

CREATING A CRUD APP ON MERN STACK

Sabse pehle make sure you downloaded mongodb and mongodb atlas from the official website and if errors occurs then check the environment variibales and youtube karlo error ko

- 1) Create a folder name project
- 2) Create a folder name 'server' inside project
- 3) Open cmd in server and install 'npm init'
- 4) Then run this command - 'npm i express mongoose cors'
- 5) Now run this command - 'npm i nodemon'
- 6) Now create a file name index.js
- 7) Now go to package.json and save this inside scripts-

```
"start" : "node index.js",  
  "dev" : "nodemon index.js"
```

Ab jab bhi nodemon run karna hai toh 'npm run dev' likhenge

- 8) Now go to index.js and type this code

```
const express = require('express')  
const cors = require('cors')  
  
const app = express()  
app.use(cors())  
  
const PORT = process.env.PORT || 8080  
  
app.get('/', (req, res) =>{  
  res.json({message: 'Server is running'})  
})  
  
app.listen(PORT, ()=>{console.log('Server is running...')})
```

- 9) Now to connect to mongodb we write these lines of code

```
const mongoose = require('mongoose')
```

Now open mongodb compass and copy the https code given there like this-
mongodb://localhost:27017

- 10) Now go to index.js and write this code

After const PORT waali line ke neeche likho ye line

```
mongoose.connect("mongodb://127.0.0.1:27017/databaseName")
```

- 11) Go to mongodb compass and connect option aayega usko kar lena

- 12) Mongoose.connect returns the promise so we can use .then and we can also catch our errors like this so write this code:-

```
mongoose.connect("mongodb://127.0.0.1:27017/databaseName")
  .then(() => { console.log('Connected to database...') })
  .catch((error) => { console.log(error) })
```

- 13) Pehle database se connect karenge fir server se toh we modify some lines of our code

App ko listen karo connect hone ke baad matlab on successful connection with database, ek promise return hoga usme app ko listen karo matlab server ko create karo.

```
mongoose.connect("mongodb://127.0.0.1:27017/databaseName")
  .then(() => {
    console.log('Connected to database...')
    app.listen(PORT, () => { console.log('Server is running...') })
  })
  .catch((error) => { console.log(error) })
```

- 14) Now we will create our schema

```
//schema
const schemaData = mongoose.Schema({
  name: String,
  email: String,
  mobile: Number
}, {
  timestamps: true
})
```

- 15) Now we will create our model

```
//model
const userModel = mongoose.model("user", schemaData)
```

- 16) Now go to mongodb compass and refresh your database name and you will find your collection name as users

- 17) To read our data we use **get method** dekho code ko kaise change kar rhe hai

```
app.get('/', (req, res) => {
  res.json({ message: 'Server is running' })
})
```

Is code ko async await banao jaise neeche banaya hai

```
//read
app.get('/', async (req, res) =>{

    const data = await userModel.find({})

    res.json({success: true, data: data})
})
```

- 18) Go to rapidapi client inside vs code and test the localhost:8080 and get your response, response ye milna chahiye

```
{
  "success": true,
  "data": []
}
```

- 19) Now create a post request like:-

```
app.post("/create", (req, res)=>{
    console.log(req.body)
    res.json({success : true, message : 'message saved successfully'})
})
```

Now go to rapidapi client and test this post request by writing localhost:8080/create
You will get output like

```
{
  "success": true,
  "message": "message sent successfully"
}
```

- 20) To print your data that you are entering in post method use this line of code to convert it into json format

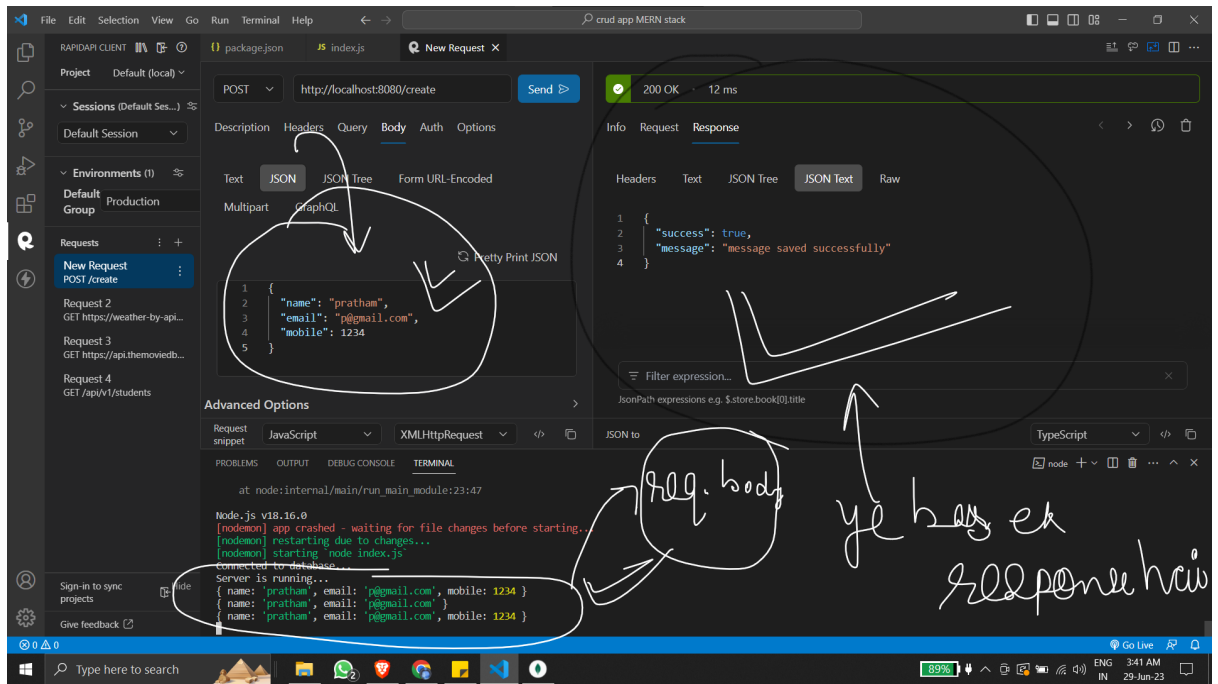
```
//accept data in json format
app.use(express.json())
```

- 21) Now go to client and test post method you will get point 19 output and now you will get

Connected to database...

Server is running...

{ name: 'pratham', email: 'p@gmail.com', mobile: 1234 } ye line milegi from the client toh yaha se hamara post method successful hua

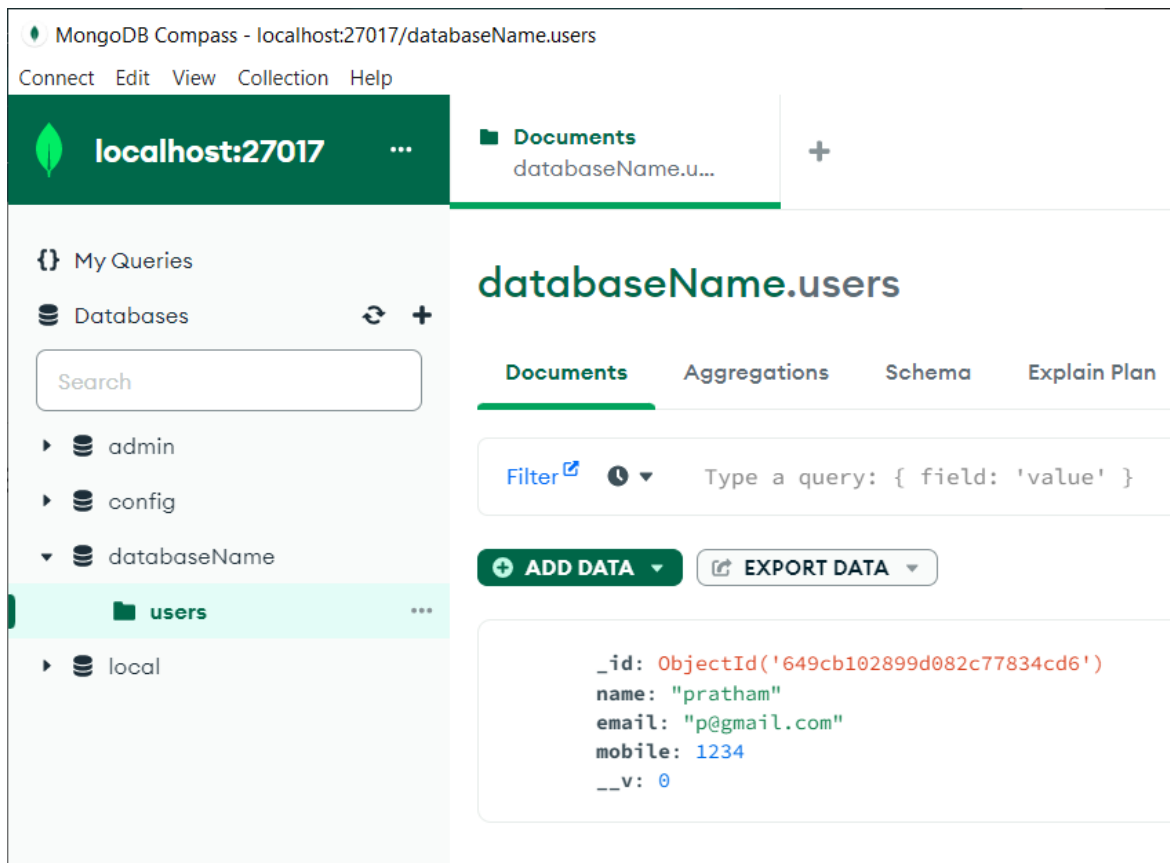


console(req.body) kiya hai apan ne upar code mai dekho

- 22) Now add req.body jo ki ek req post karte time kari hogi kuch details dekar toh details apan abhi kaha de rhe hai? Rapidapiclient par!!!!
- 23) Now add req.body in your mongodb database
post wale code ko thode bhot changes kiye hai async banaya hai and data ko store karwaya hai userModel call karke and data ko await karke save karwaya hai database mai

```
//create data || save data in mongodb
app.post("/create", async (req, res)=>{
  console.log(req.body)
  const data = new userModel(req.body)
  await data.save()
  res.json({success : true, message : 'message saved successfully'})
})
```

- 24) We will get this rapidapiclient mai details daalne ke baad-



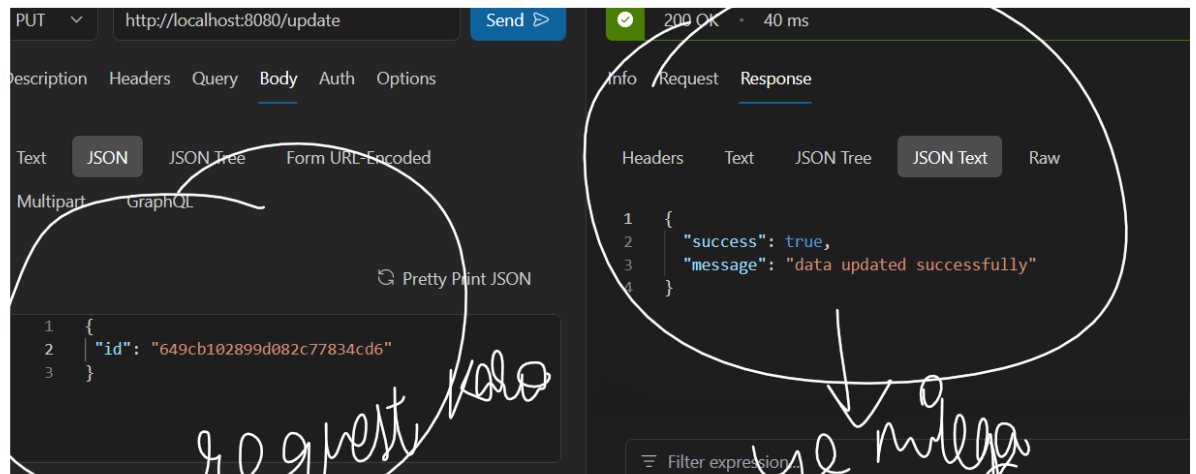
25) To update we use put method in nodeJs

```

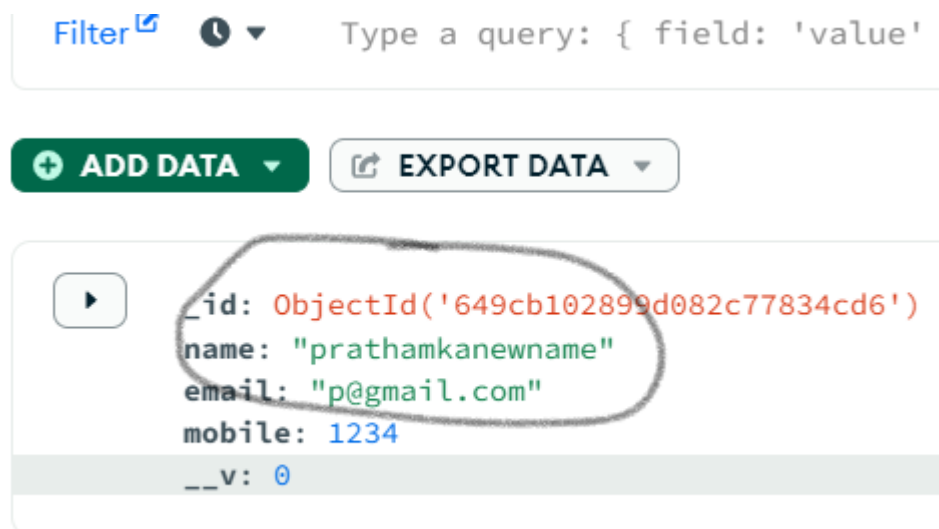
//for update we use the "put method in node"
app.put('/update', async (req, res)=>{
  console.log(req.body)
  await userModel.updateOne({ _id : req.body.id}, {name:
'prathamkanewname'})
  res.send({success: true, message:'data updated
successfully'})
})

```

Rapidapiclient mai jaao ye code likho aur ye response milega fir database check karo name updated milega



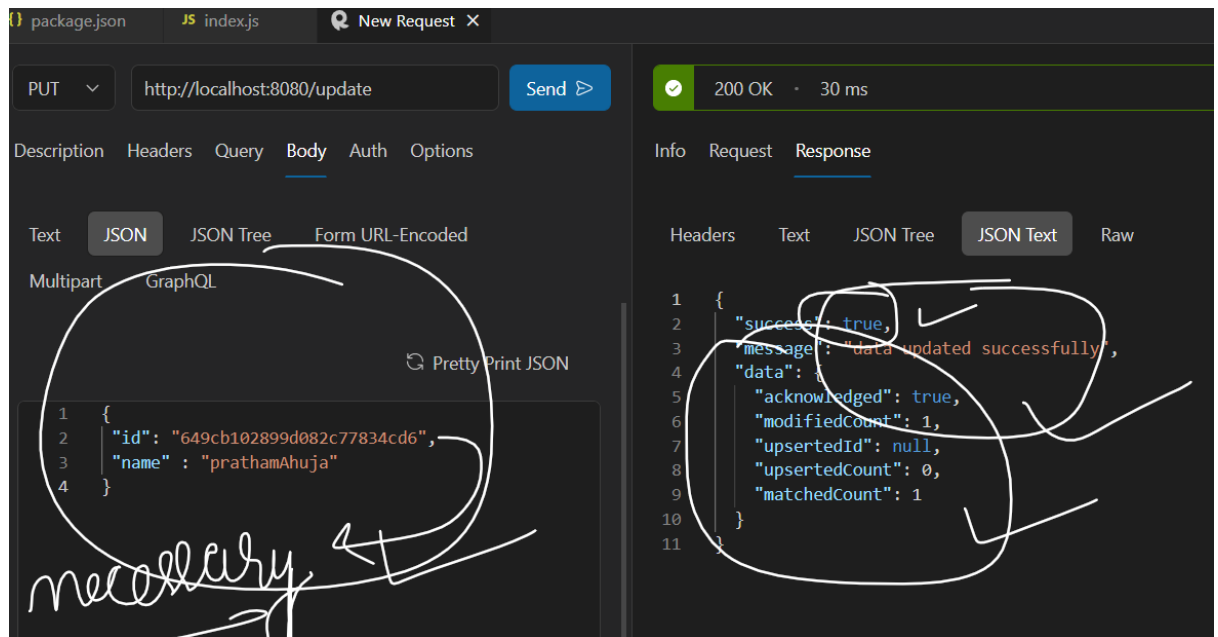
Name has been changed-



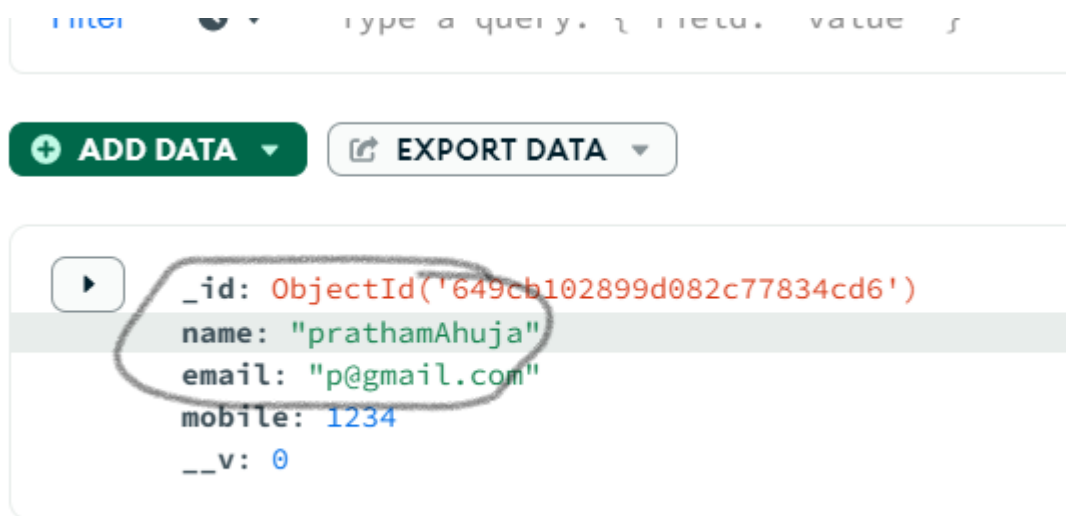
26) Update karo update wale code ko

```
//for update we use the "put method in node"
app.put('/update', async (req, res)=>{
  console.log(req.body)
  const{id,...rest} = req.body
  console.log(rest)

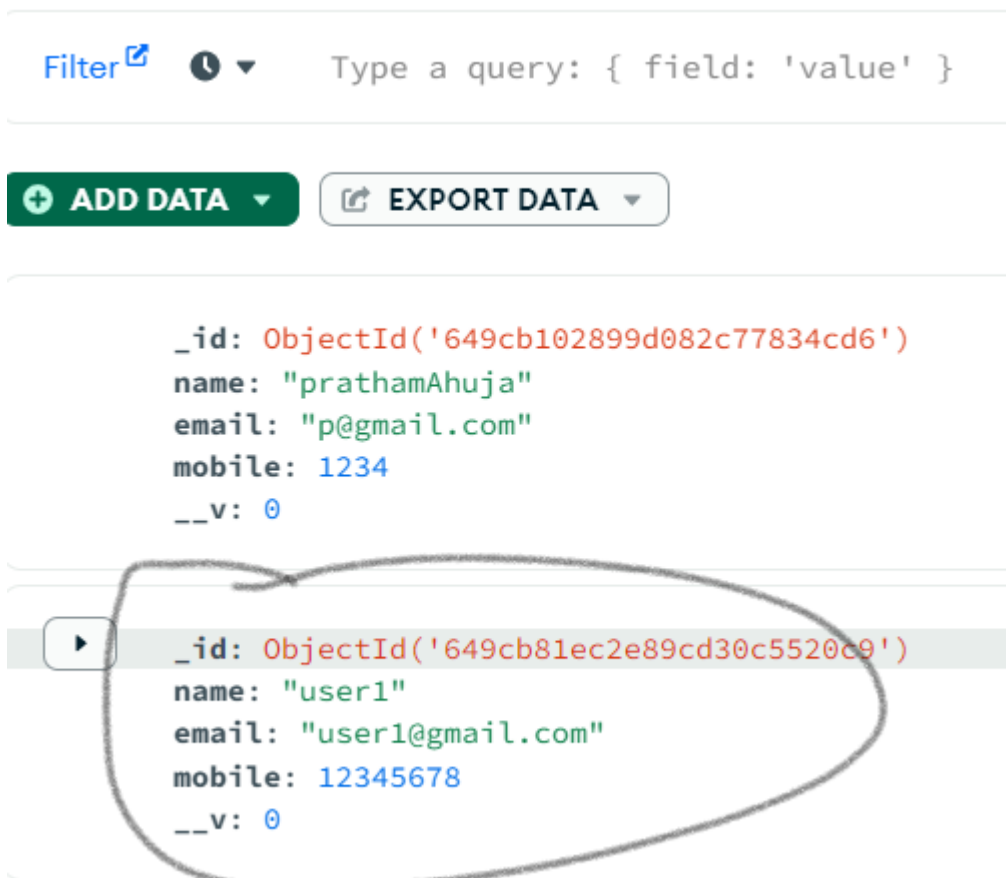
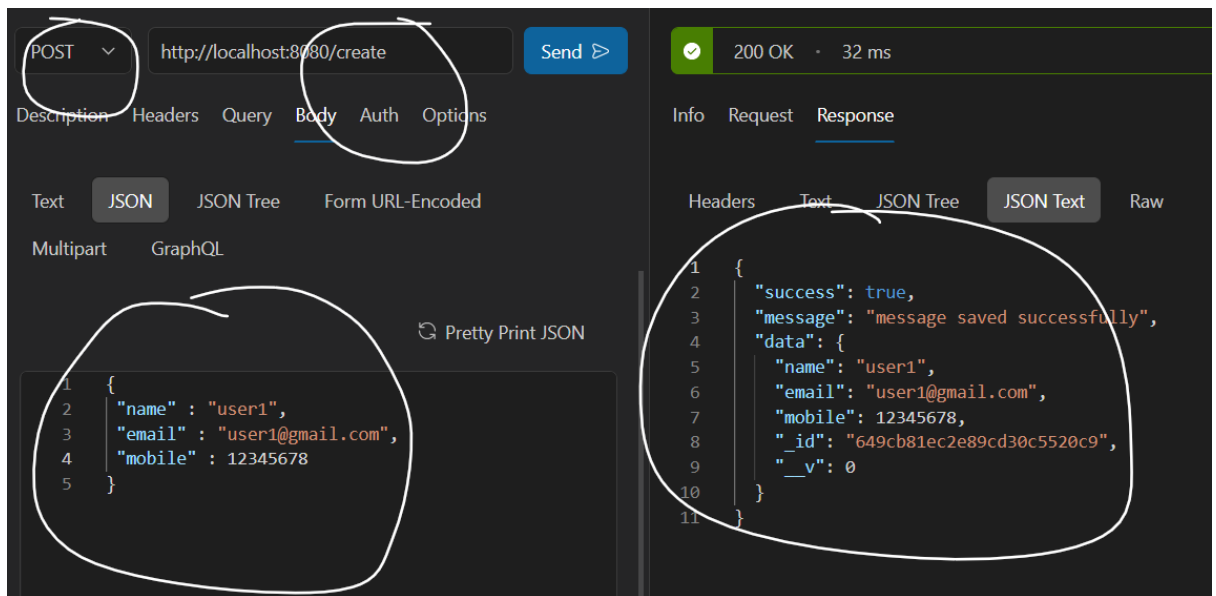
  const data = await userModel.updateOne({ _id :
req.body.id},rest)
  res.send({success: true, message:'data updated successfully',
data: data})
})
```



Username updated from prathamkanewname to prathamAhuja



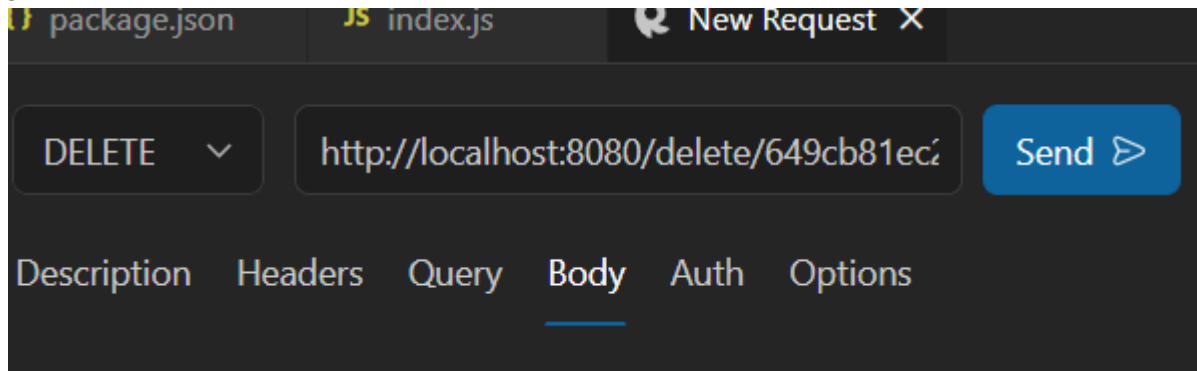
Ab agar koi bhi user ko new add karna website par to /create par jao client mai aur post method click karke details enter karo and database mai vo save ho jayegi like this-



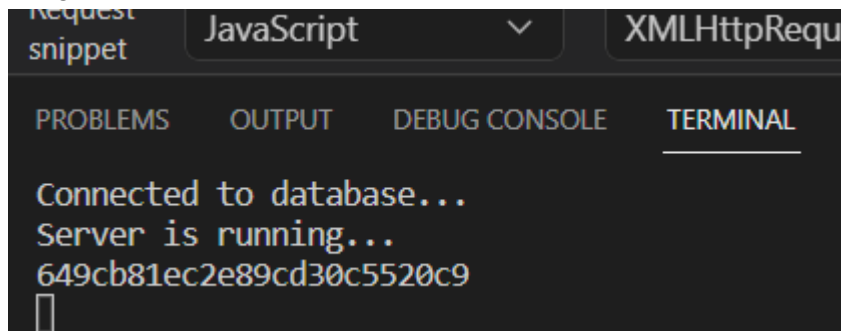
27) To delete-

```
//delete a user data fully
app.delete('/delete/:id_ek_variable_hai', async (req, res)=>{
  const id = req.params.id_ek_variable_hai;
  console.log(id)
})
```


To delete we use route params and we use id as our key and save id in id variable and go to rapidapi client and do this

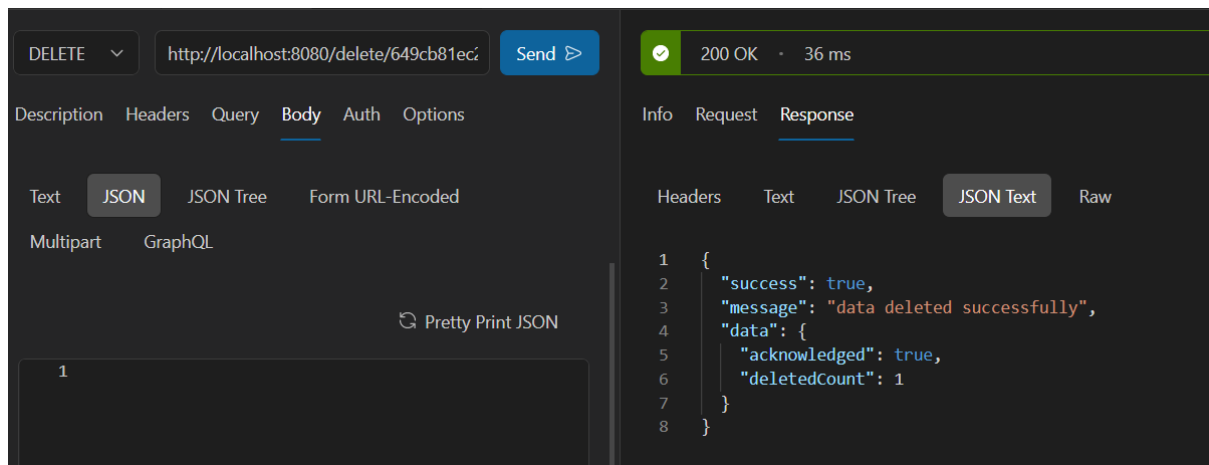


We get id on our console



That means delete is working now lets delete our data from the database

```
//delete a user data fully
app.delete('/delete/:id_ek_variable_hai', async (req, res)=>{
  const id = req.params.id_ek_variable_hai;
  console.log(id)
  const data = await userModel.deleteOne({_id: id})
  res.send({success: true, message:'data deleted successfully', data:
data})
})
```



[ADD DATA](#) [EXPORT DATA](#)

```
_id: ObjectId('649cb102899d082c77834cd6')
name: "prathamAhuja"
email: "p@gmail.com"
mobile: 1234
__v: 0
```

user
deleted

Now to make front-end open another terminal and type
Npm create-react-app client
This will create node modules inside the client folder
Now write cd client
Now write npm start
So react will open at localhost:3000
Now write npm install react-icons --save

28) NOW WE ARE CONNECTING FRONTEND TO BACKEND

Now write npm i axios

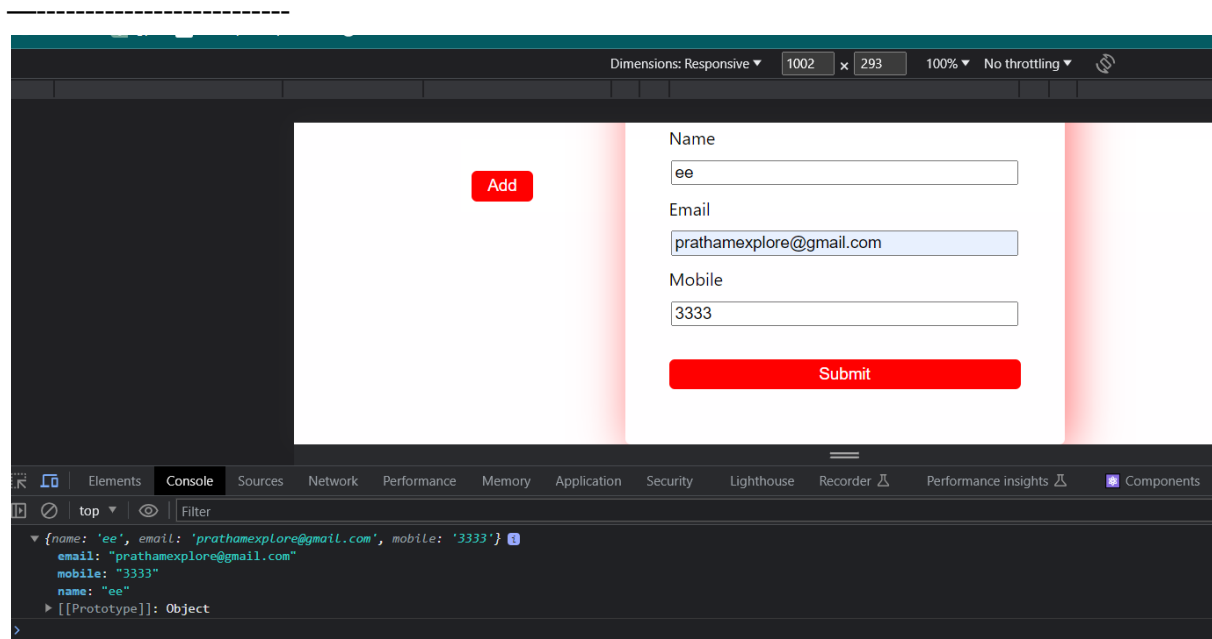
29) Now import after making react page i understood concept like useState using
objects GPT SE SIKHA hai bhot ache se, computed variables in JS,

onchange events and onclick events, event.target waghera sab GPT se seekha hai

Import axios from "axios"

axios.defaults.baseURL = "<http://localhost:8080/>" -> ye backend jaha hosted hai uski link hai

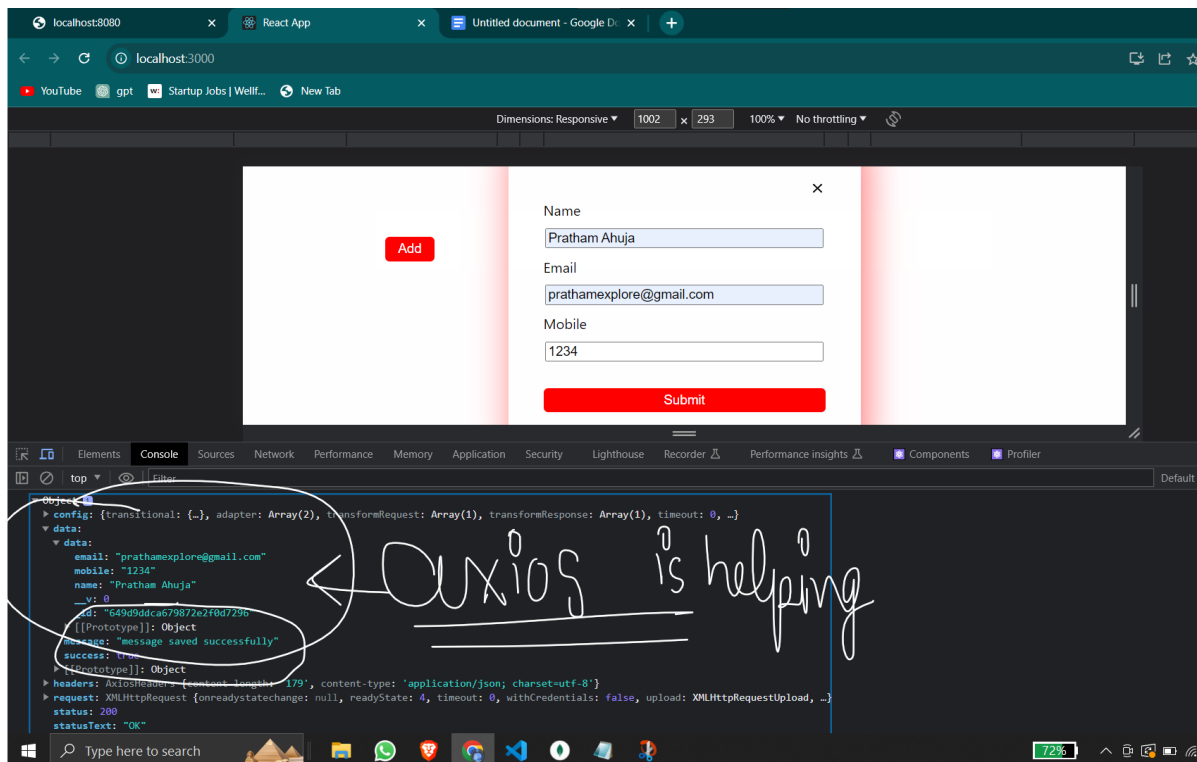
30) Now go to handleSubmit function and make it async and write this line ->
const data = await axios.post("/create", formData)



Using onchange in form and using onclick in submit button

We use axios for fetching the data from the server and from the mongodb

```
const handleSubmit = async (eventt) => {  
  eventt.preventDefault();  
  const data = await axios.post('/create', formData)  
  console.log(data)  
}
```



31) We get data in our mongodb database in the type of formData

By writing below code in handlesubmit function

Ye jo data.data kiya hai na vo axios bhej raha hai apan ko json form mai backend se laake vo console se dekh lo konsa data chahiye uske hisab se data ko access karna hai

Axios is a Javascript library used to make HTTP requests from node. js or XMLHttpRequests from the browser

```
const handleSubmit = async (eventt) => {
  eventt.preventDefault();
  const data = await axios.post('/create', formData)
  console.log(data)
  if(data.data.success){
    setAddSection(false)
    alert(data.data.message)
  }
}
```

32) Now to fetch the data we write or to get the data we write

```
const getFetchData = async() => {
  const data = await axios.get('/')
}
```

```
}
```

33) Now make useState of dataList to fetch

```
const [dataList, setDataList] = useState([])
```

Strict mode ko off karna index.js se vrna useeffect error dega

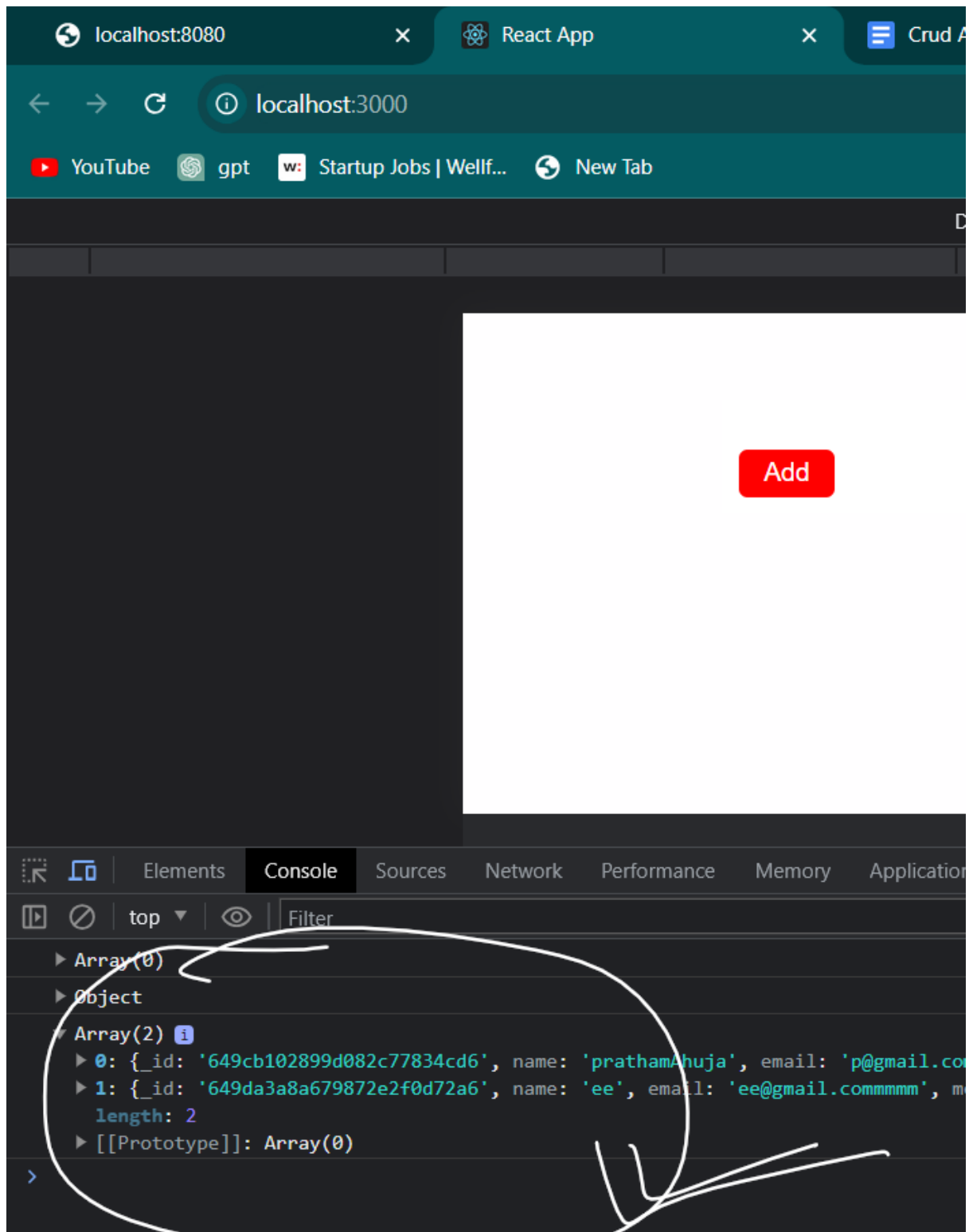
34) Saare data jo bhi database mai present hai usko console par print karwaya using the below code:-

```
const getFetchData = async() => {
  const data = await axios.get('/')
  console.log(data)
  if(data.data.success){
    setDataList(data.data.data) //data.data.data apan ko axios backend se
    laakar de raha hai aur ye list of all data hai saari information
    //of all people that have registered, now ye data (dataList) mai aa
    raha hai
  }
}

//pehli baar chalane k liye render ke just baad useEffect use karo and
empty array second argument mai pass karo
useEffect(() => {
  getFetchData()
}, [])

console.log(dataList)
```

We get all the data below in our console that was saved in the mongodb database



Apan ka sara data (dataList) naam ke variable mai hai
Ab is dataList ko iterate karo and print it

Using the map method apan ye data jo console mai hai usko apan apne table mai show karwa sakte hai

```

<tbody>
  { /* using the map method we display our data in the table */
  {
    dataList.map((element) => {
      return (
        <tr>
          <td>
            {element.name}
          </td>
          <td>
            {element.email}
          </td>
          <td>
            {element.mobile}
          </td>
        </tr>
      )
    })
  }
</tbody>

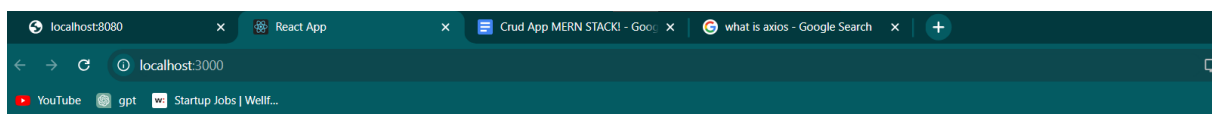
```

35) To delete the data:-

```

const handleDelete = async (id) => {
  const data = await axios.delete('/delete/'+id) //route params
  nodejs mai padha hai na vohi id pass kar rahe hai
  alert(data.data.message)
}

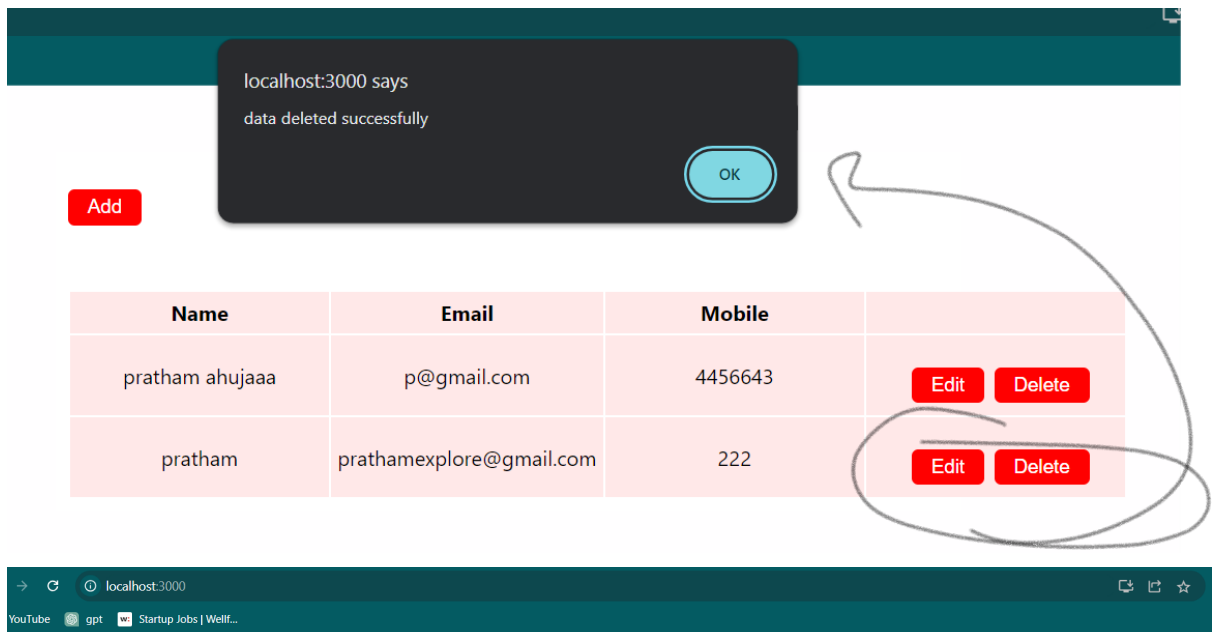
```



Currently
Before delete

Add

Name	Email	Mobile		
pratham ahujaaa	p@gmail.com	4456643	Edit	Delete
pratham	prathamexplore@gmail.com	222	Edit	Delete



Refresh to get new data

Name	Email	Mobile	
pratham ahujaaa	p@gmail.com	4456643	Edit Delete
pratham	prathamexplore@gmail.com	222	Edit Delete

Add

Name	Email	Mobile	
pratham ahujaaa	p@gmail.com	4456643	Edit Delete

data deleted but after refresh

Toh refresh apne aap ho delete hone ke baad to kya kare?

Neeche code likha hai useEffect mai jo function hai use call kardo taaki render hone ke baad apne aap call ho jaye


```
const handleDelete = async (id) => {  
  const data = await axios.delete('/delete/'+id) //route params nodejs  
  mai padha hai na vohi id pass kar rahe hai  
  alert(data.data.message)  
  if(data.data.success) {  
    getFetchData() //useEffect mai hai matlab render hone ke baad ye apne  
    aap chalega  
  }  
}
```

Now to show if no data available we do:-

dataList[0] ? (return table vala code) : (<p> No data available to display </p>)

Edit ke liye bhi alag se saare functions banaye hai dekhlo lakin edit ni chl rha pta ni kyu bhot time waste hua h debug krne mai but not working