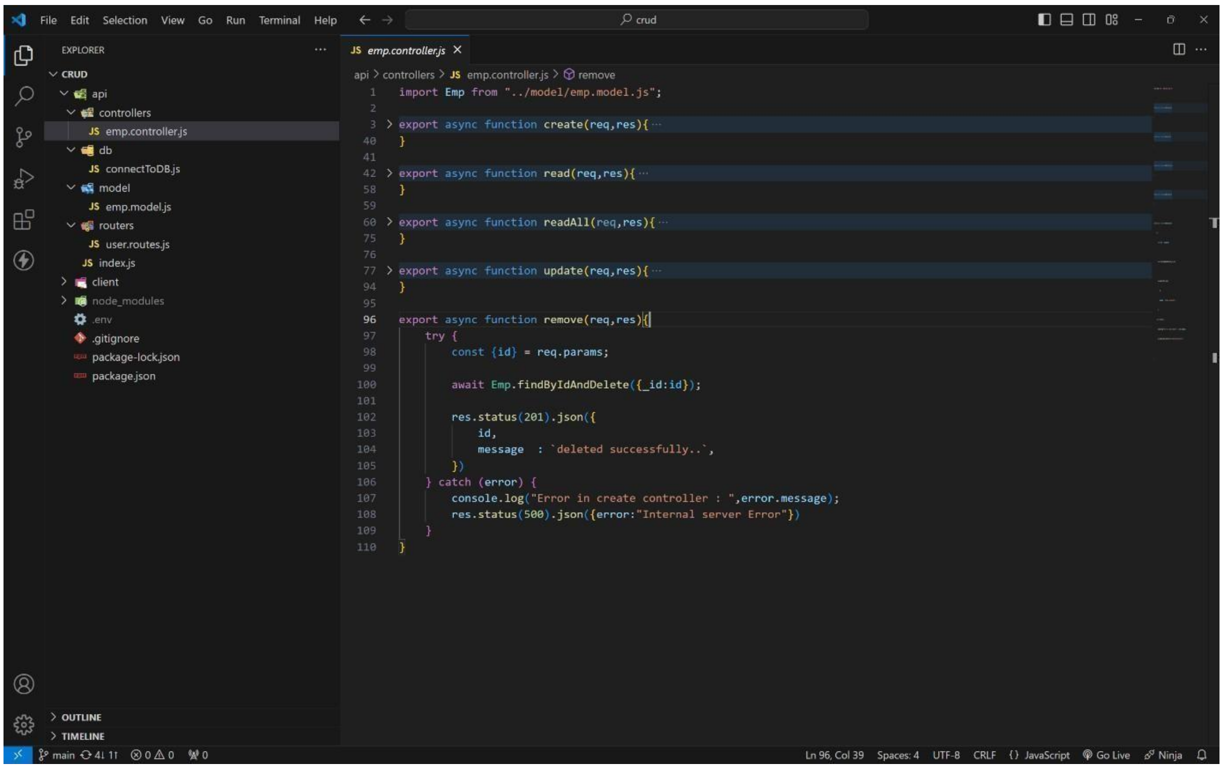
```
import Emp from "../model/emp.model.js";

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){···
}
```

UPDATE :

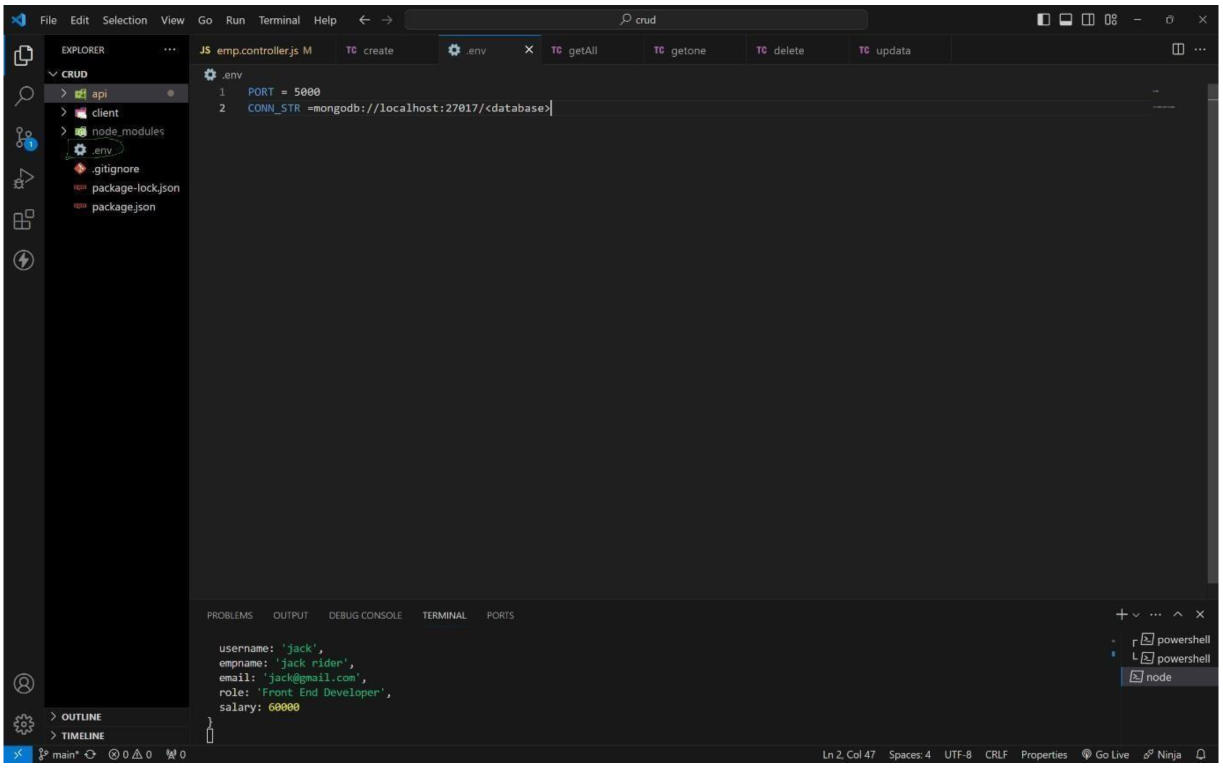


```
import Emp from "../model/emp.model.js";

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){···
}
```

DELETE :



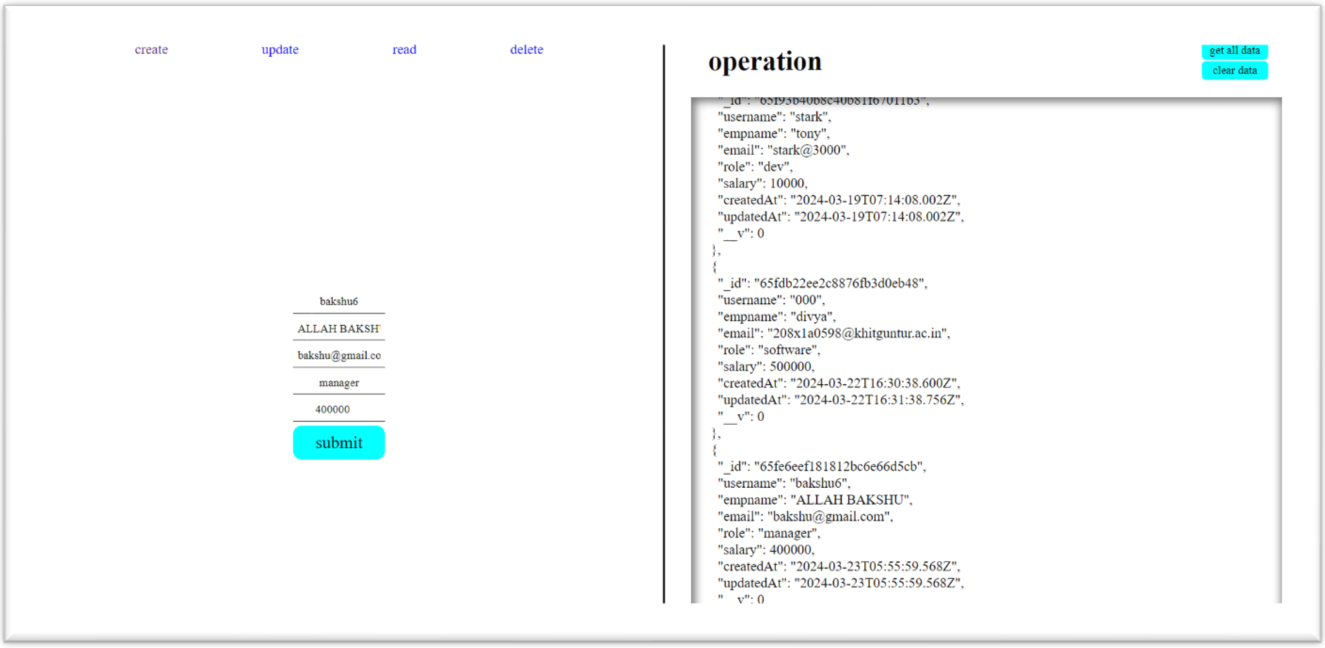
```
import Emp from "../model/emp.model.js";

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
}
```

OUTPUT :



operation

```
  "_id": "65f93b40b8c40b81f6701163",
  "username": "stark",
  "empname": "tony",
  "email": "stark@3000",
  "role": "dev",
  "salary": 10000,
  "createdAt": "2024-03-19T07:14:08.002Z",
  "updatedAt": "2024-03-19T07:14:08.002Z",
  "__v": 0
},
{
  "_id": "65fdb22ee2c8876fb3d0eb48",
  "username": "000",
  "empname": "divya",
  "email": "208x1a0598@khitguntur.ac.in",
  "role": "software",
  "salary": 500000,
  "createdAt": "2024-03-22T16:30:38.600Z",
  "updatedAt": "2024-03-22T16:31:38.756Z",
  "__v": 0
},
{
  "_id": "65fe6eef181812bc6e66d5cb",
  "username": "bakshu6",
  "empname": "ALLAH BAKSHU",
  "email": "bakshu@gmail.com",
  "role": "manager",
  "salary": 400000,
  "createdAt": "2024-03-23T05:55:59.568Z",
  "updatedAt": "2024-03-23T05:55:59.568Z",
  "__v": 0
```