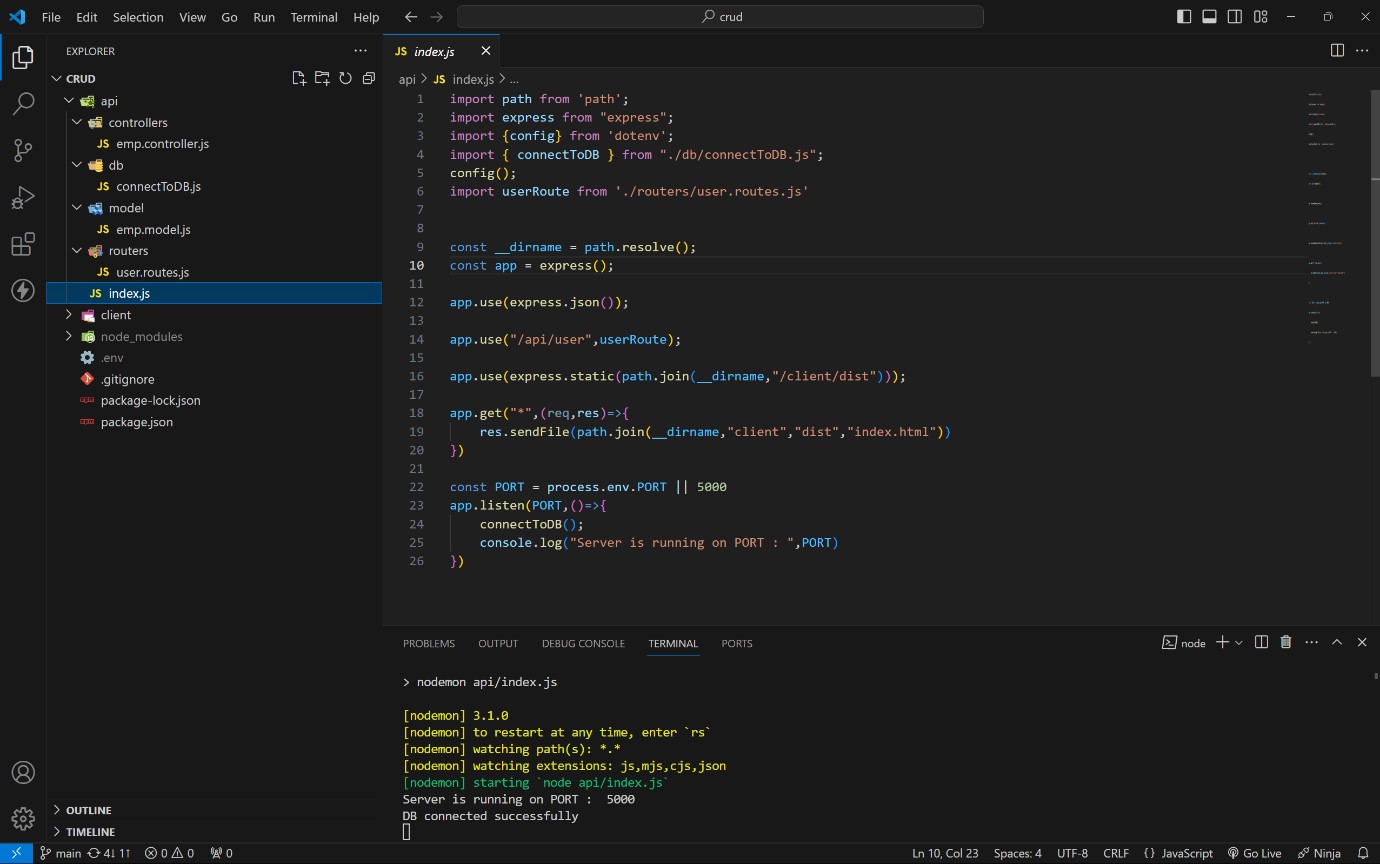
Creating a RESTful API using express.js and creating a database and index in MongoDB.

Name:Maruthi Lankalapalli

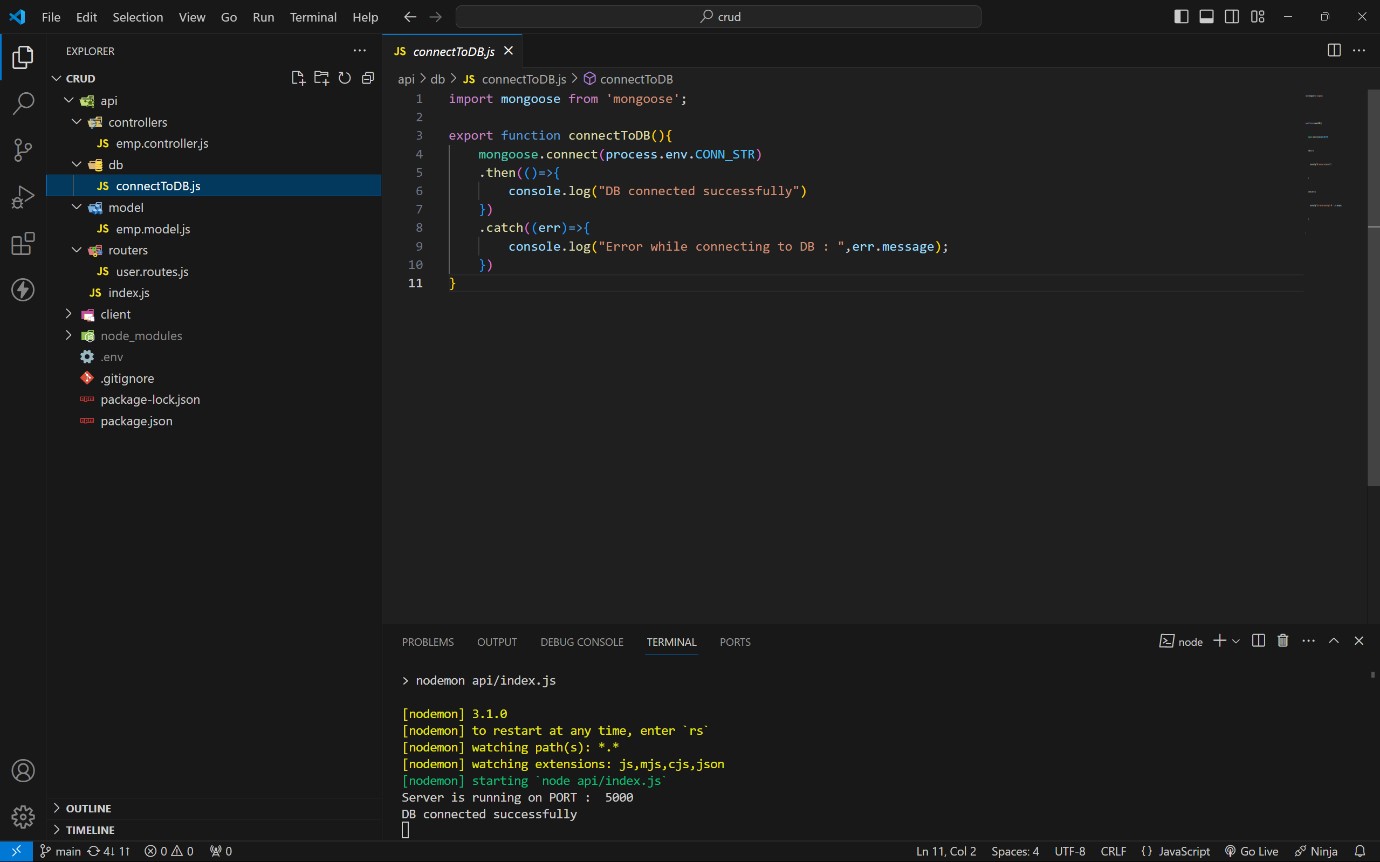
Roll no:20NR1A0559

source code :

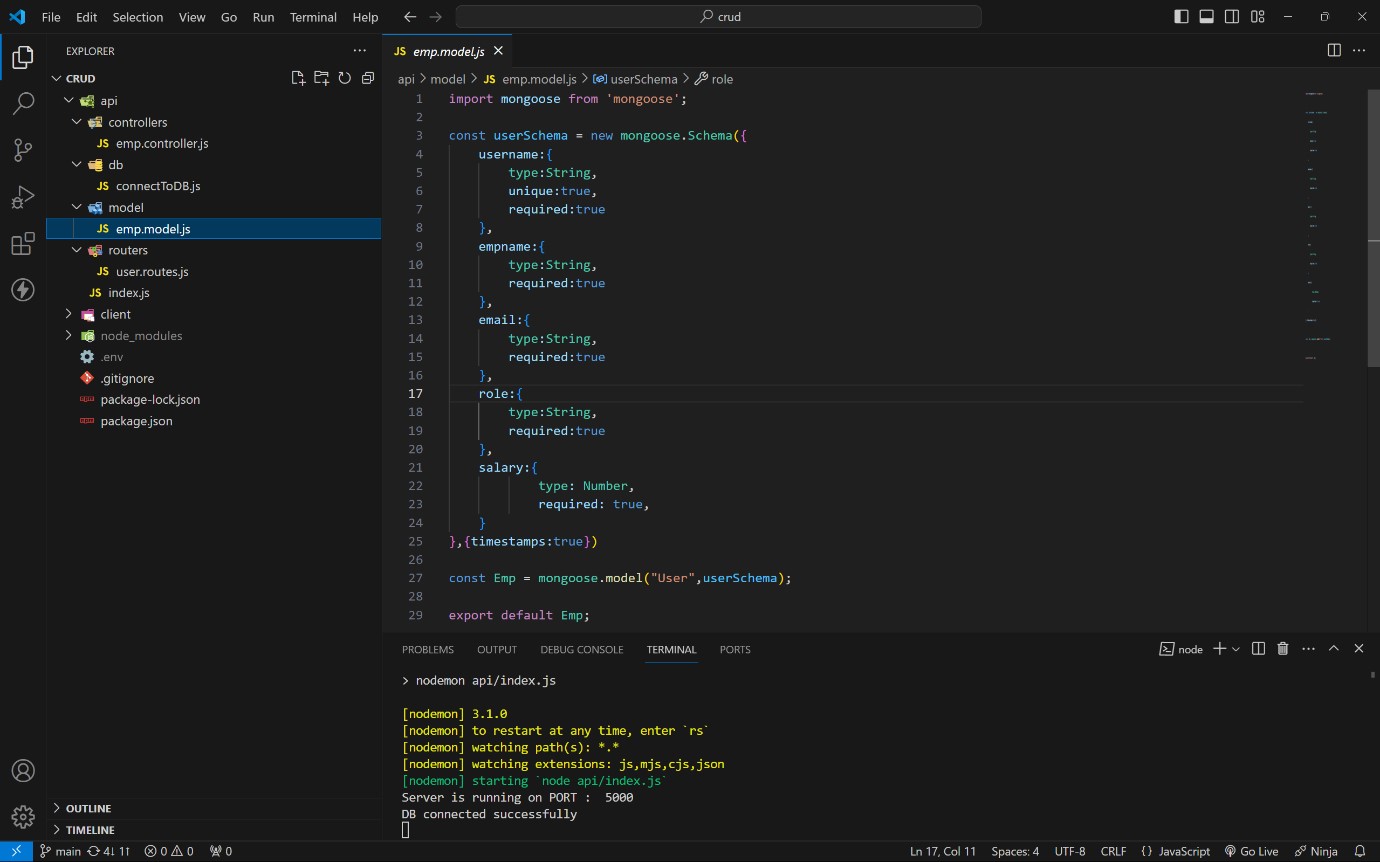
index.js file :



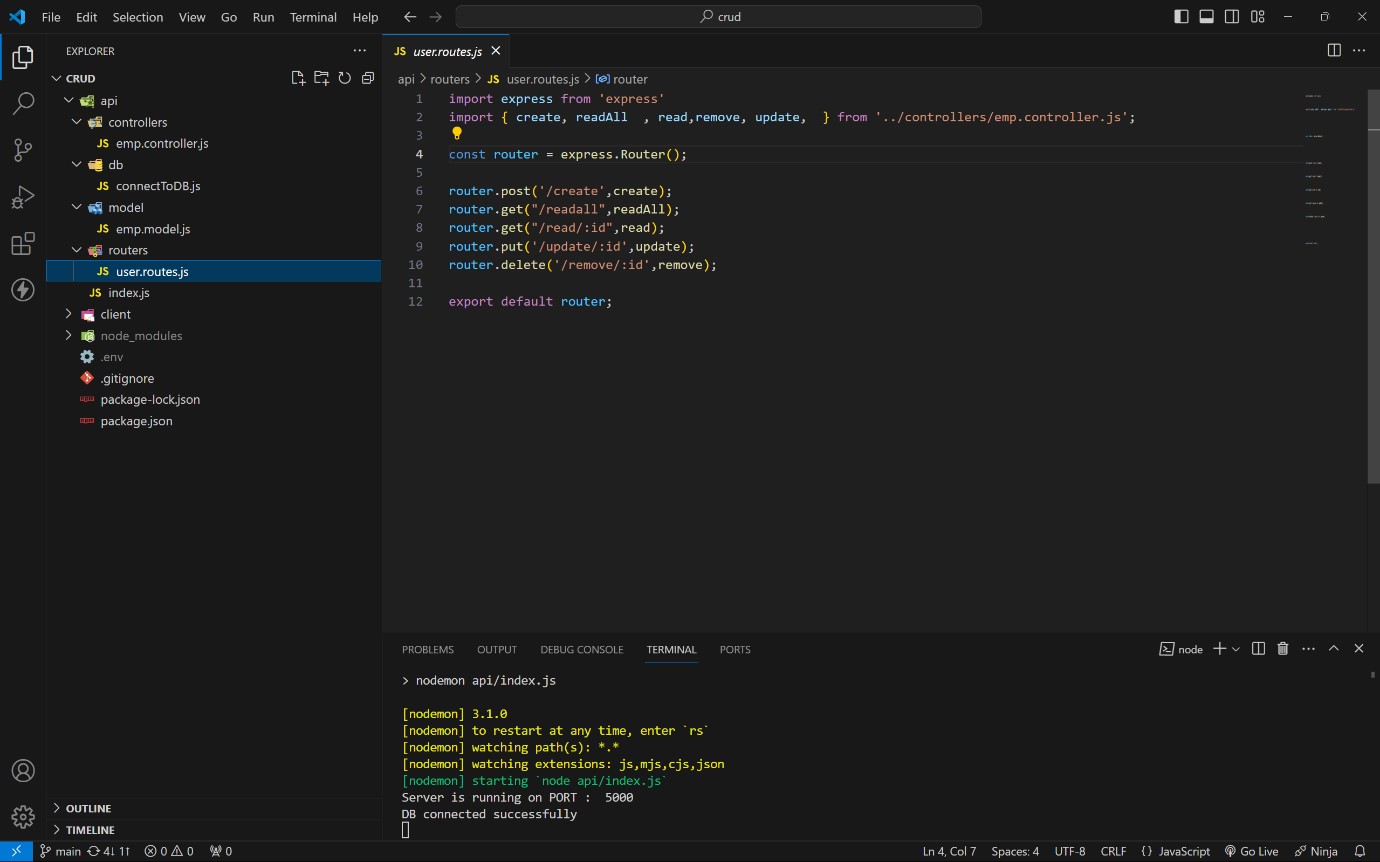
MONGODB CONNECTION :



MODEL :

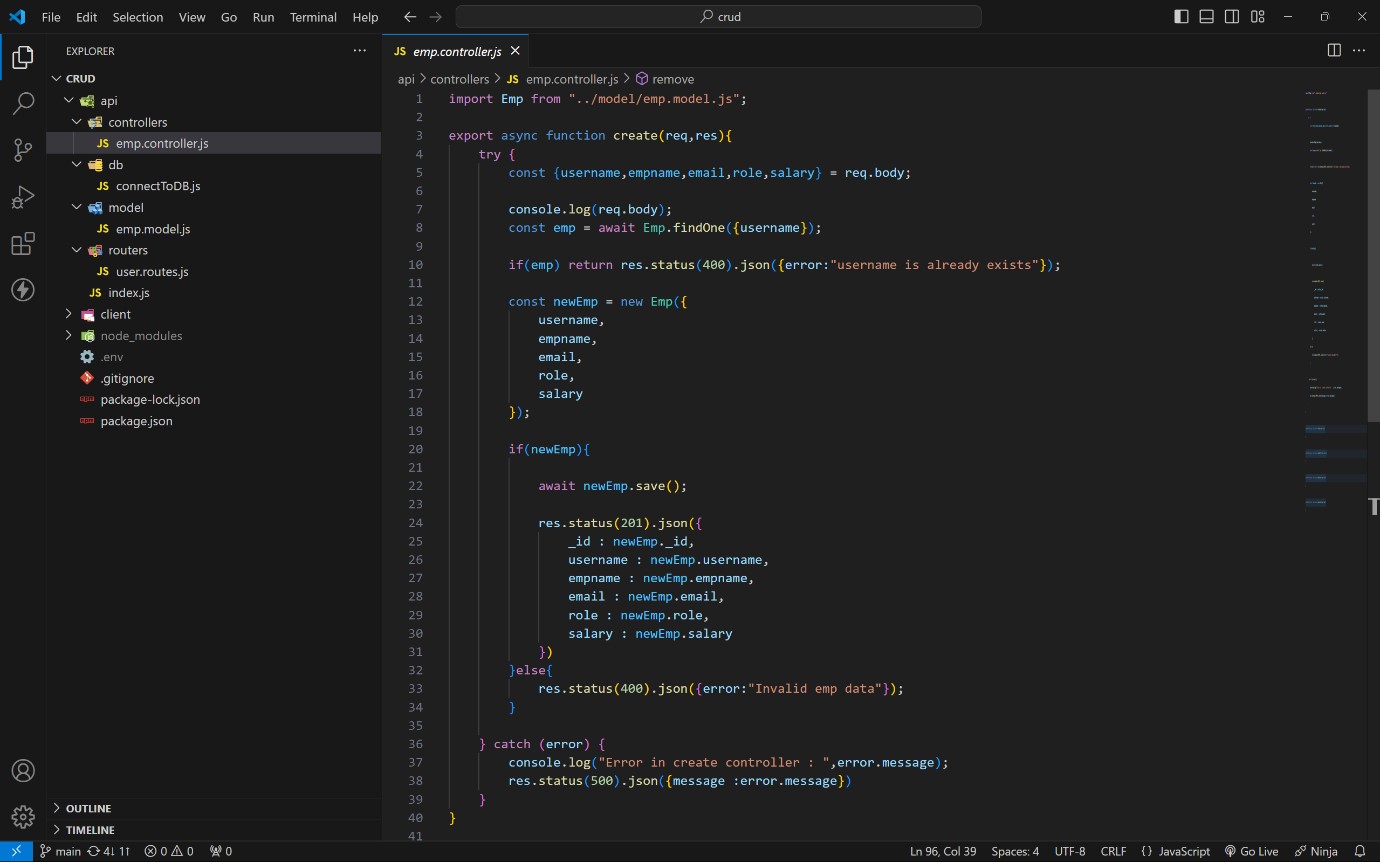


ROUTES:

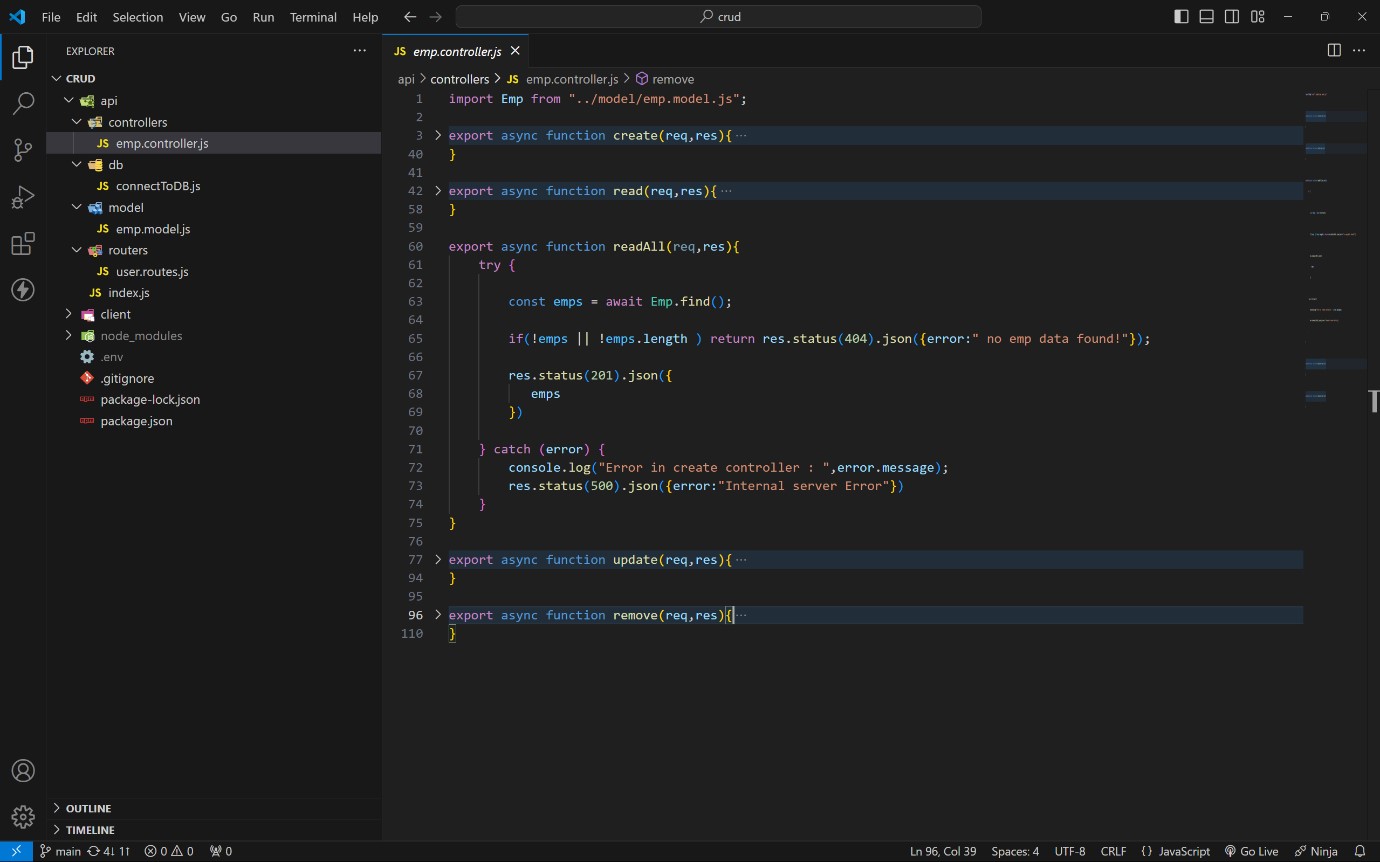


CONTROLLERS :

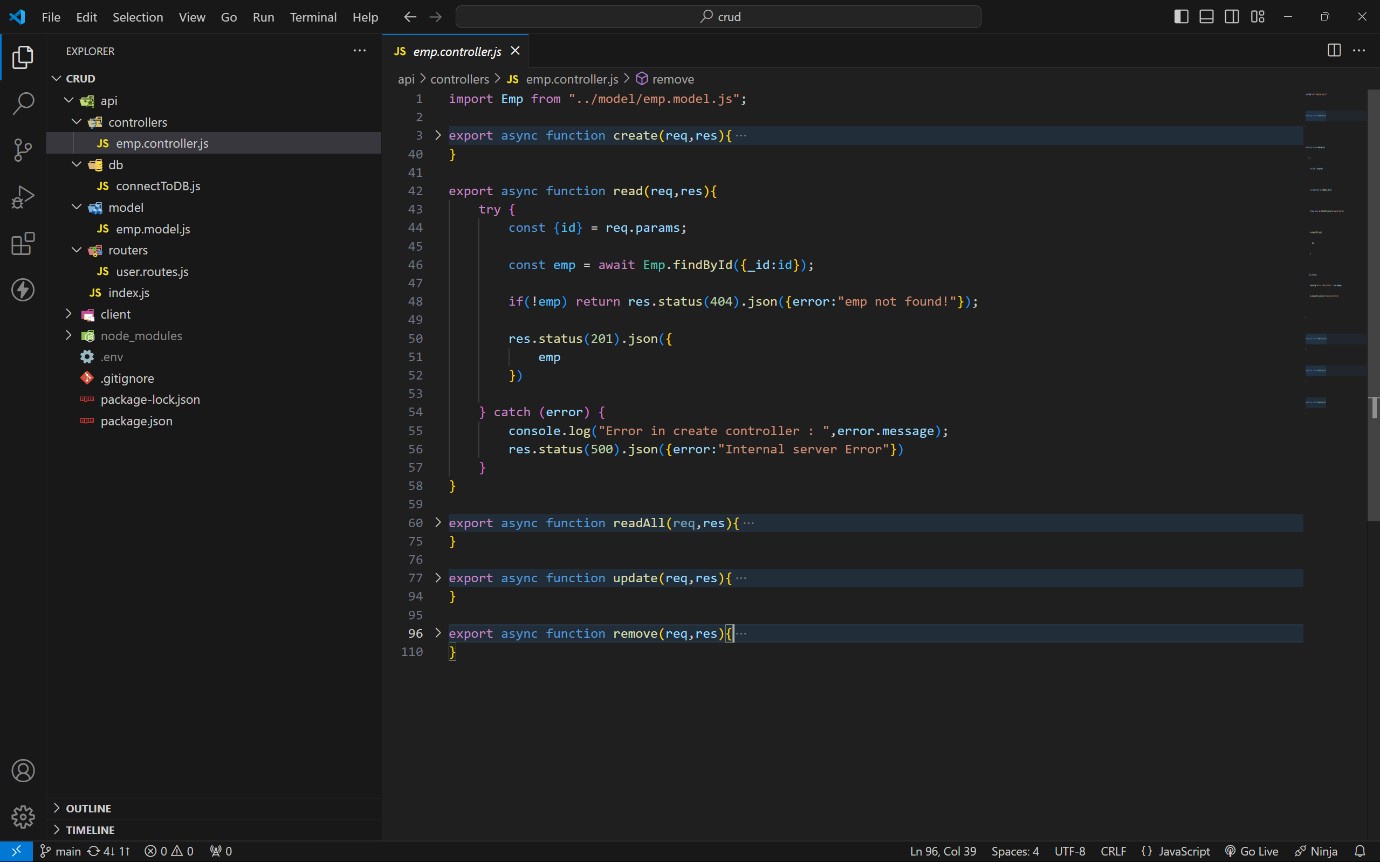
CREATE :



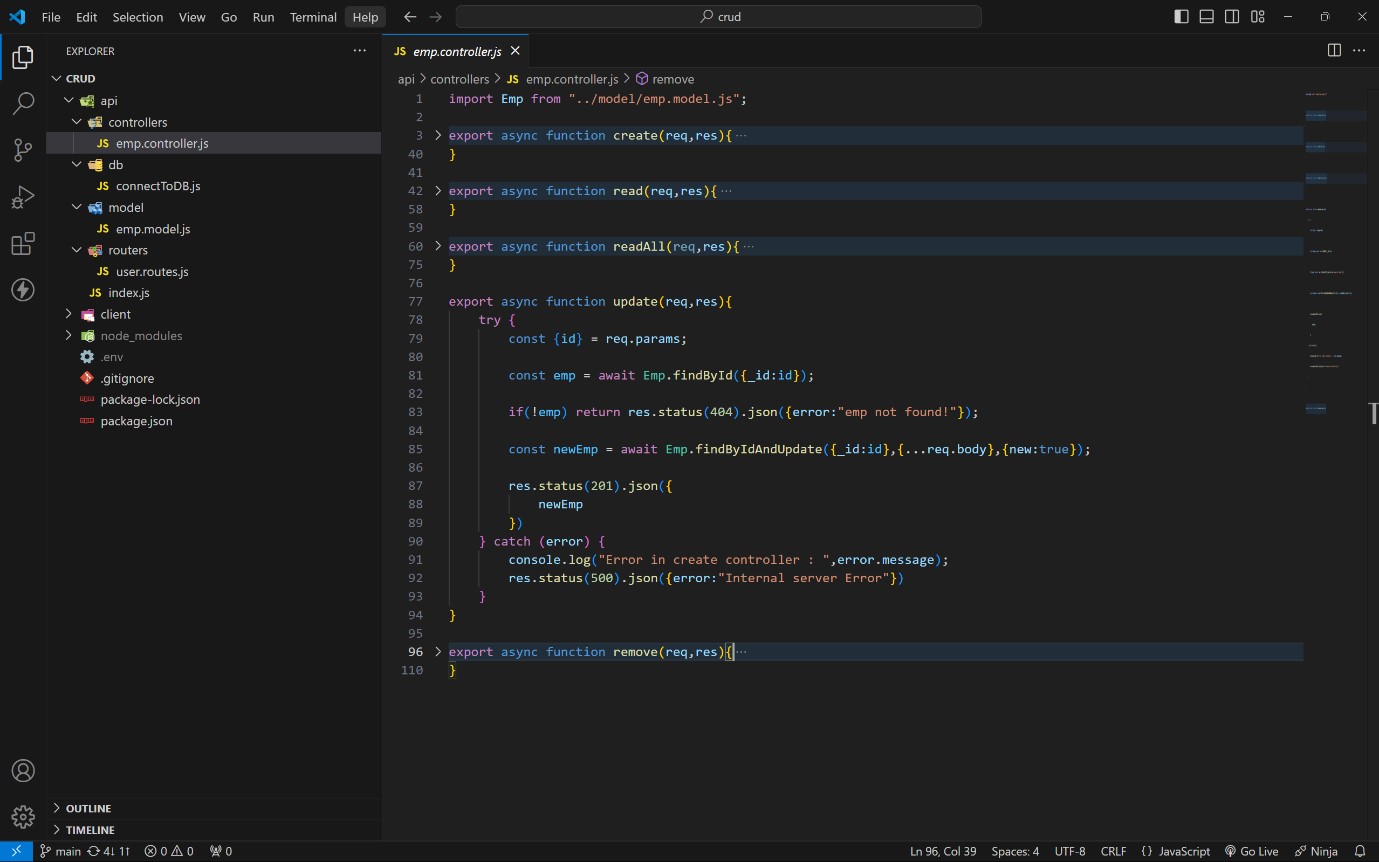
READALL:



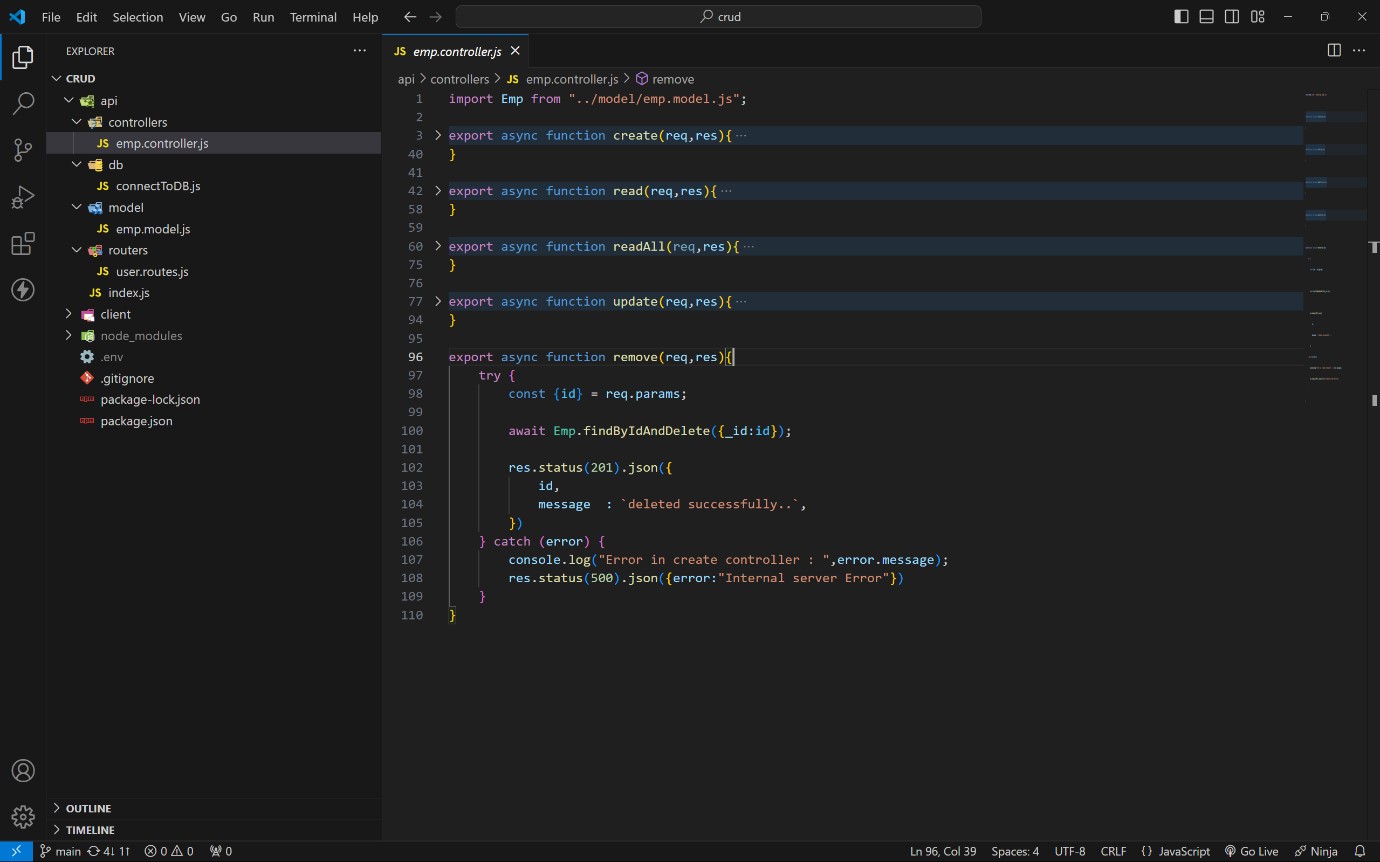
READONE :



UPDATE :



DELETE :



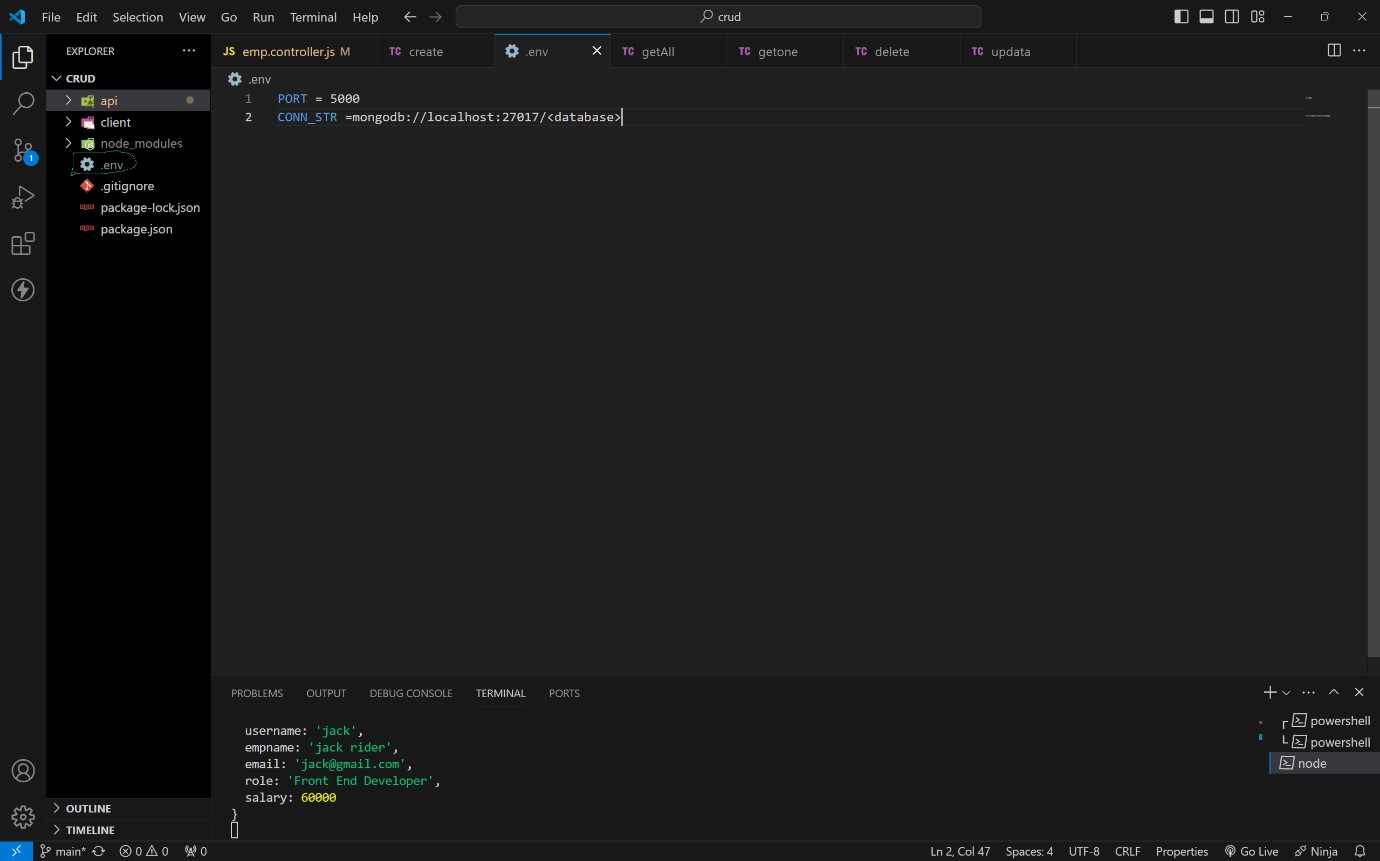
HOW TO RUN ON LOCALLY :

1. . Create a folder as any name.
2. . Open that folder in any code editor (vs code).
3. . Open terminal ( ctrl + ~ ) on code editor.
4. . Now move to crud folder (cd crud in terminal)
5. . Ignore client folder.
6. . Here crud is root folder.
7. . In root folder create a .env file and create a PORT and

CONN\_STR variables and assign value.

ex : PORT = 3000 ( commonly any number between 3000 - 8080).

CONN\_STR = your mongodb\_connection\_string.



--- trouble in above process ? : simply paste this code in .env file .

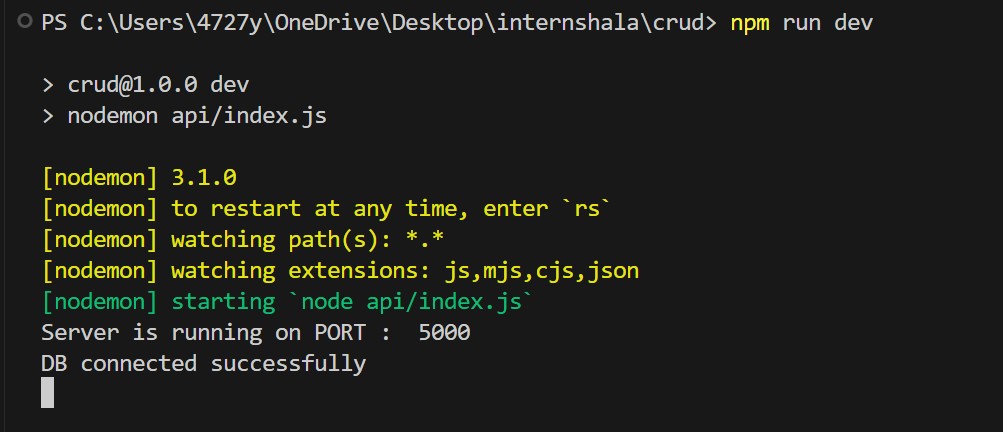
PORT = 5000

CONN\_STR=mongodb+srv://4727yesuraju:rough@cluster0.wbclvtg.mongodb.net /?retryWrites=true&w=majority&appName=Cluster0

1. . After in terminal (in crud folder as root folder) type this command to run server.

npm i (installing all dependencies) npm run dev (to run server)

1. . if you get below message in terminal then your server will running successfully.



route and its functionality :

For this use any API using tools like Postman or Thunder Client.

i use THUNDER CLIENT.

CREATE ROUTE :

1. . This route is used to create a new employee in database with a

below fields.

username, empname, email, role, salary

1. . in thunder client click on new request and select this options

method as post url as http://localhost:5000/api/user/create pass this json data as a body as your required value.

{

"username": "jack",

"empname": "jack rider",

"email": "jack@gmail.com",

"role": "Front End Developer",

"salary": 60000

}

1. . finally press send to insert data in mongodb data base and get a

inserted data as a response.

1. . If user is already in db it will return User is already exist as response. for more details visit below output images...

READONE :

1. . This route is used to read specific user info by passing that user id

as a param. method as get

url as http://localhost:5000/api/user/read/65ed7b3d76e1dcc9a51654ca

1. . After sending you will get that specific user details as response.

READALL :

1. . Read all route is used to get all the user data existing in the mongodb

data base .

method as get url as http://localhost:5000/api/user/readall

1. . After sending you will get that all user details as response.

UPDATE :

1. . This route is used to update specific user by passing that user id as

a param. method as put

url as http://localhost:5000/api/user/update/65ed7b3d76e1dcc9a51654ca

1. . After sending you will get updated user details as response.

DELETE :

1. . This route is used to delete specific user by passing that user id as

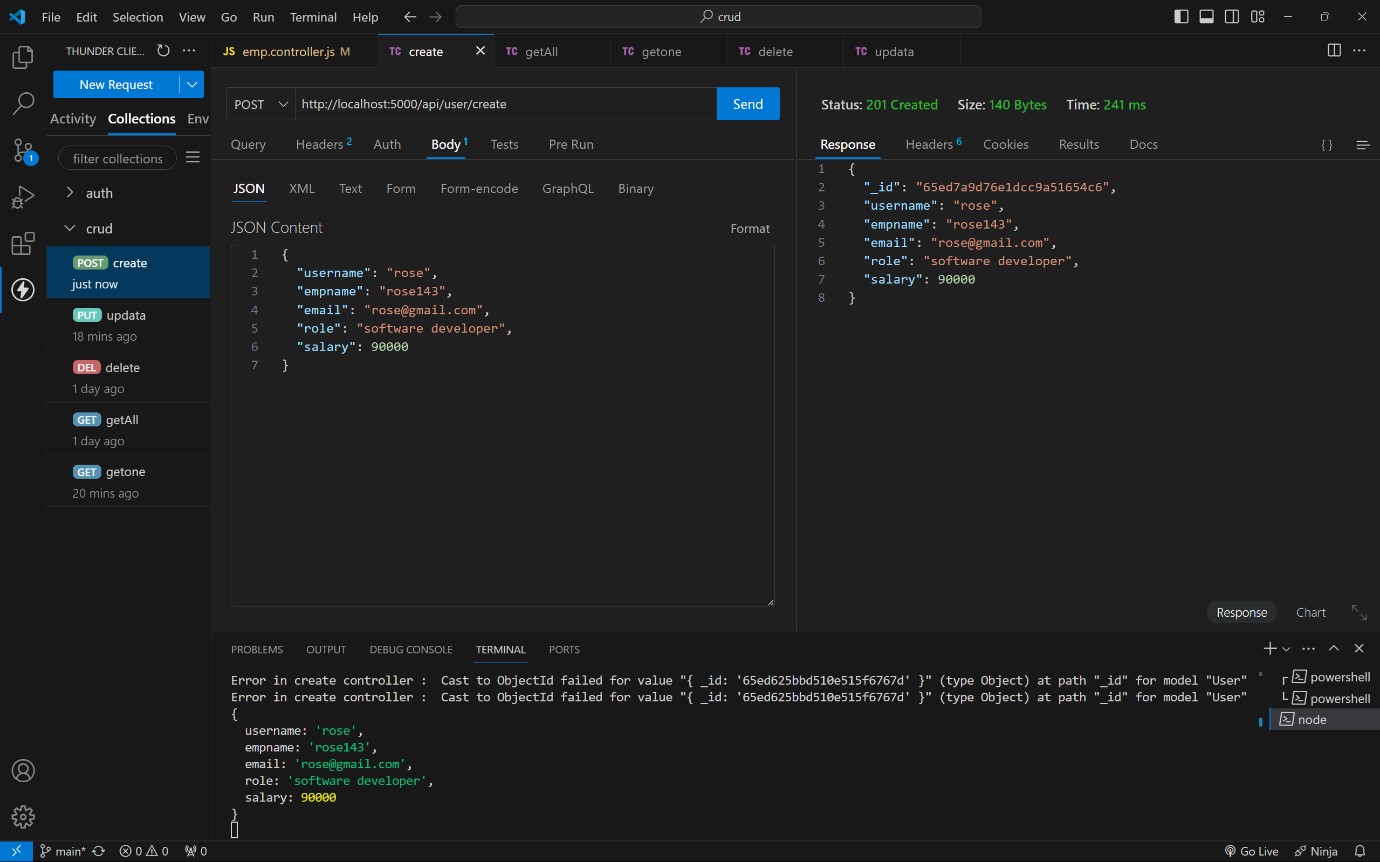
a param. method as delete

url as http://localhost:5000/api/user/delete/65ed7b3d76e1dcc9a51654ca

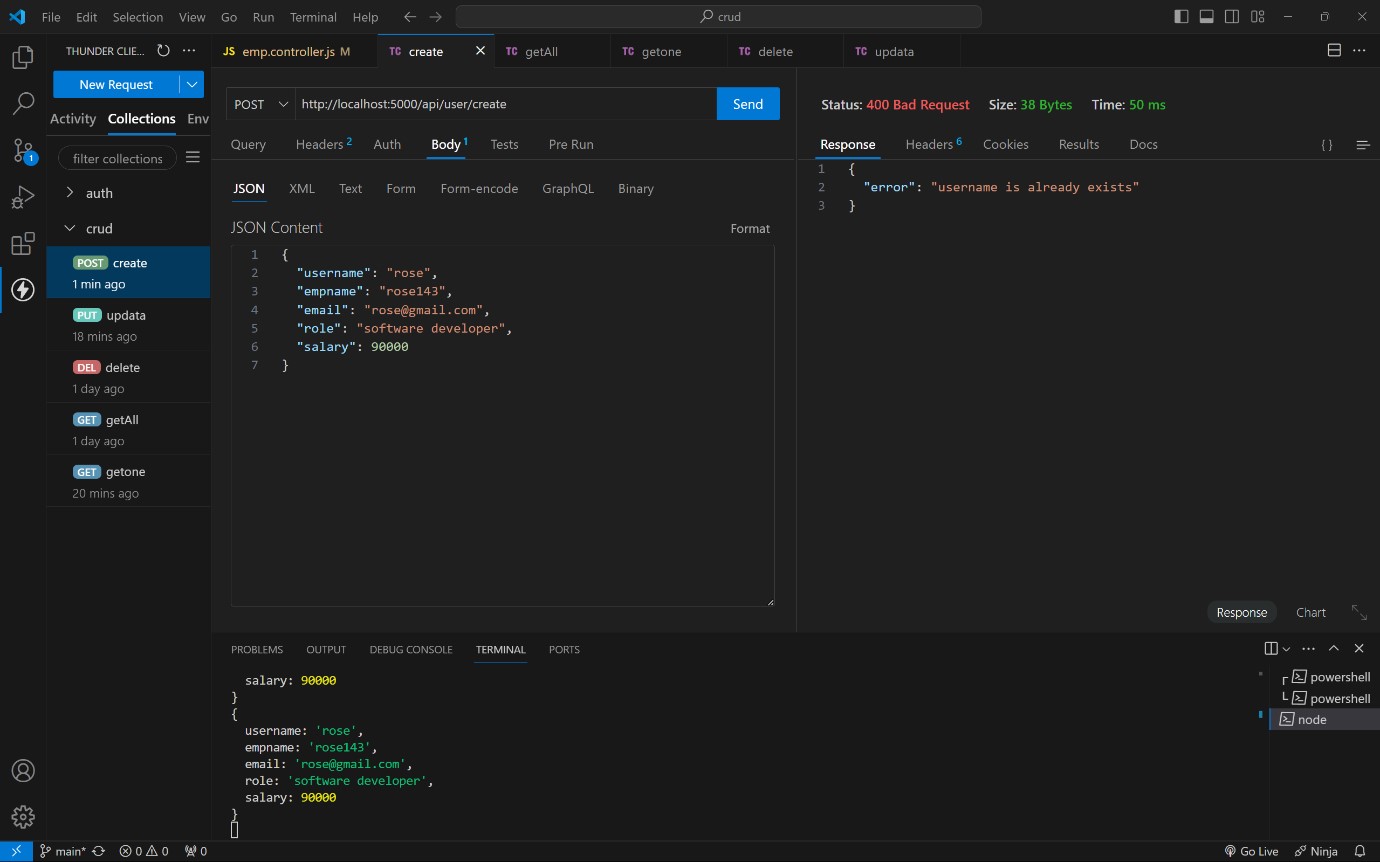
1. . After sending you will deleted successfully as response.

OUTPUT :

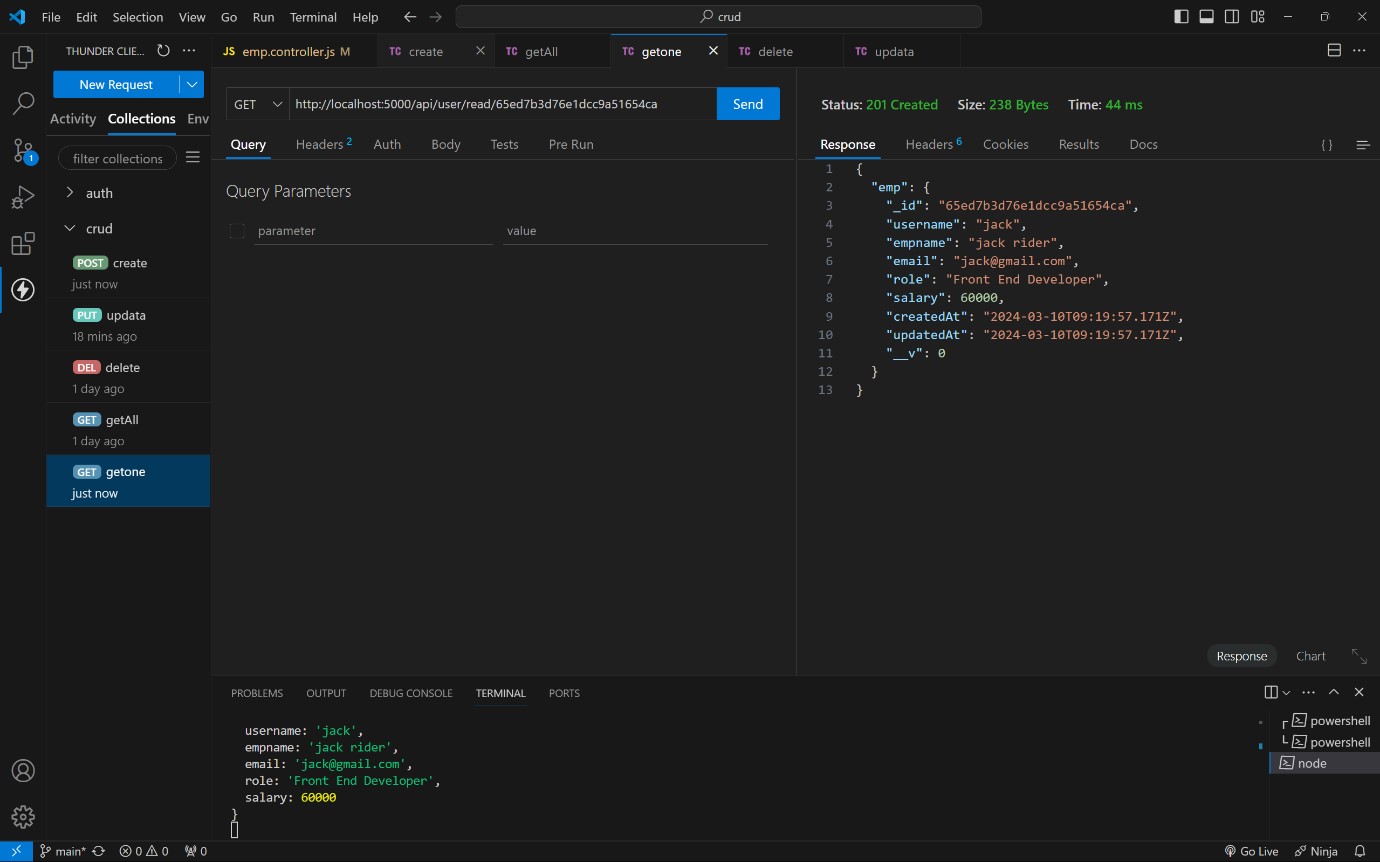
CREATE A NEW USER :



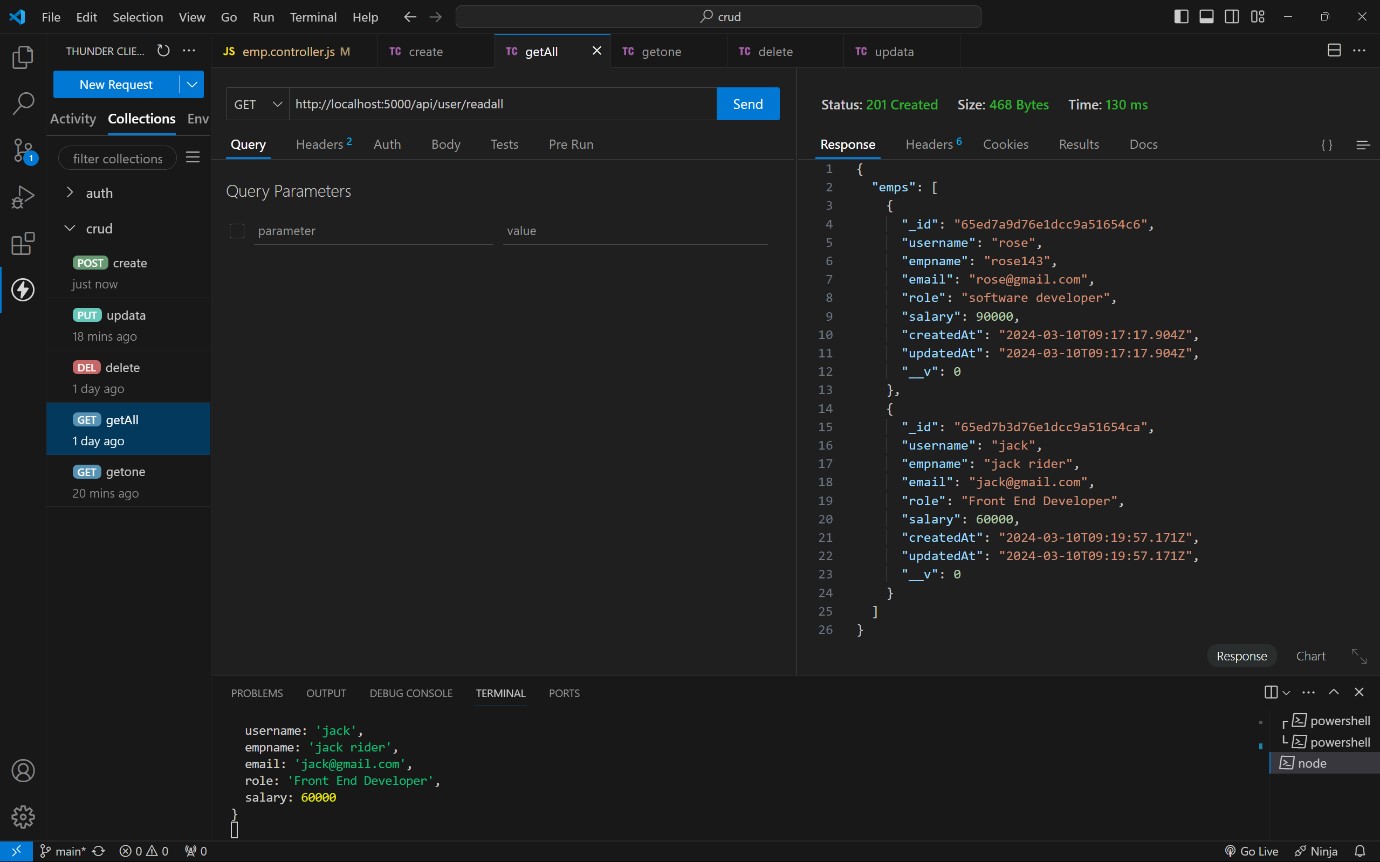
CREATING USER WITH EXISTING USERNAEM :



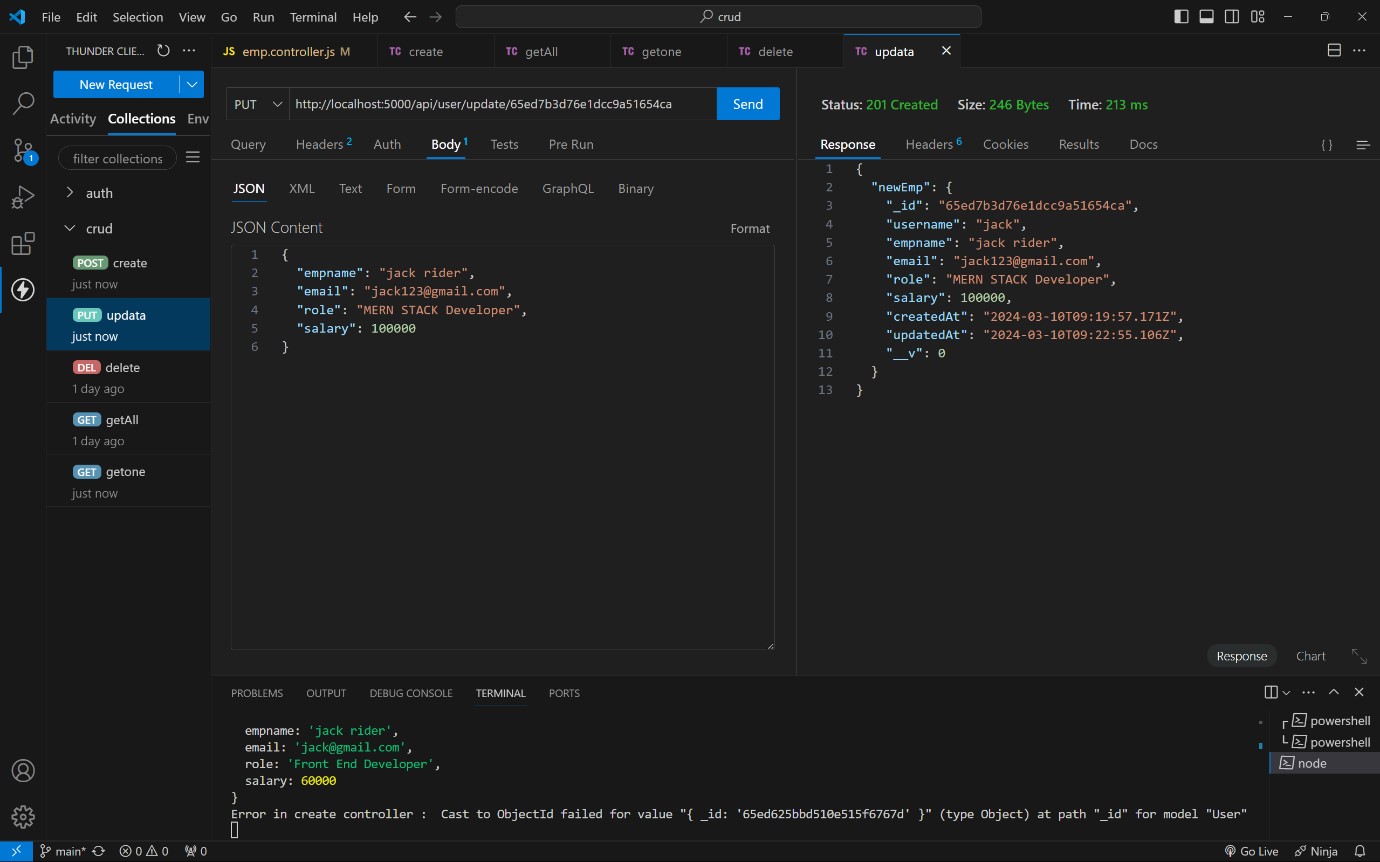
READONE :



READ ALL :



UPDATE :



DELETE :

