

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

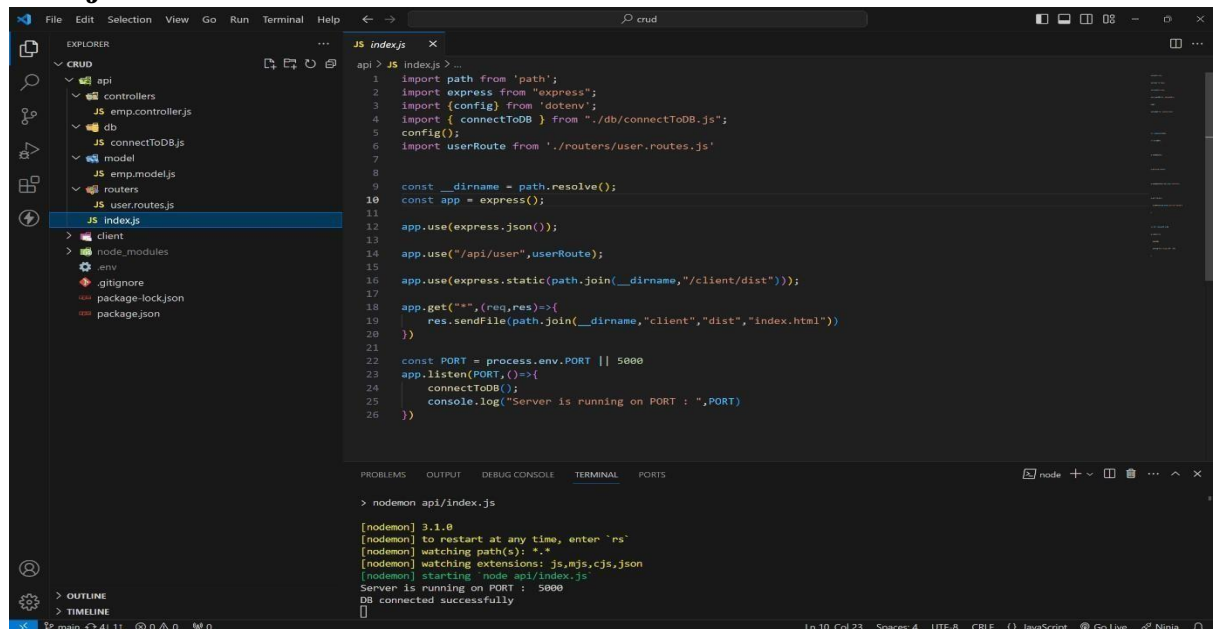
NAME : KOTHA VENKATA SATYA SAI SURYA

EMAIL ID : 208X1a05b0@khitguntur.ac.in

PHONE NO : 7386675365

ROLL NO : 208X1A05B0 (CSE)

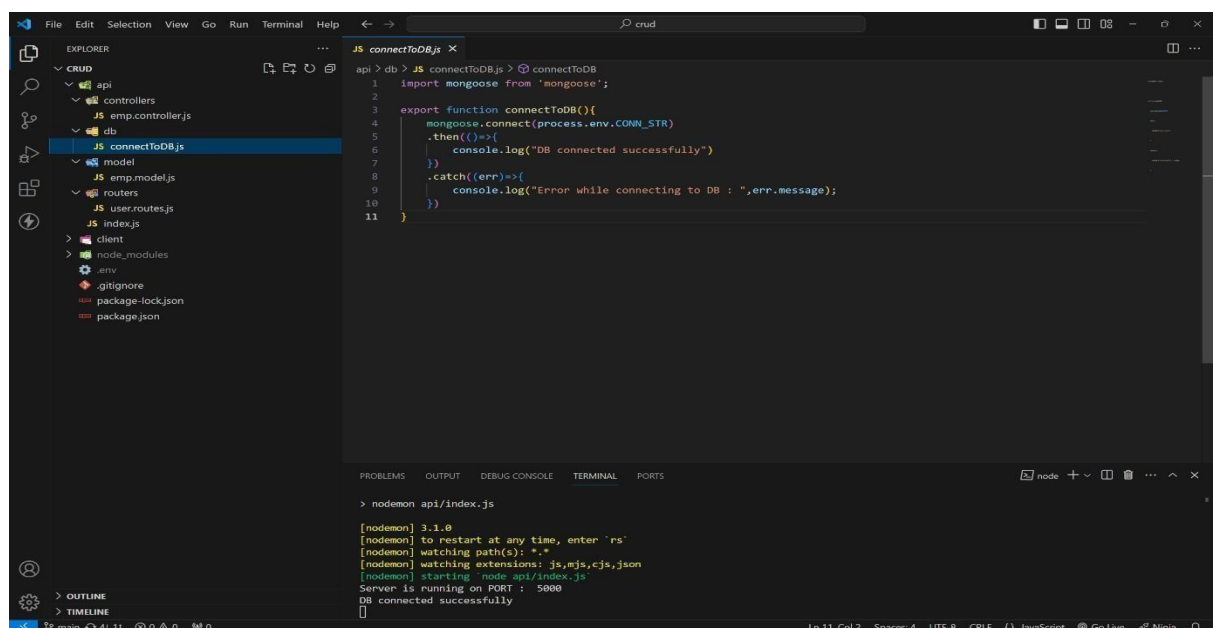
COLLEGE : KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY, GUNTUR **source code :**
index.js file :



```
api > JS index.js >
1 import path from 'path';
2 import express from 'express';
3 import {config} from 'dotenv';
4 import {connectToDB} from './db/connectToDB.js';
5 config();
6 import userRoute from './routes/user.routes.js'
7
8
9 const __dirname = path.resolve();
10 const app = express();
11
12 app.use(express.json());
13
14 app.use("/api/user",userRoute);
15
16 app.use(express.static(path.join(__dirname,"/client/dist")));
17
18 app.get("",(req,res)=>{
19   res.sendFile(path.join(__dirname,"client","dist","index.html"))
20 })
21
22 const PORT = process.env.PORT || 5000
23 app.listen(PORT,()=>{
24   connectToDB();
25   console.log("Server is running on PORT : ",PORT)
26 })
```

```
> 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
```

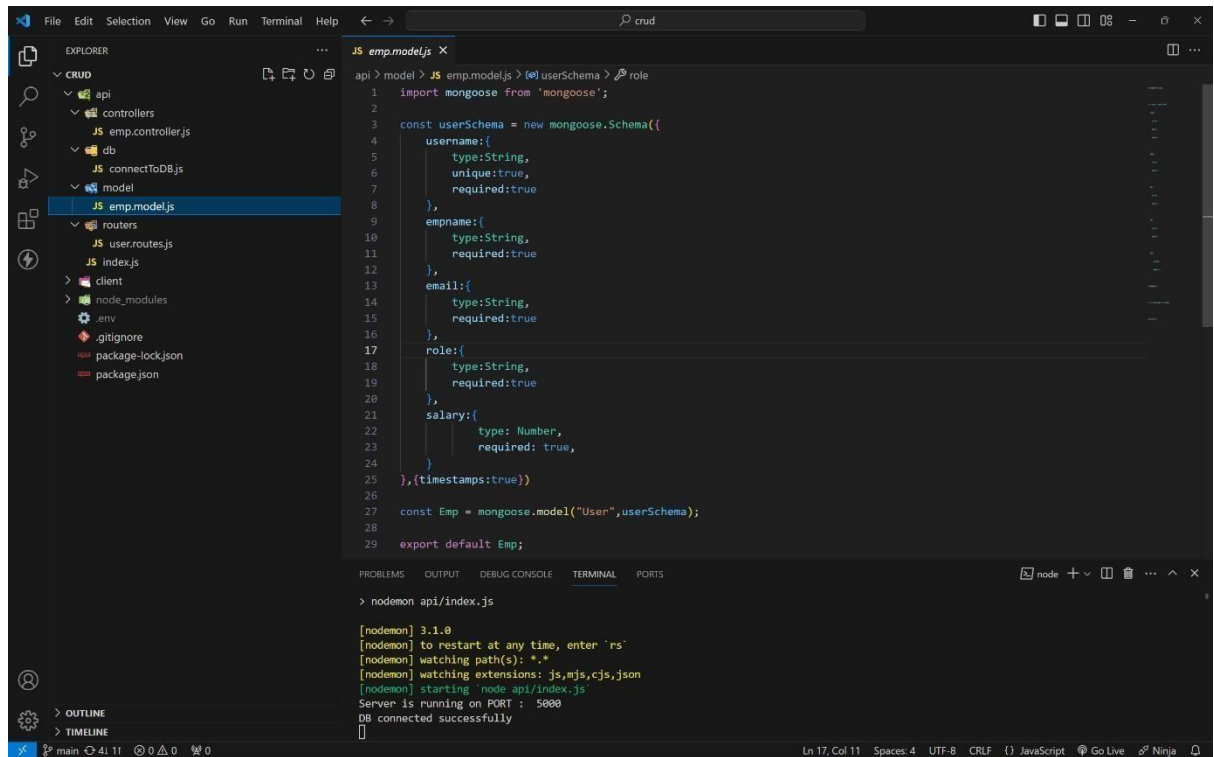
MONGODB CONNECTION :



```
api > db > JS connectToDB.js > connectToDB
1 import mongoose from 'mongoose';
2
3 export function connectToDB(){
4   mongoose.connect(process.env.CONN_STR)
5   .then(()=>{
6     console.log("DB connected successfully")
7   })
8   .catch((err)=>{
9     console.log("Error while connecting to DB : ",err.message);
10  })
11 }
```

```
> 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 :

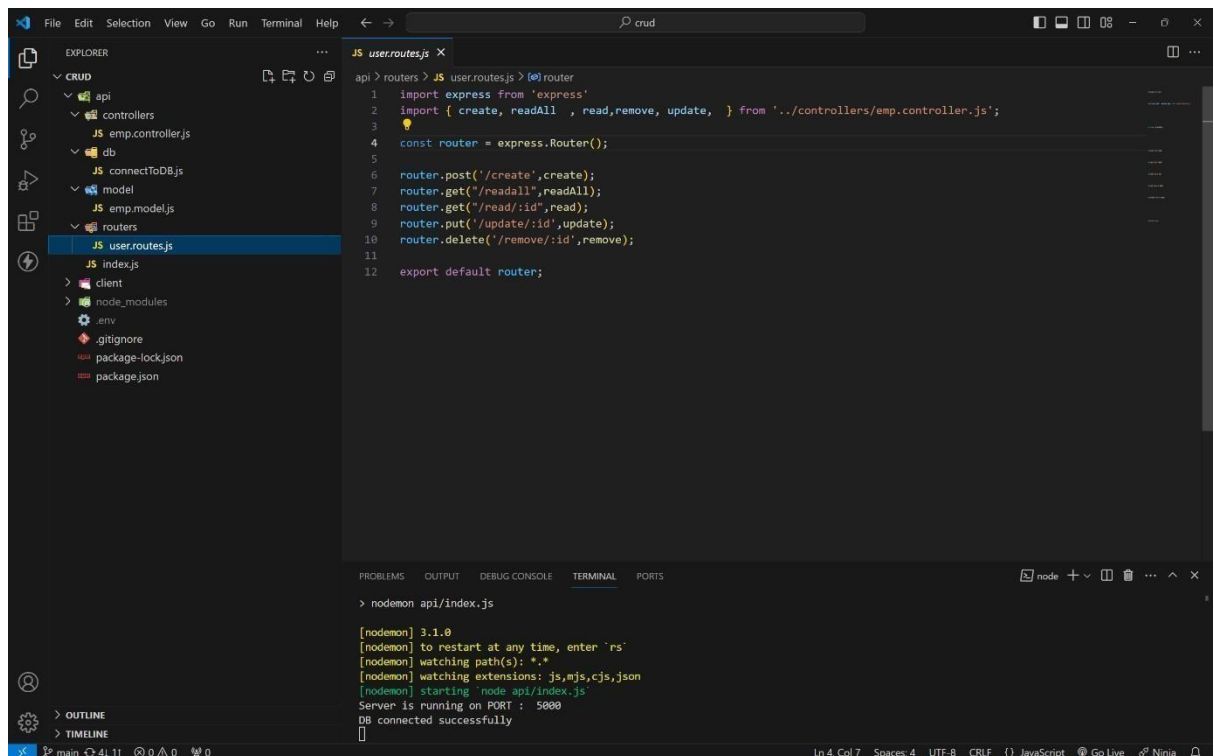


The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with folders like 'api', 'controllers', 'db', 'model', 'routers', and 'client'. The file 'emp.model.js' is selected in the 'model' folder. The editor displays the code for 'emp.model.js', which defines a Mongoose schema for a user with fields: username, empname, email, role, and salary. The schema is named 'User' and the model is named 'Emp'. The terminal at the bottom shows the command 'nodemon api/index.js' and the output indicating that the server is running on port 5000 and the database is connected successfully.

```
api > model > JS emp.model.js > @userSchema > role
1 import mongoose from 'mongoose';
2
3 const userSchema = new mongoose.Schema({
4   username: {
5     type: String,
6     unique: true,
7     required: true
8   },
9   empname: {
10    type: String,
11    required: true
12  },
13   email: {
14    type: String,
15    required: true
16  },
17   role: {
18    type: String,
19    required: true
20  },
21   salary: {
22    type: Number,
23    required: true,
24  }
25 }, { timestamps: true })
26
27 const Emp = mongoose.model("User", userSchema);
28
29 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:



The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with folders like 'api', 'controllers', 'db', 'model', 'routers', and 'client'. The file 'user.routes.js' is selected in the 'routers' folder. The editor displays the code for 'user.routes.js', which defines an Express router with routes for create, readAll, read, update, and delete. The terminal at the bottom shows the command 'nodemon api/index.js' and the output indicating that the server is running on port 5000 and the database is connected successfully.

```
api > routers > JS user.routes.js > @router
1 import express from 'express'
2 import { create, readAll, read, remove, update, } from '../controllers/emp.controller.js';
3
4 const router = express.Router();
5
6 router.post('/create', create);
7 router.get("/readall", readAll);
8 router.get("/read/:id", read);
9 router.put('/update/:id', update);
10 router.delete('/remove/:id', remove);
11
12 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 :

The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with folders like api, controllers, db, model, routers, and client. The file emp.controller.js is selected. The main editor shows the implementation of the create function in emp.controller.js. The code is as follows:

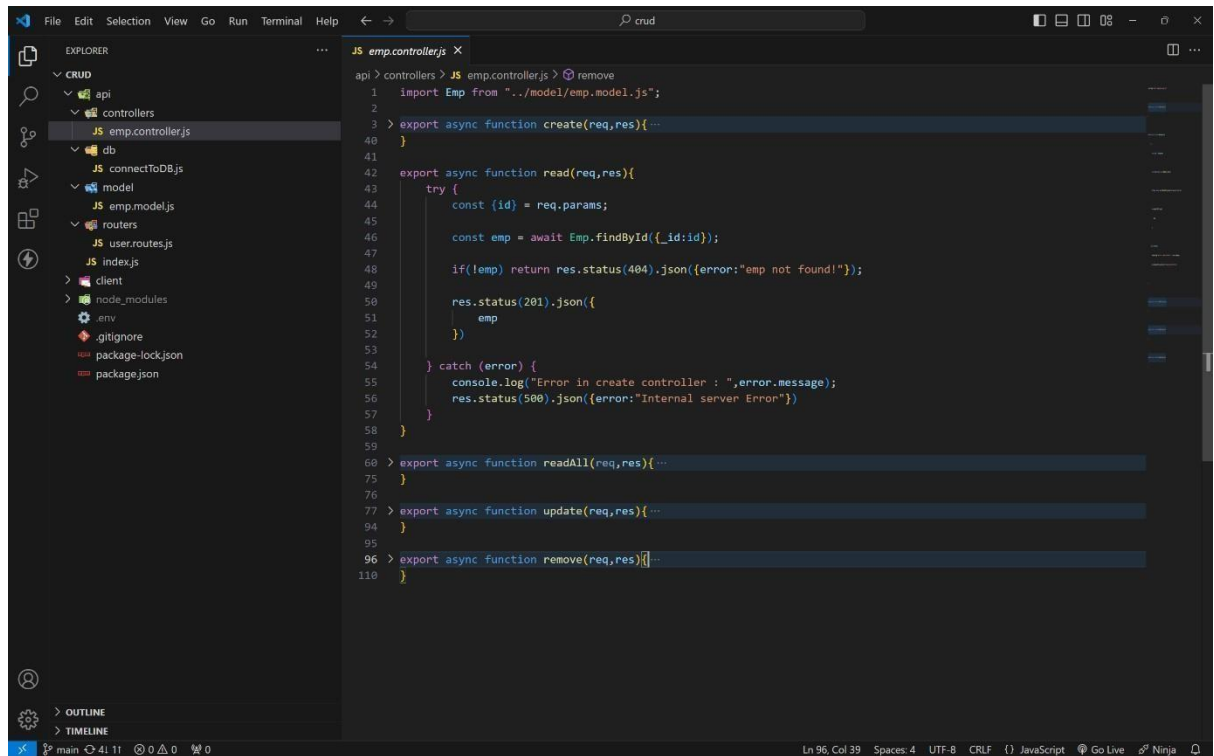
```
api > controllers > JS emp.controller.js > remove
1  import Emp from "../model/emp.model.js";
2
3  export async function create(req,res){
4      try {
5          const {username,empname,email,role,salary} = req.body;
6
7          console.log(req.body);
8          const emp = await Emp.findOne({username});
9
10         if(emp) return res.status(400).json({error:"username is already exists"});
11
12         const newEmp = new Emp({
13             username,
14             empname,
15             email,
16             role,
17             salary
18         });
19
20         if(newEmp){
21             await newEmp.save();
22
23             res.status(201).json({
24                 _id : newEmp._id,
25                 username : newEmp.username,
26                 empname : newEmp.empname,
27                 email : newEmp.email,
28                 role : newEmp.role,
29                 salary : newEmp.salary
30             })
31         }else{
32             res.status(400).json({error:"Invalid emp data"});
33         }
34     } catch (error) {
35         console.log("Error in create controller : ",error.message);
36         res.status(500).json({message :error.message})
37     }
38 }
39
40
41
```

READALL:

The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with folders like api, controllers, db, model, routers, and client. The file emp.controller.js is selected. The main editor shows the implementation of the readAll function in emp.controller.js. The code is as follows:

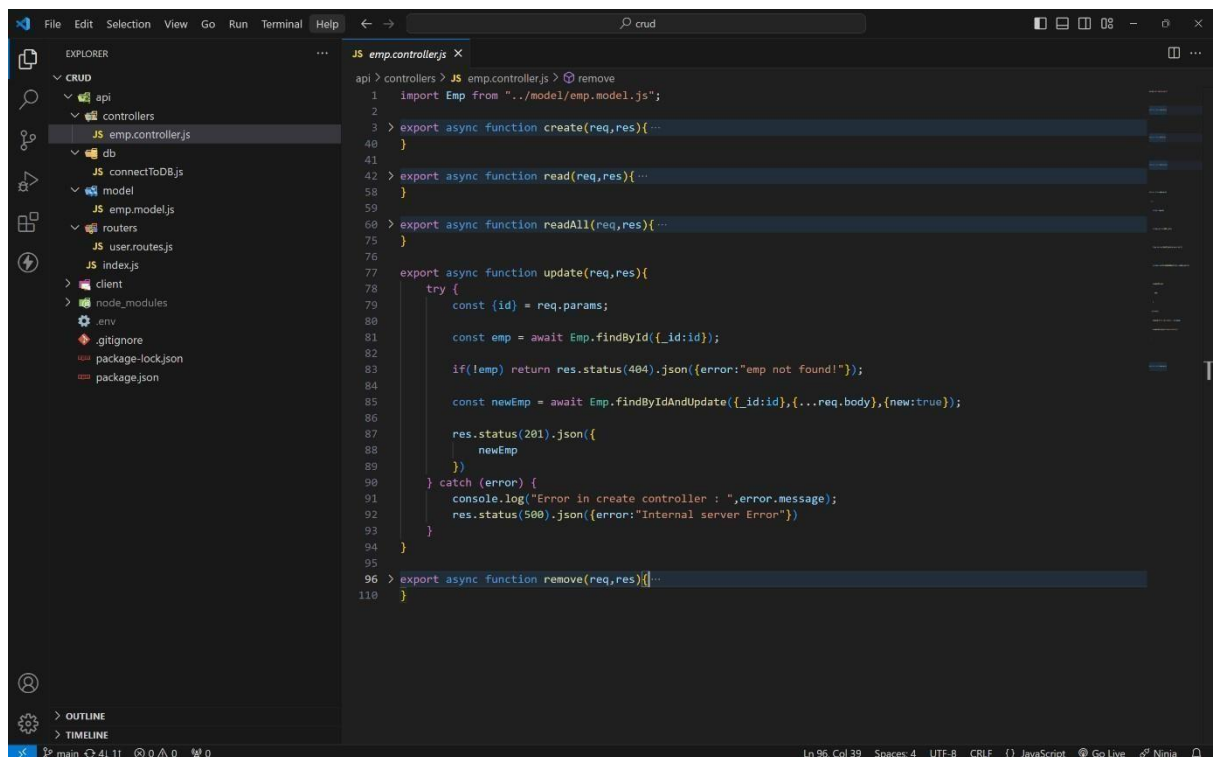
```
api > controllers > JS emp.controller.js > remove
1  import Emp from "../model/emp.model.js";
2
3  export async function create(req,res){ ...
40 }
41
42 export async function read(req,res){ ...
58 }
59
60 export async function readAll(req,res){
61     try {
62         const emps = await Emp.find();
63
64         if(!emps || !emps.length ) return res.status(404).json({error:" no emp data found!"});
65
66         res.status(201).json({
67             emps
68         })
69     } catch (error) {
70         console.log("Error in create controller : ",error.message);
71         res.status(500).json({error:"Internal server Error"})
72     }
73 }
74
75
76
77 export async function update(req,res){ ...
94 }
95
96 export async function remove(req,res){ ...
110 }
```

READONE :



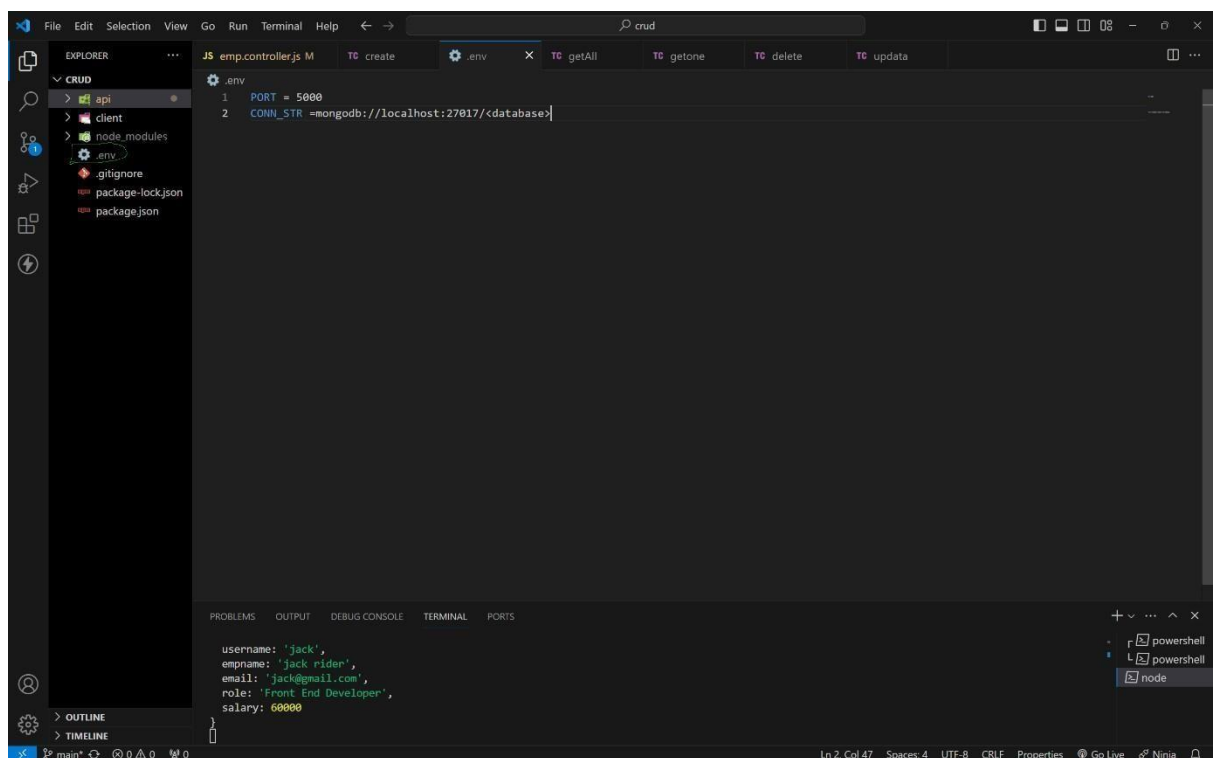
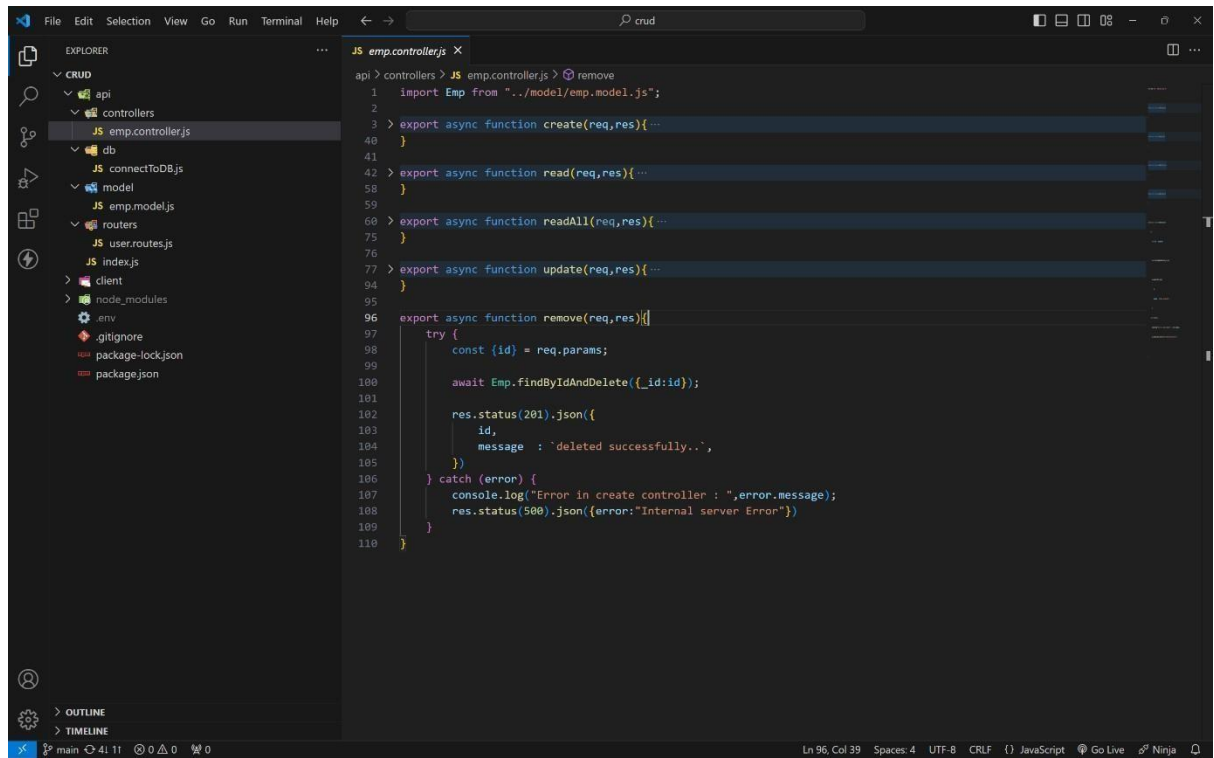
```
api > controllers > JS emp.controller.js > remove
1 import Emp from "../model/emp.model.js";
2
3 > export async function create(req,res){ ...
40 }
41
42 export async function read(req,res){
43   try {
44     const {id} = req.params;
45
46     const emp = await Emp.findById({_id:id});
47
48     if(!emp) return res.status(404).json({error:"emp not found!"});
49
50     res.status(201).json({
51       emp
52     })
53   } catch (error) {
54     console.log("Error in create controller : ",error.message);
55     res.status(500).json({error:"Internal server Error"})
56   }
57 }
58
59
60 > export async function readAll(req,res){...
75 }
76
77 > export async function update(req,res){ ...
94 }
95
96 > export async function remove(req,res){ ...
110 }
```

UPDATE :



```
api > controllers > JS emp.controller.js > remove
1 import Emp from "../model/emp.model.js";
2
3 > export async function create(req,res){ ...
40 }
41
42 > export async function read(req,res){ ...
58 }
59
60 > export async function readAll(req,res){...
75 }
76
77 export async function update(req,res){
78   try {
79     const {id} = req.params;
80
81     const emp = await Emp.findById({_id:id});
82
83     if(!emp) return res.status(404).json({error:"emp not found!"});
84
85     const newEmp = await Emp.findByIdAndUpdate({_id:id},{...req.body},{new:true});
86
87     res.status(201).json({
88       newEmp
89     })
90   } catch (error) {
91     console.log("Error in create controller : ",error.message);
92     res.status(500).json({error:"Internal server Error"})
93   }
94 }
95
96 > export async function remove(req,res){ ...
110 }
```

DELETE :



OUTPUT :

createupdatereaddelete

suresh_varma

SURESH

Suresh7@gmail.c

Social Expertize

70000

submit

operation

get all data

clear data

```
"createdAt": "2024-03-23T05:55:59.568Z",
"updatedAt": "2024-03-23T05:55:59.568Z",
"__v": 0
},
{
  "_id": "65ffc8f7d4fef0fcb95c3939",
  "username": "kvss_surya",
  "empname": "KVSS SURYA",
  "email": "Surya7@gmail.com",
  "role": "Technical Expert",
  "salary": 100000,
  "createdAt": "2024-03-24T06:32:23.894Z",
  "updatedAt": "2024-03-24T06:32:23.894Z",
  "__v": 0
},
{
  "_id": "65ffc92bd4fef0fcb95c393c",
  "username": "Mani_reddy",
  "empname": "MANI",
  "email": "Mani7@gmail.com",
  "role": "Software",
  "salary": 49000,
  "createdAt": "2024-03-24T06:33:15.402Z",
  "updatedAt": "2024-03-24T06:33:15.402Z",
  "__v": 0
},
{
  "_id": "65ffc97bd4fef0fcb95c3940",
  "username": "suresh_varma",
  "empname": "SURESH",
  "email": "Suresh7@gmail.com",
  "role": "Social Expertize",
  "salary": 70000,
  "createdAt": "2024-03-24T06:34:35.935Z",
  "updatedAt": "2024-03-24T06:34:35.935Z",
  "__v": 0
}
}
```