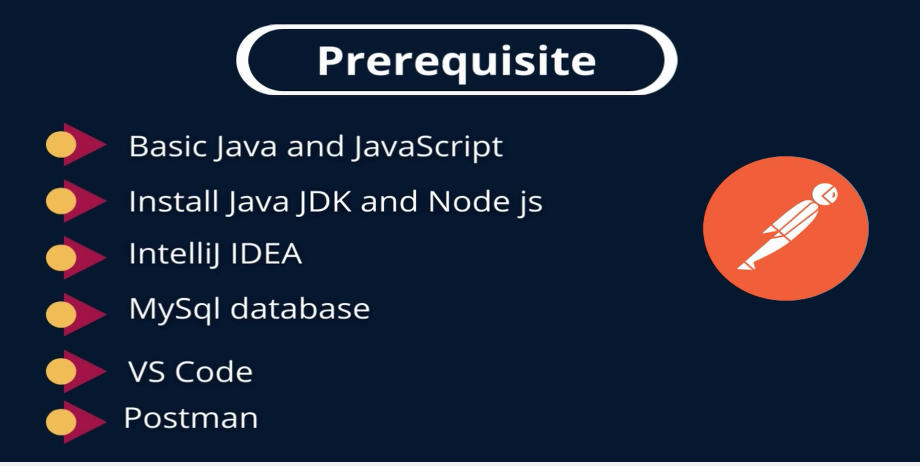
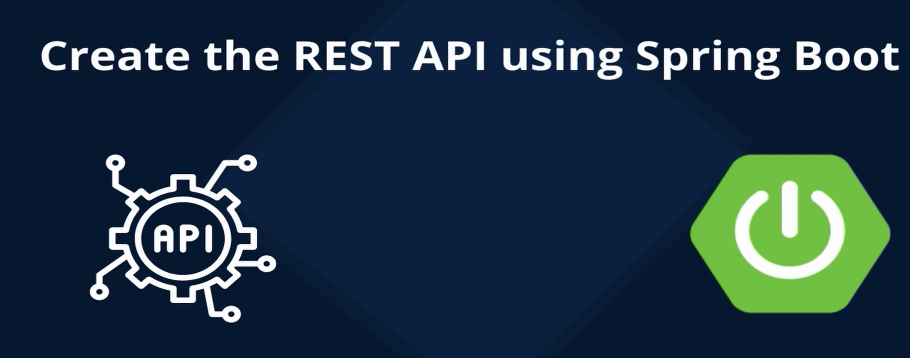
**FULL STACK JAVA CRUD USER APPLICATION**

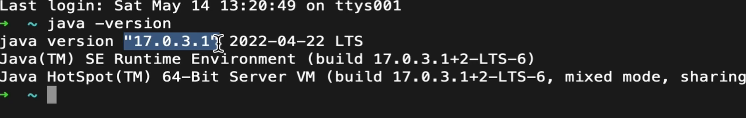
**Using springboot, Rest API , Mysql, Intellij Idea, Postman**





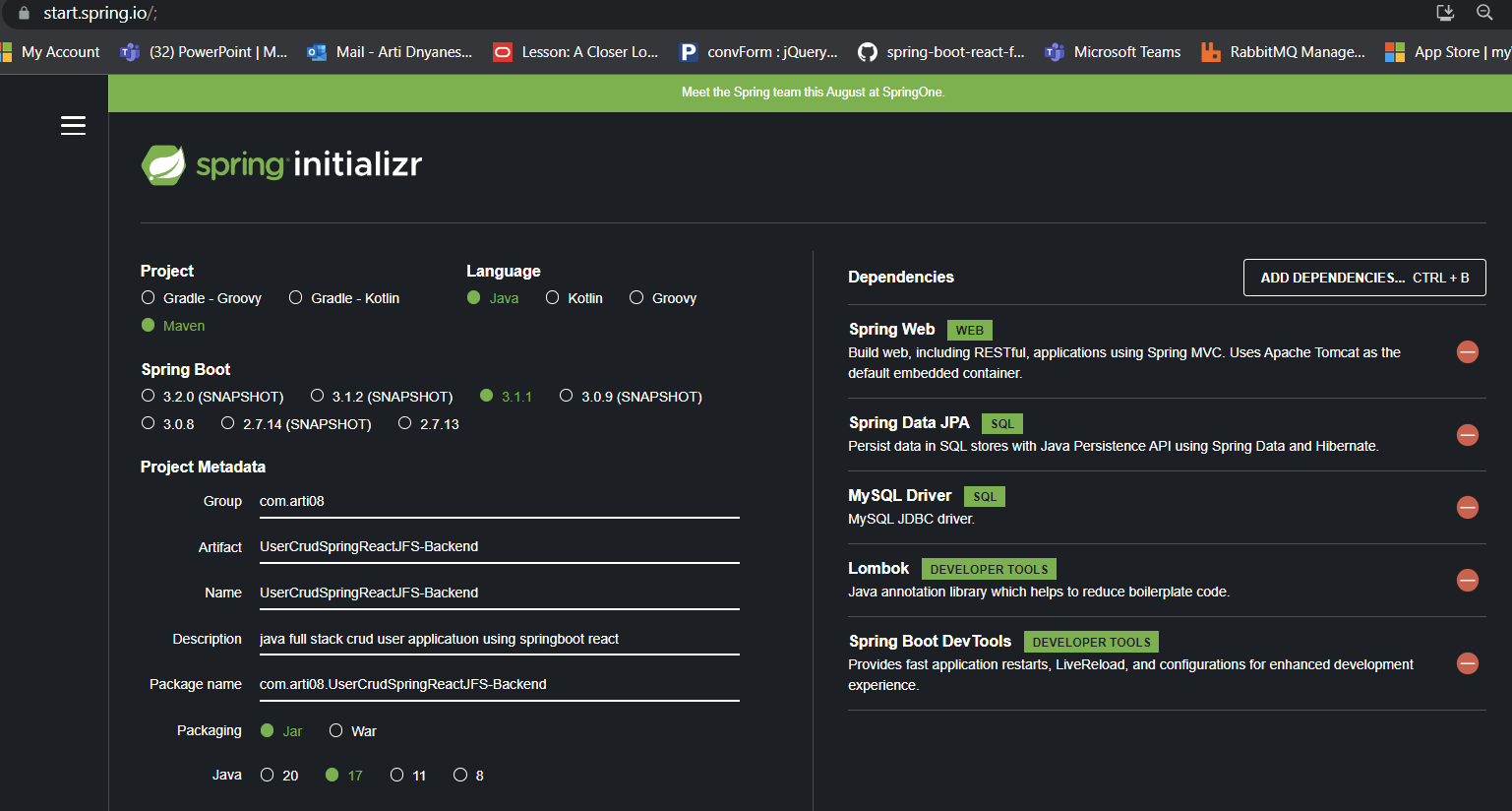
MAKE SURE 🡪 INSTALLED JAVA

Open cmd🡪 java –version



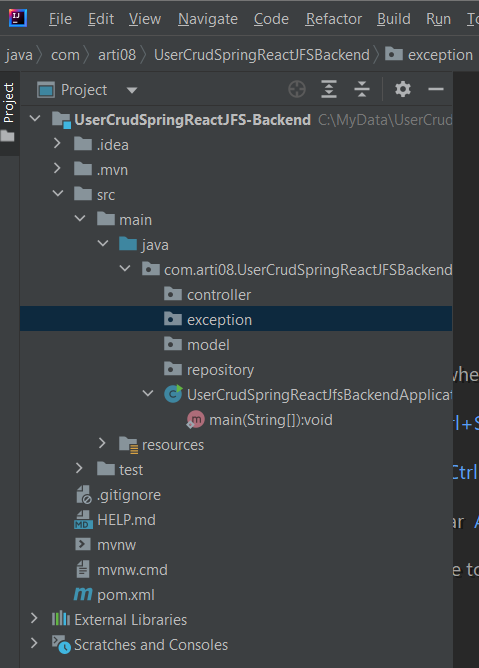
GO TO **start.spring.io**

* **Create springboot initializer application**



Generate > extract > Open intellij idea >open project folder one which is b4 src folder of the project

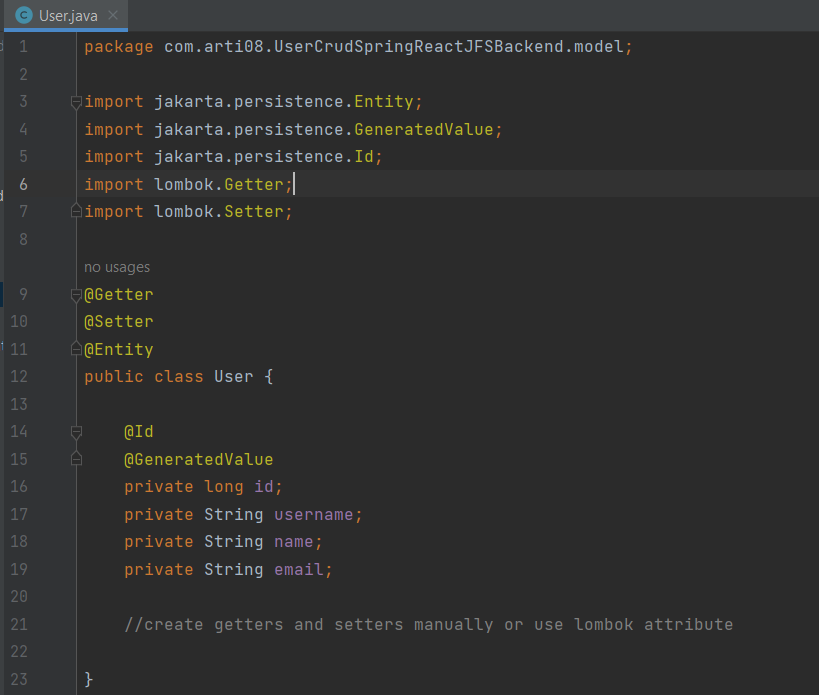
Create packages> model, controller, repository, exception





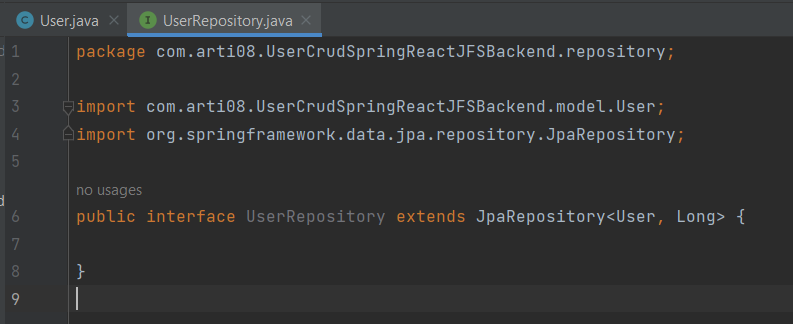
Create entity classes inside model

**1.user.java**



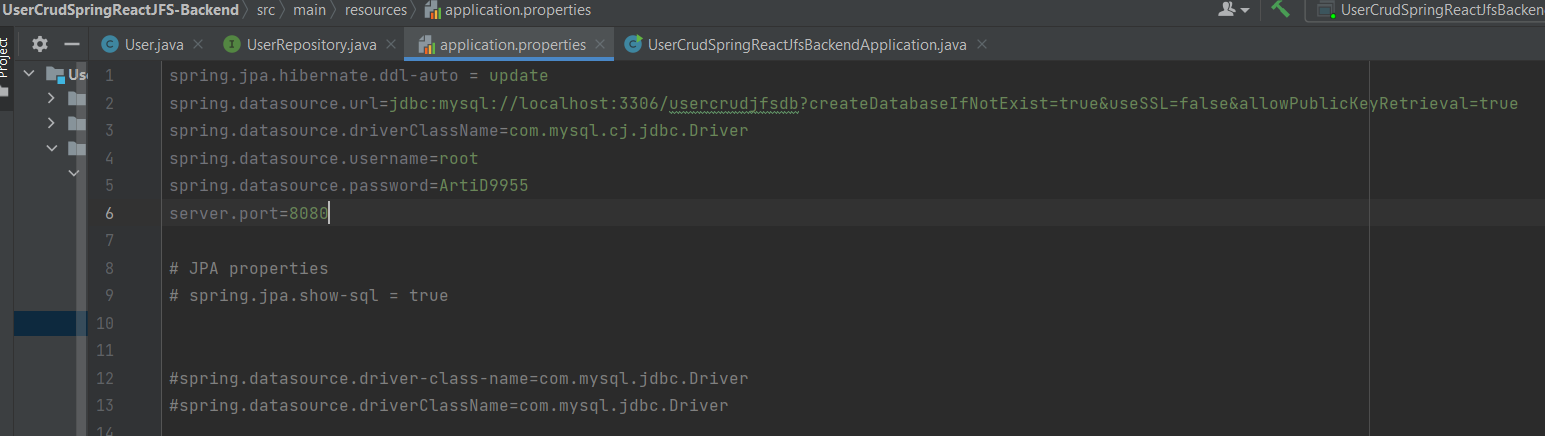
CREATE JAVA INTERFACE REPOSITORY

**2. UserRepository extending JPARepository**



**CONNECT SPRINGBOOT WITH MYSQL DB**

**3.APPLICATION PROPERTIES**

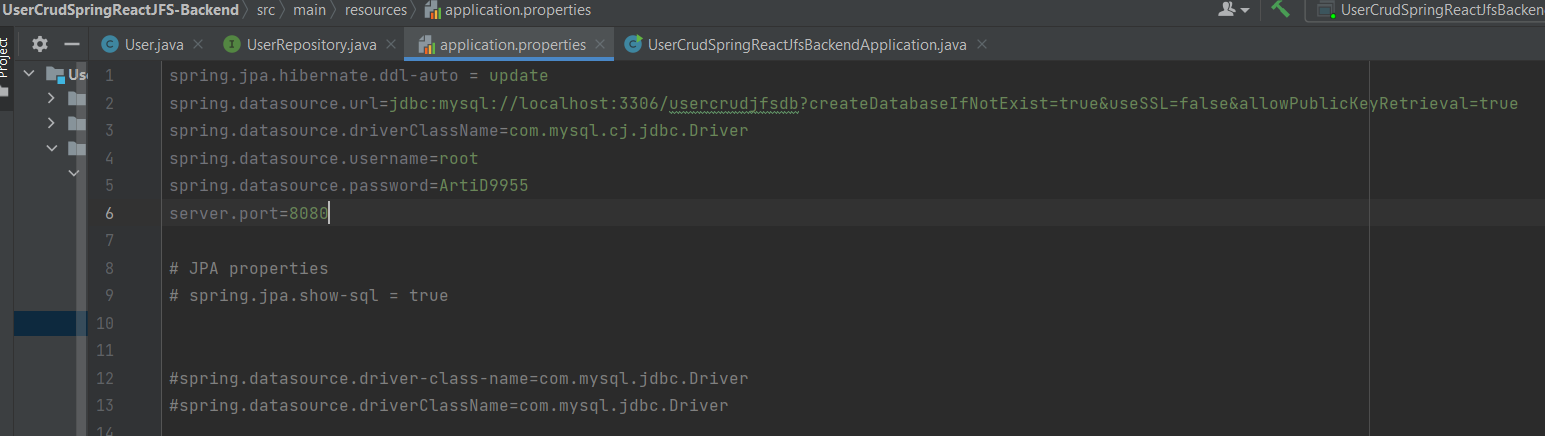


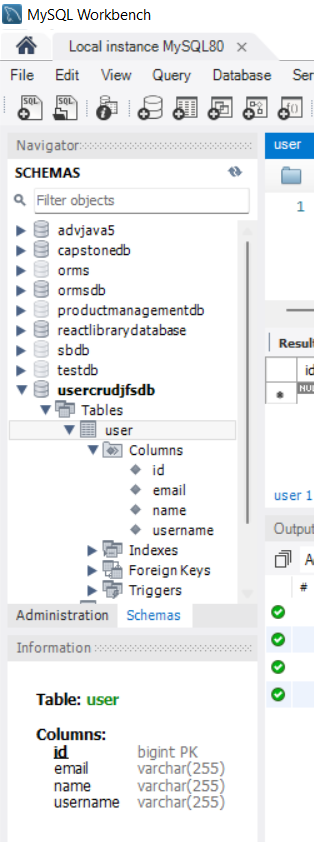
Run the application from runnable main class

If server started successfully

check if db created and tables inside it

Table created using @Entity



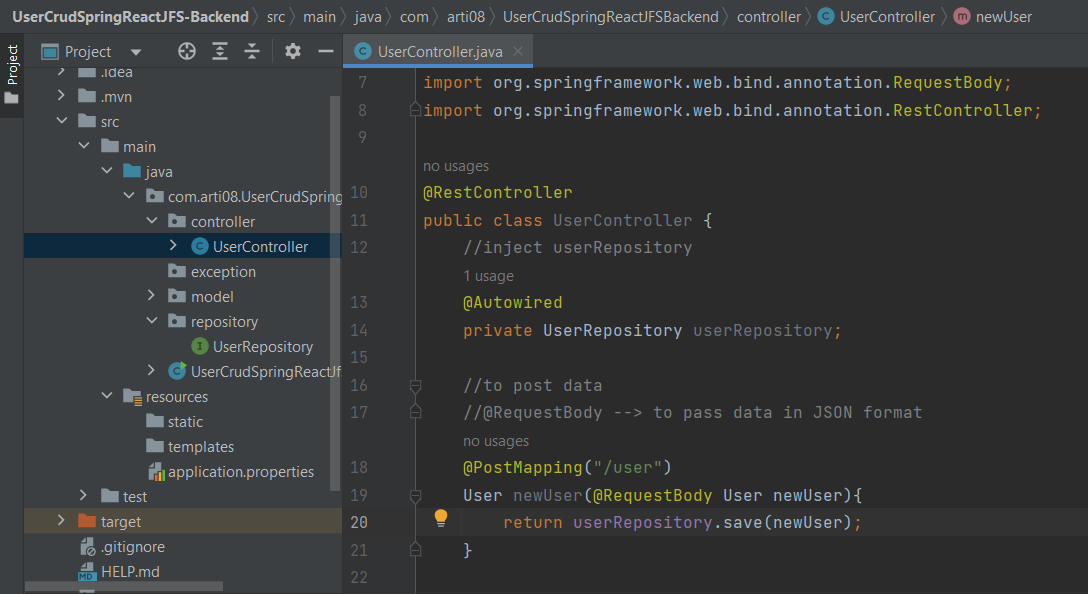




Create UserController inside controller pkg

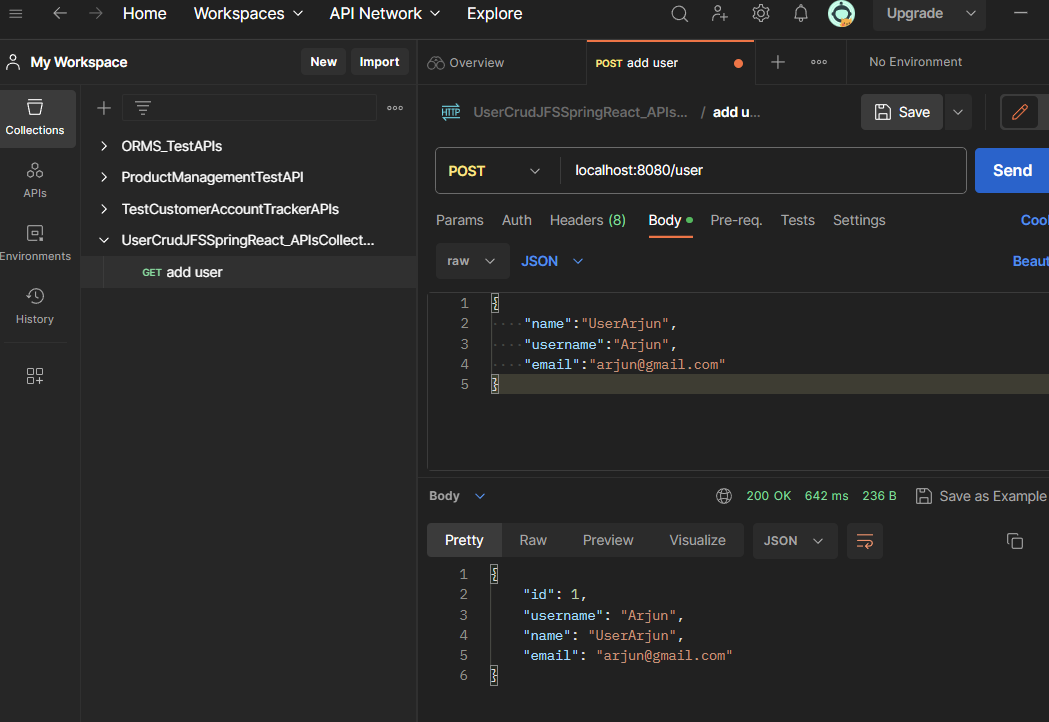
**4. UserController.java**

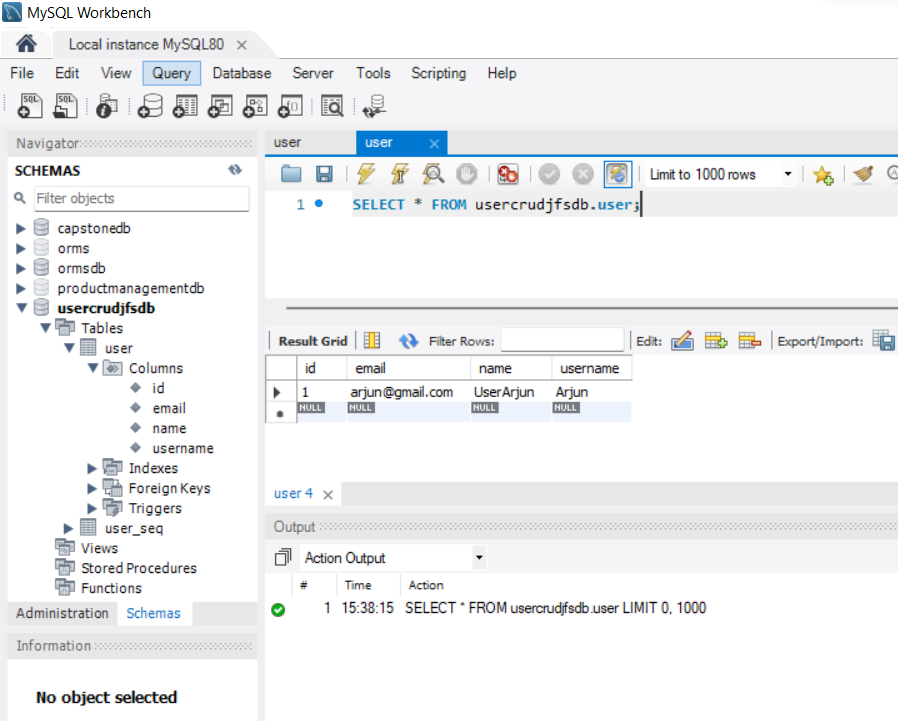
Annotate it with @RestController

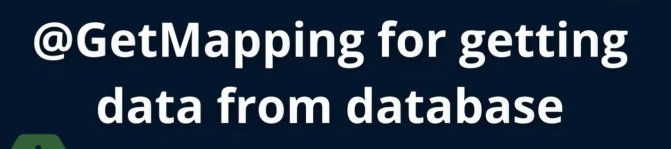


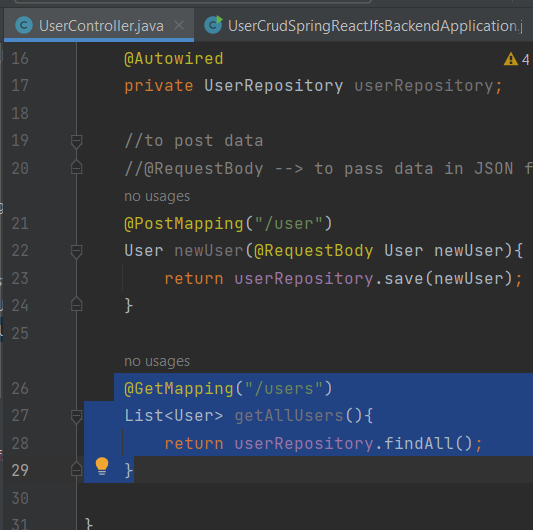
Run application

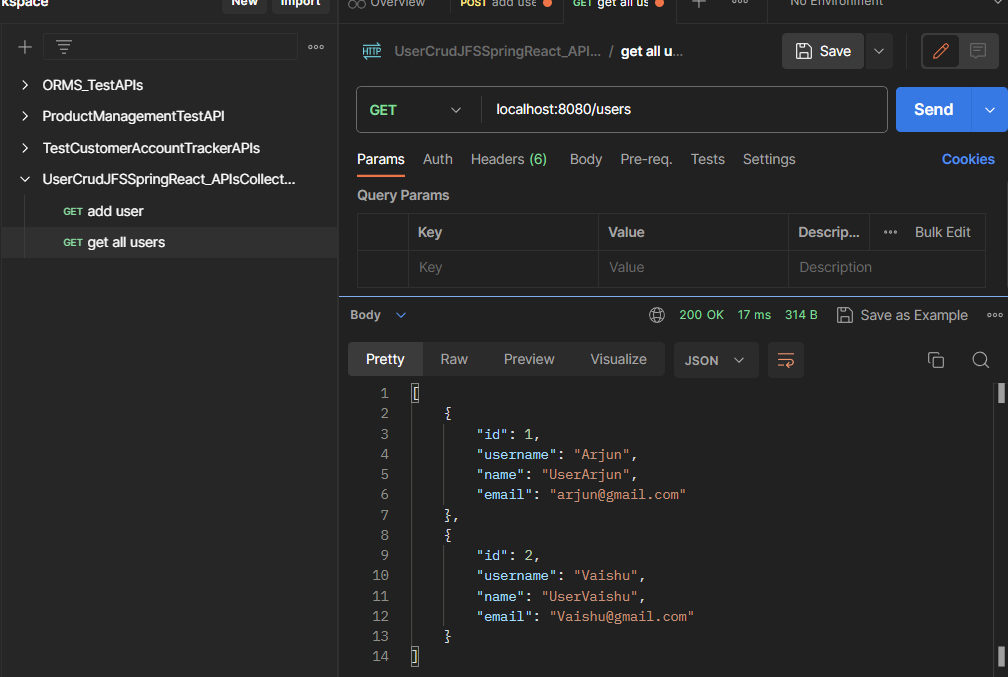
Open postman

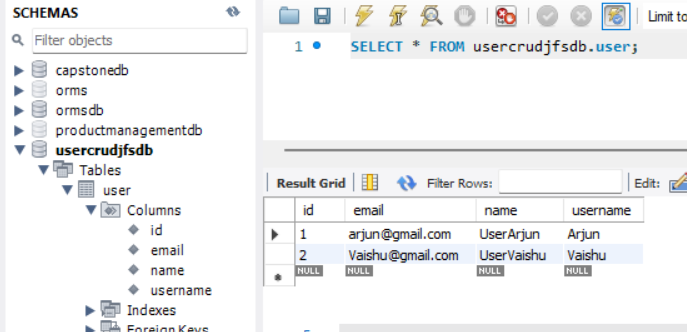


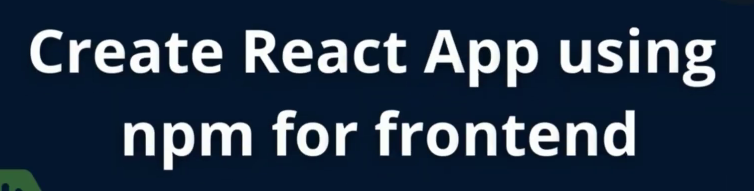










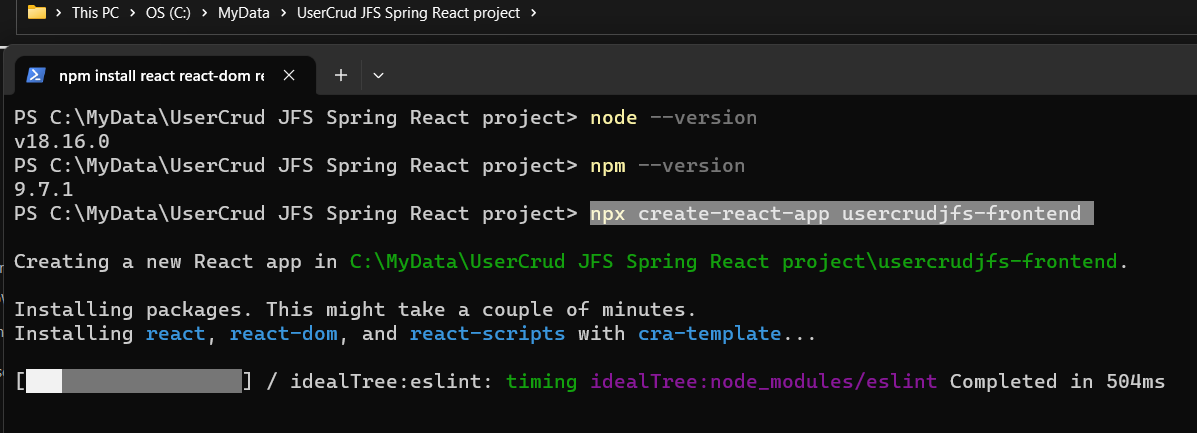


Check id node and npm is installed

TO CREATE REACT APP’

IN CMD> GO TO UR FOLDER PATH

Project folder or any folder> Npx create-react-app appName

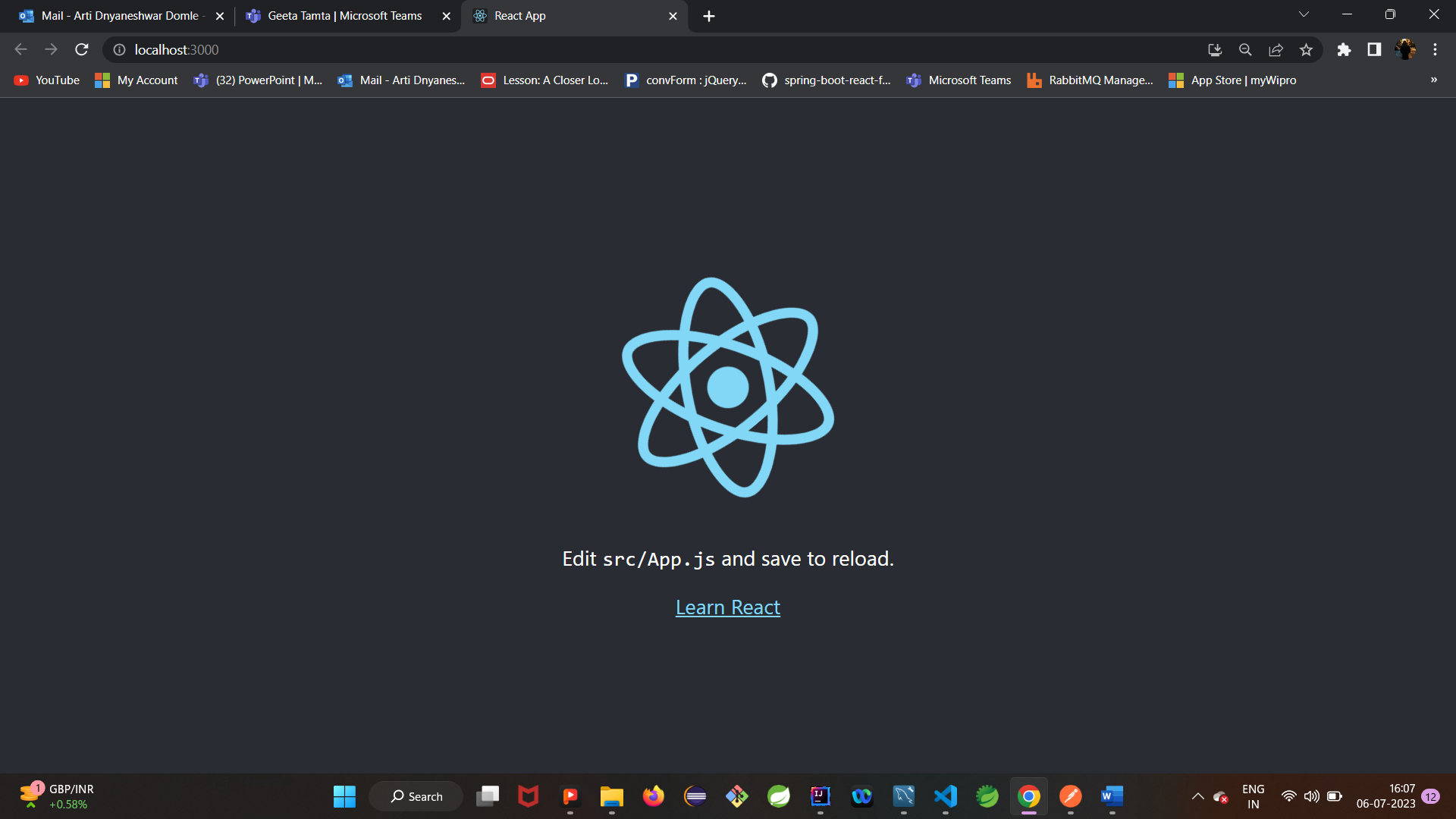


Open Project folder with VS code

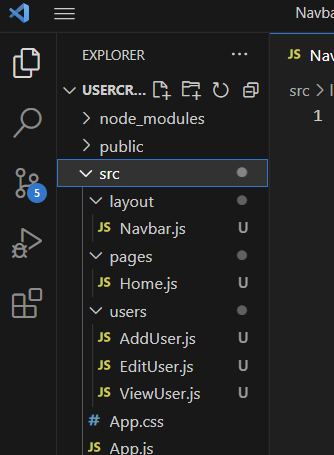
Go to Terminal tab > open new terminal

>npm start

Open crome> app display will be there



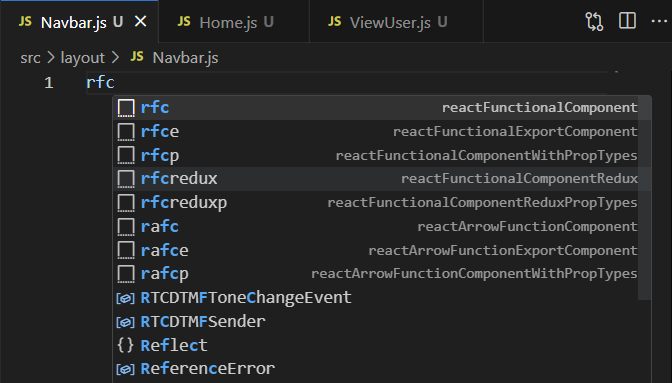
Create following file structure



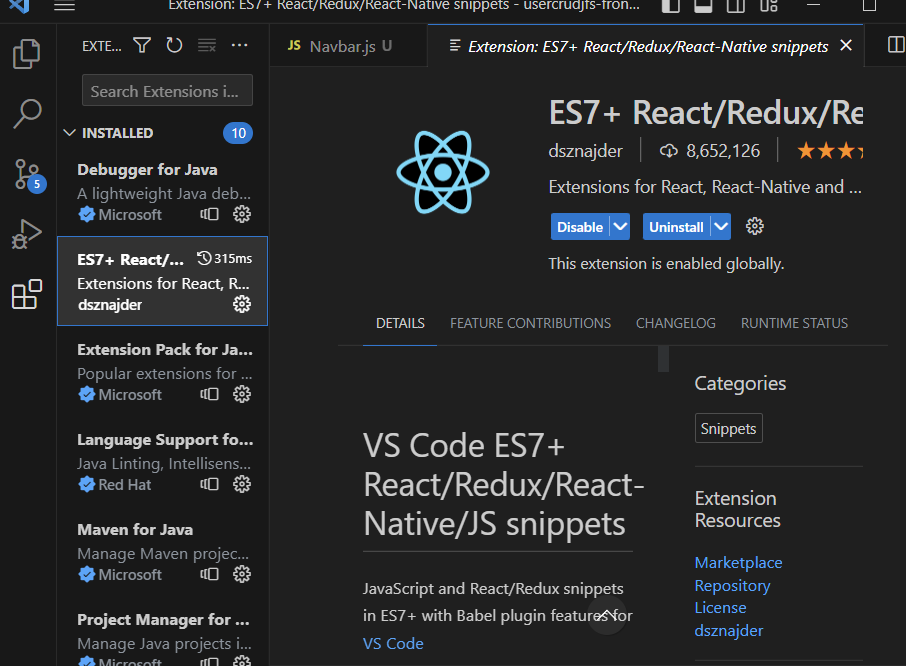


**ADD SNIPPLET EXTENTION 🡪 TO CREATE fuctional components**

Initial code **using rfc (react fuctional component)**

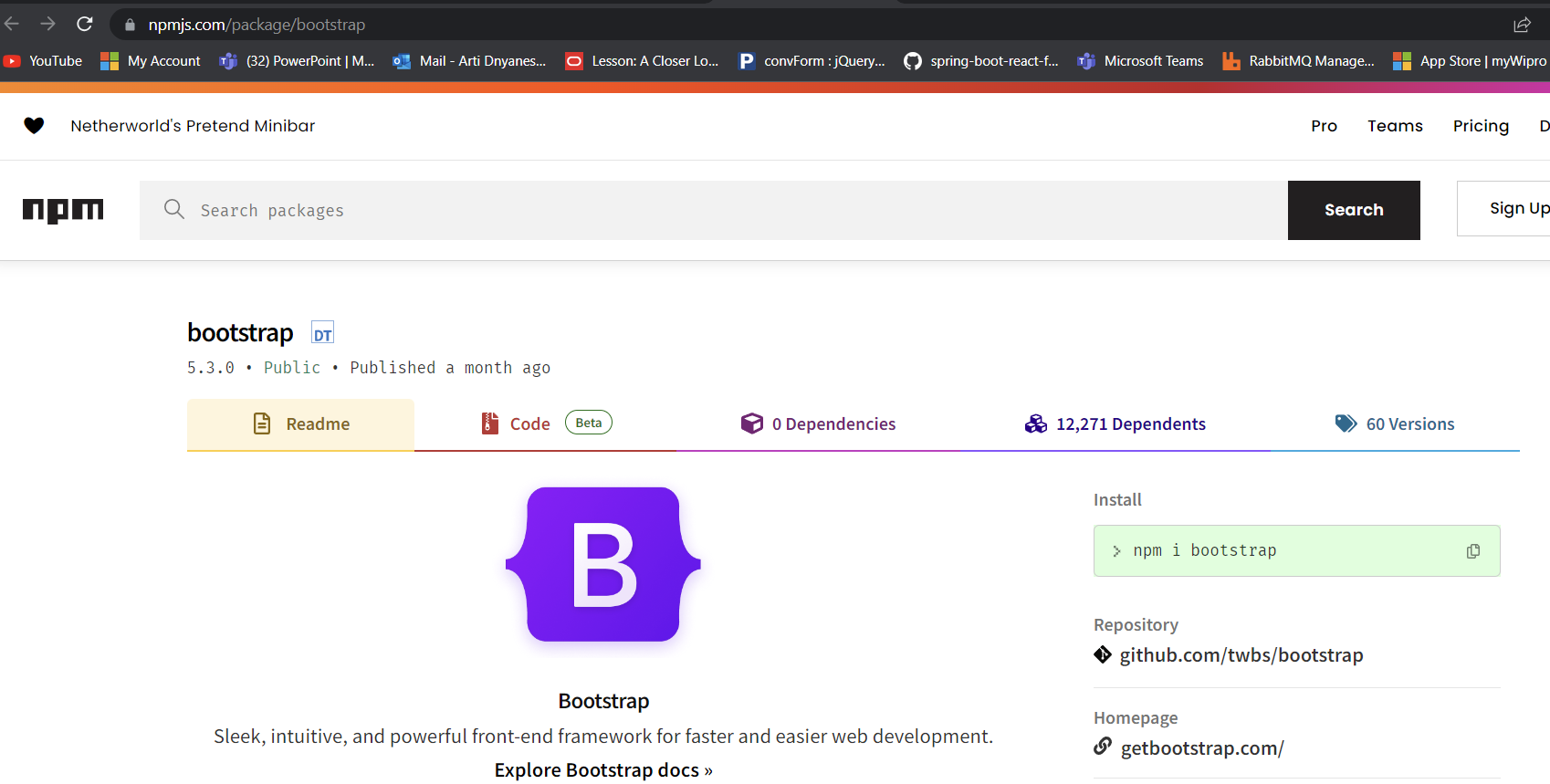
****

**To get this functionality Install snipplet extention in vscode**

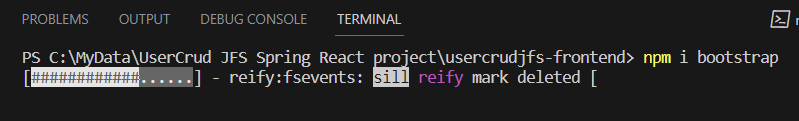


**Install Bootstrap🡪** Go to npm bootstrap site

Copy bootstrap installation command



In vs code

Terminal > 

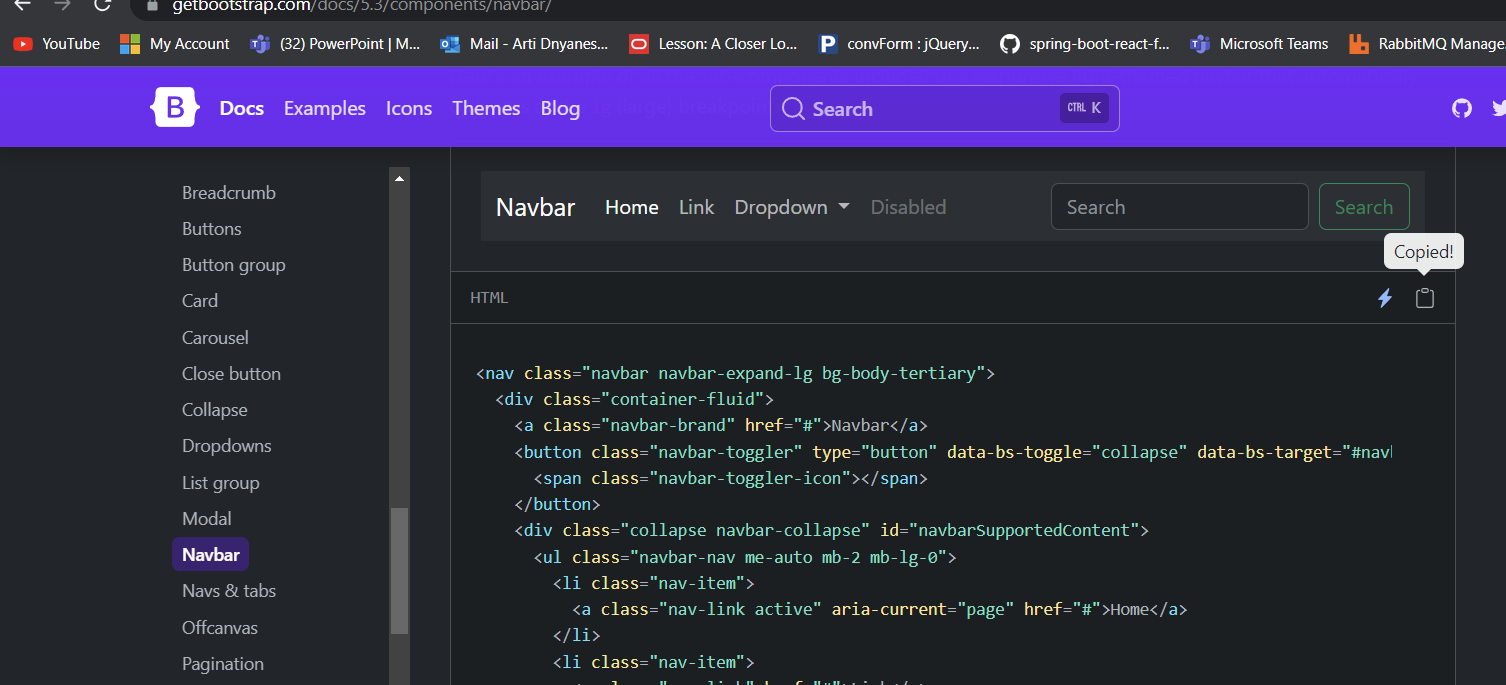
To use bootsrap into ouir application

Configure sm things

**Go to app.js**

**Import “../node\_modules/bootstrap/dist/css/bootstrap.min.css”;**

Then again go to> bootstrap>docs>navbar>**copy navbar component code**



Paste in vscode> Navbar.js> into <div> </div>

Chage html to js element ..for that > all “class” to “className”

Ctr+F or findall> class >replace with className

import React from 'react'

export default function () {

    return (

        <div>

            <nav className="navbar navbar-expand-lg bg-body-tertiary">

                <div className="container-fluid">

                    <a className="navbar-brand" href="#">Navbar</a>

                    <button className="navbar-toggler"

                        type="button"

                        data-bs-toggle="collapse"

                        data-bs-target="#navbarSupportedContent"

                        aria-controls="navbarSupportedContent"

                        aria-expanded="false"

                        aria-label="Toggle navigation"

                    >

                        <span className="navbar-toggler-icon"></span>

                    </button>

                </div>

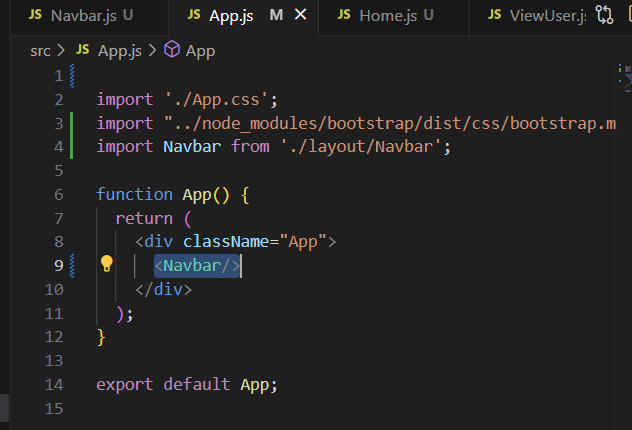
            </nav >

        </div >

    )

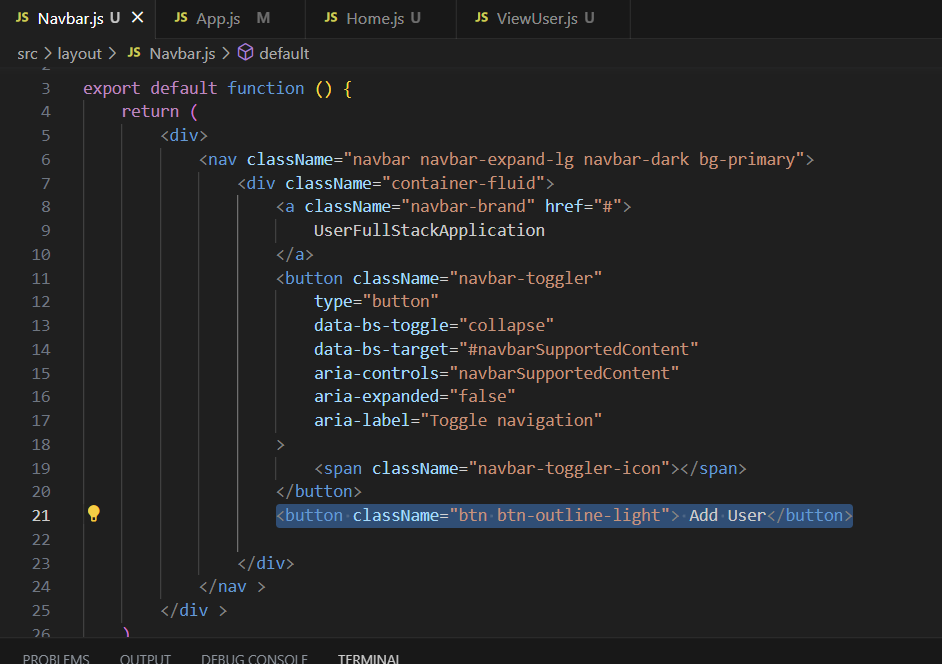
}

To call this navbar call it from app.js

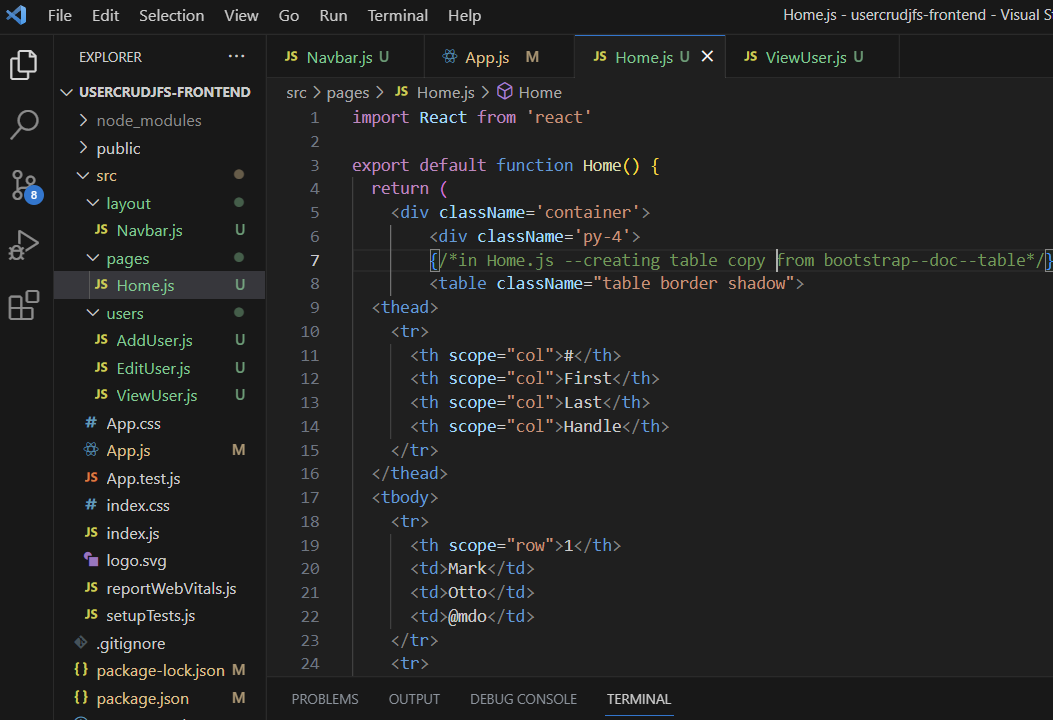


Rerun application >terminal> npm start

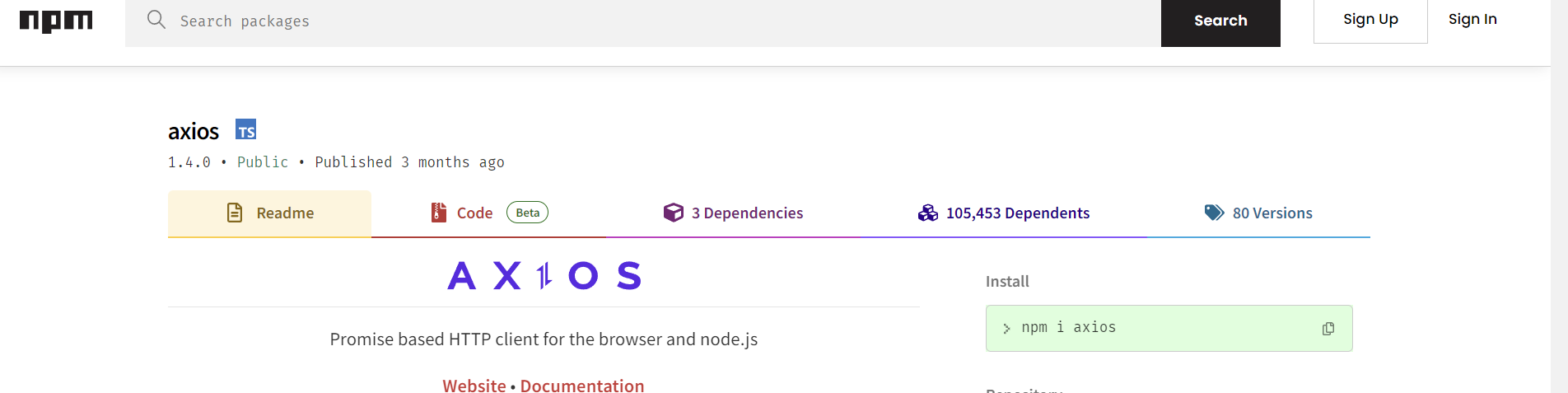
**Add btn> Add User in navbar**

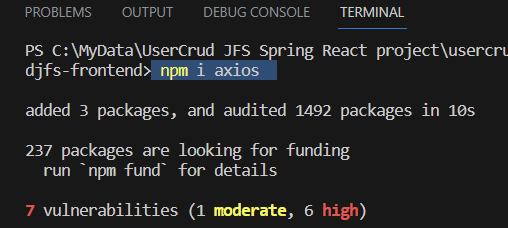




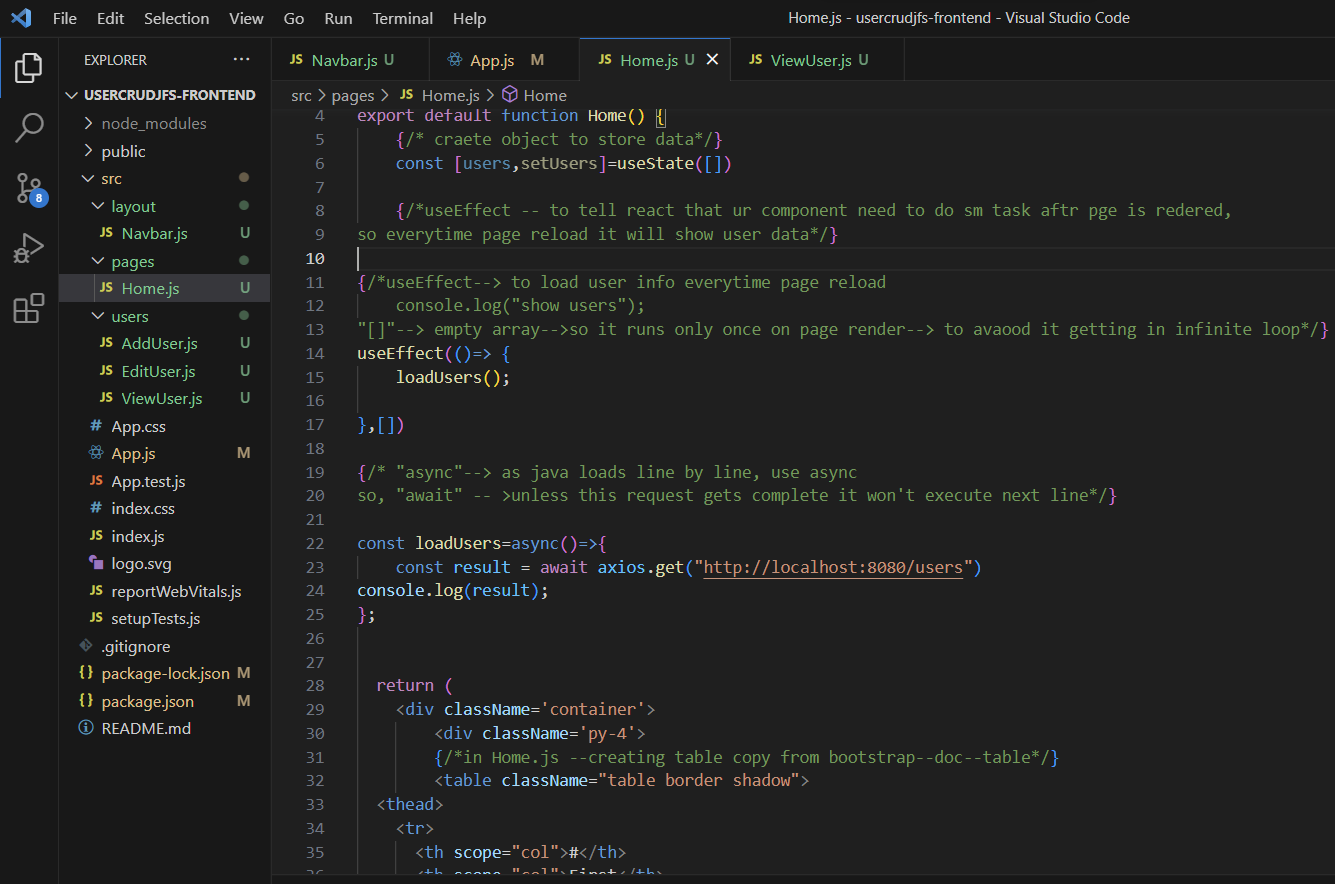


**Install axios🡪 to connect react to sprongboot**

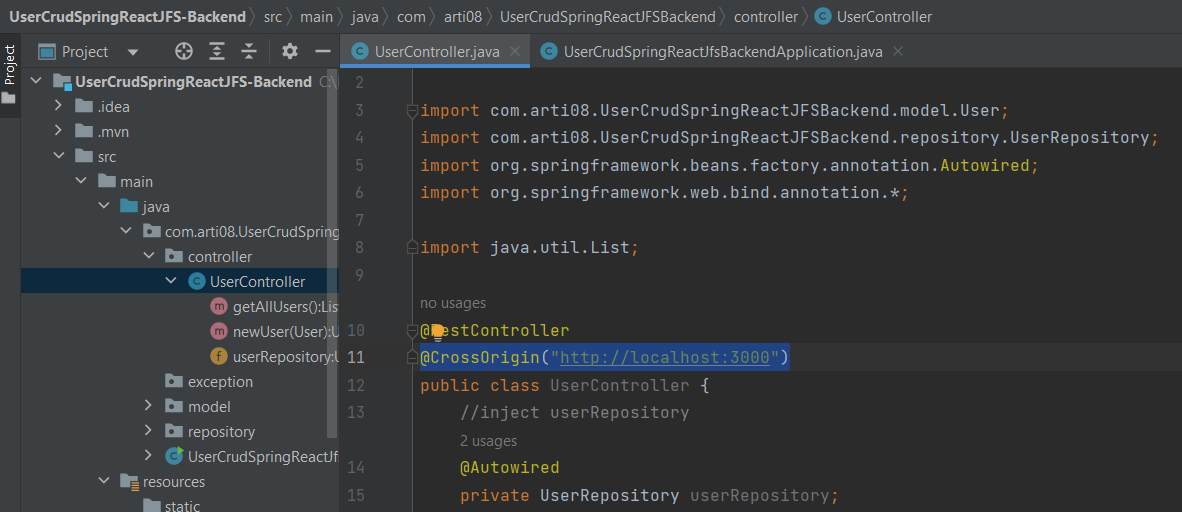


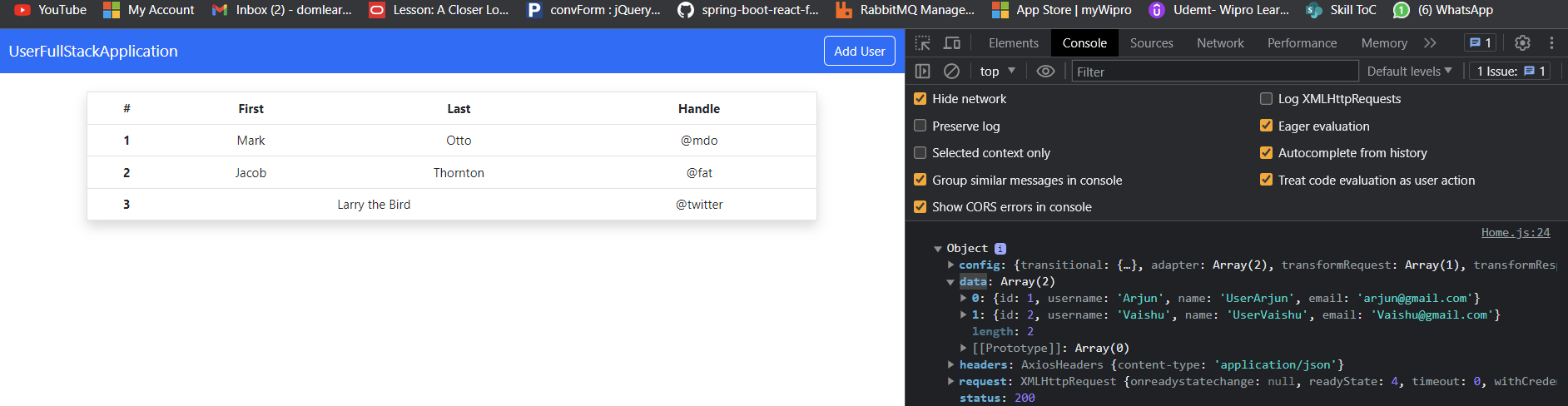


**Import axios from “axios” in Home.js**

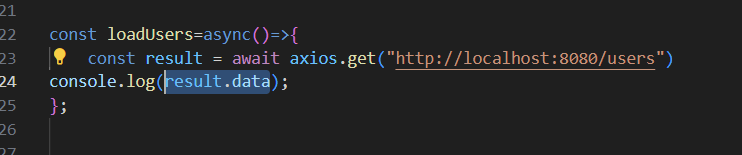


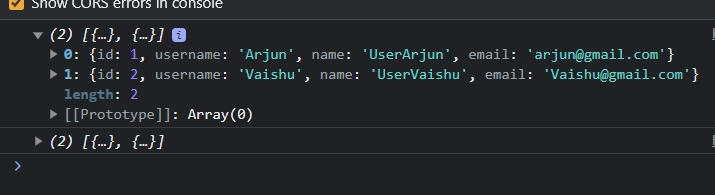
**//CONNECT FRONTEND TO BACKEND 🡪 CROSSOPRIGIN**





After using **result.data** instead of only result



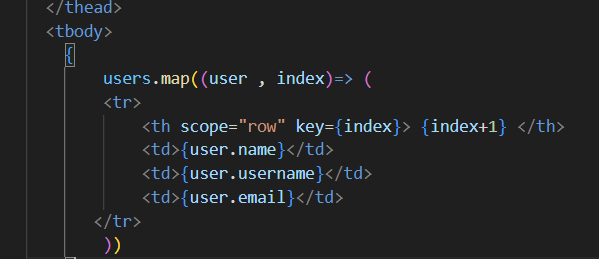


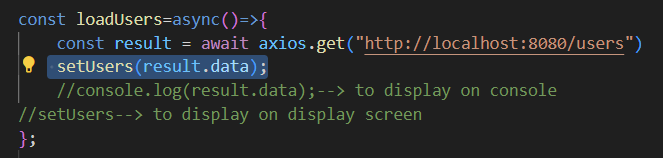
**To display this data on frontend screen**

In vscode> home.js

Instead of static data we will code for dynamic data to display from db

We will use **map** for that





import React, { useEffect, useState } from 'react';

import axios from "axios";

export default function Home() {

    {/\* craete object to store data\*/}

    const [users,setUsers]=useState([])

    {/\*useEffect -- to tell react that ur component need to do sm task aftr pge is redered,

so everytime page reload it will show user data\*/}

{/\*useEffect--> to load user info everytime page reload

    console.log("show users");

"[]"--> empty array-->so it runs only once on page render--> to avaood it getting in infinite loop\*/}

useEffect(()=> {

    loadUsers();

},[])

{/\* "async"--> as java loads line by line, use async

so, "await" -- >unless this request gets complete it won't execute next line\*/}

const loadUsers=async()=>{

    const result = await axios.get("http://localhost:8080/users")

    setUsers(result.data);

    //console.log(result.data);--> to display on console

//setUsers--> to display on display screen

};

  return (

    <div className='container'>

        <div className='py-4'>

        {/\*in Home.js --creating table copy from bootstrap--doc--table\*/}

        <table className="table border shadow">

  <thead>

    <tr>

      <th scope="col">#</th>

      <th scope="col">NAME</th>

      <th scope="col">USERNAME</th>

      <th scope="col">EMAIL</th>

      <th scope="col">ACTION</th>

    </tr>

  </thead>

  <tbody>

    {/\* map -- to get dynamic data from db\*/}

    {

        users.map((user , index)=> (

        <tr>

            <th scope="row" key={index}> {index+1} </th>

            <td>{user.name}</td>

            <td>{user.username}</td>

            <td>{user.email}</td>

       </tr>

        ))

    }

  </tbody>

</table>

        </div>

    </div>

  )

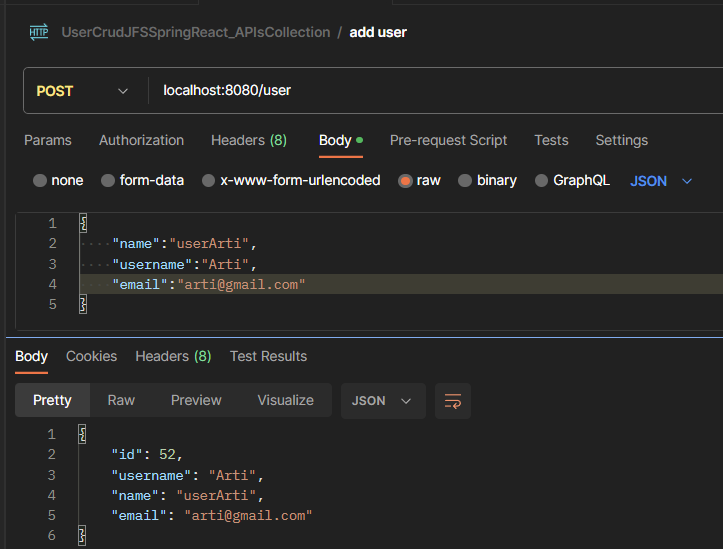
}

OUTPUT

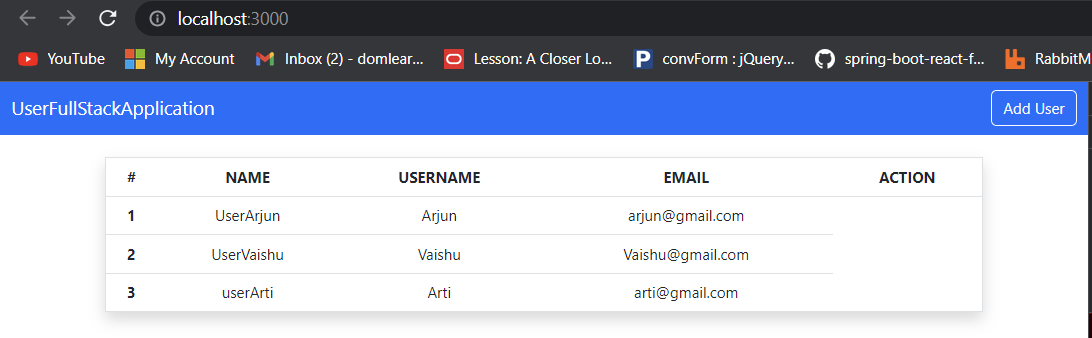


TEST THE APP ONCE

Try add user via postman post request

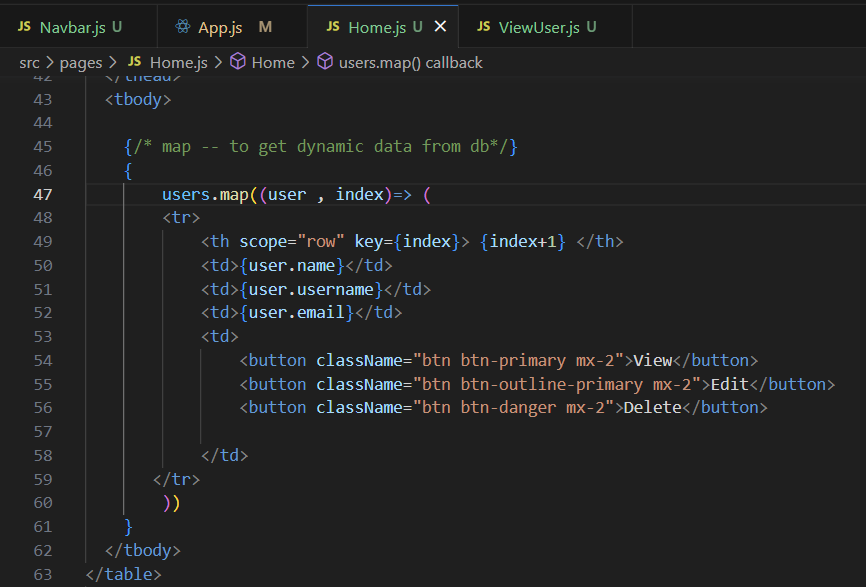


Output:





Create buttons:

****

**CODE FOR Adding user**

**AddUser.js**

**Rfc**

**//TO ADD ROUTING**

**GO TO npmjs.com**

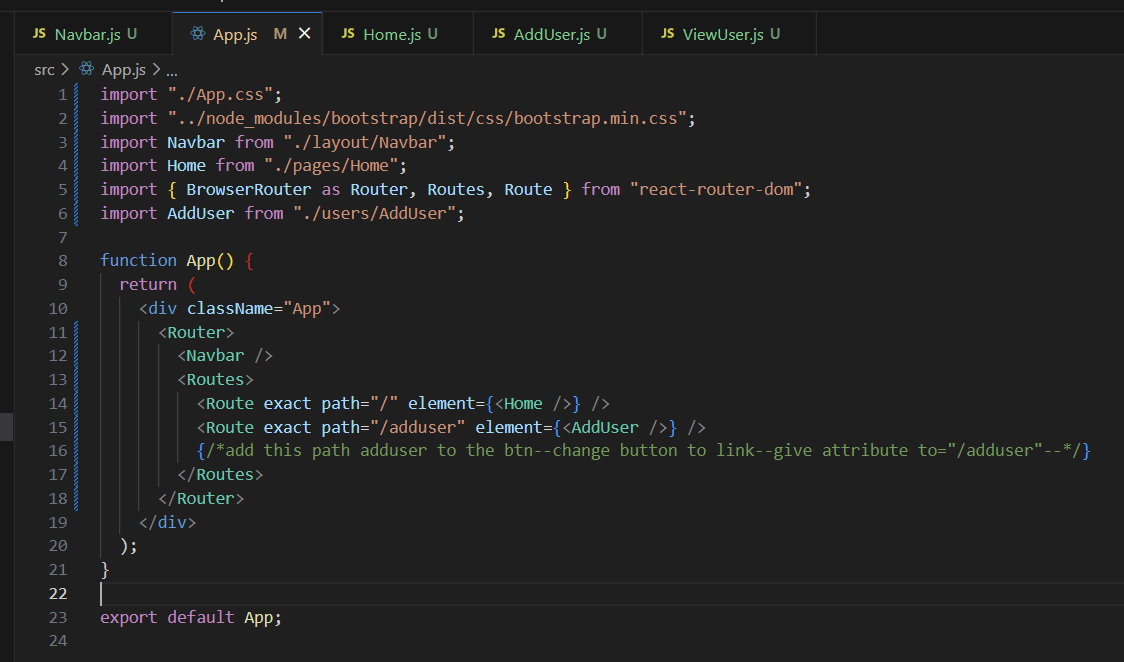
**React router dom 🡪 install in our vscode project**

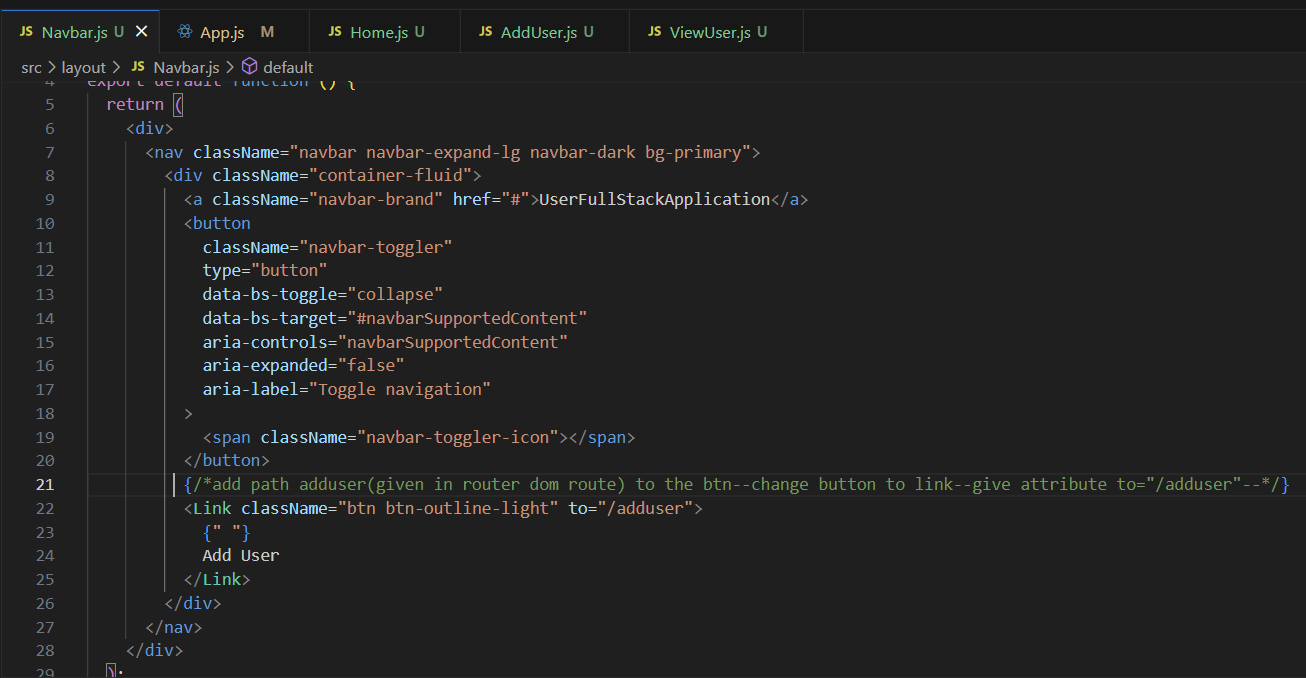
npm i react-router-dom

---------------------------------------

In app.js

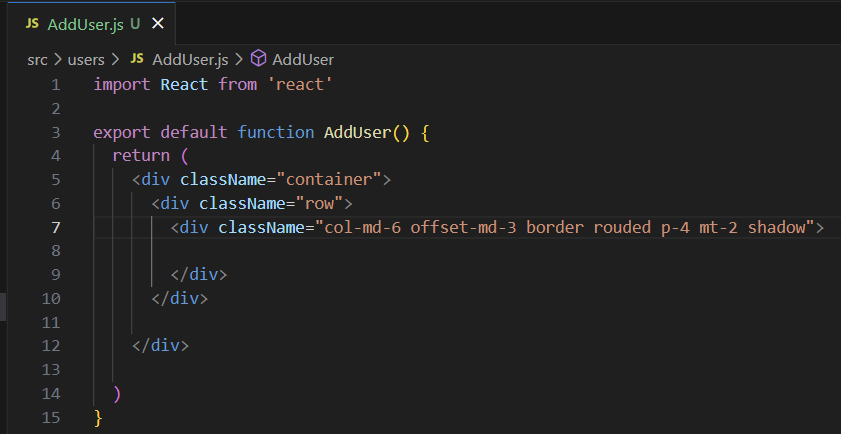
import { BrowserRouter as Router,Routes,Route } from 'react-router-dom';

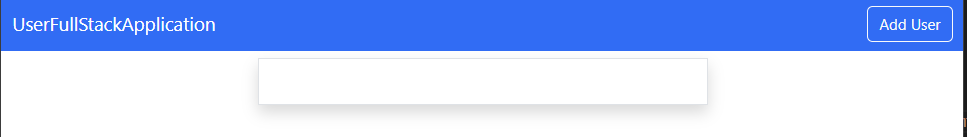






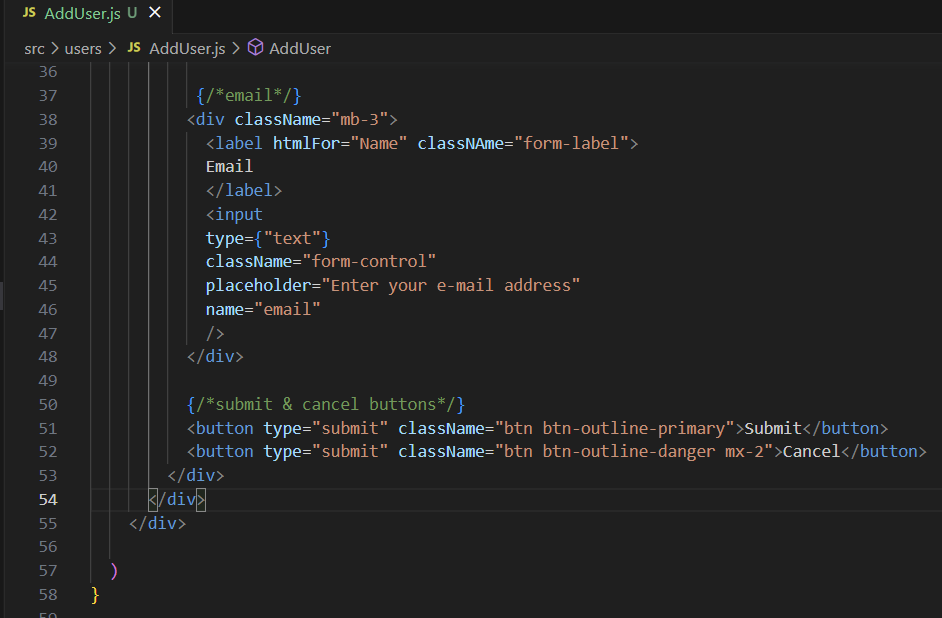
In AddUser.js

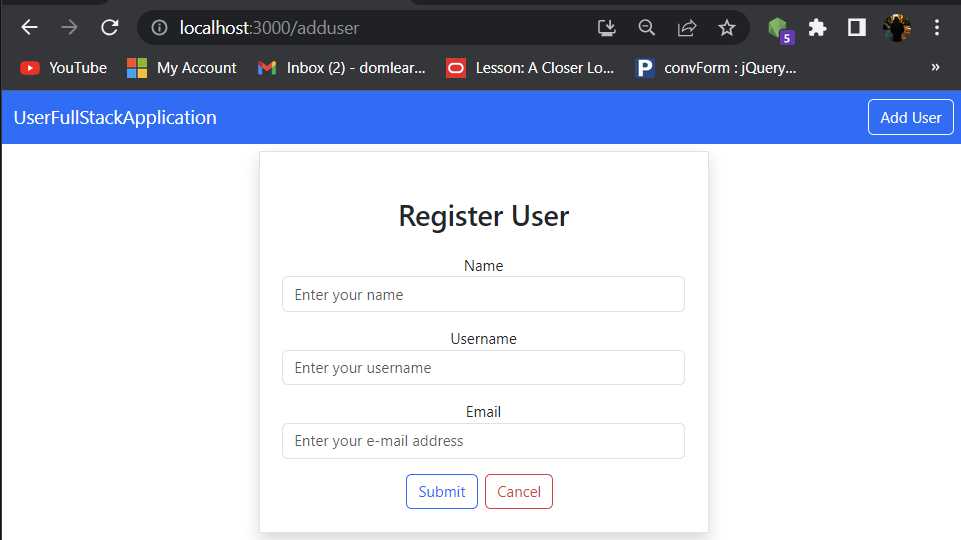




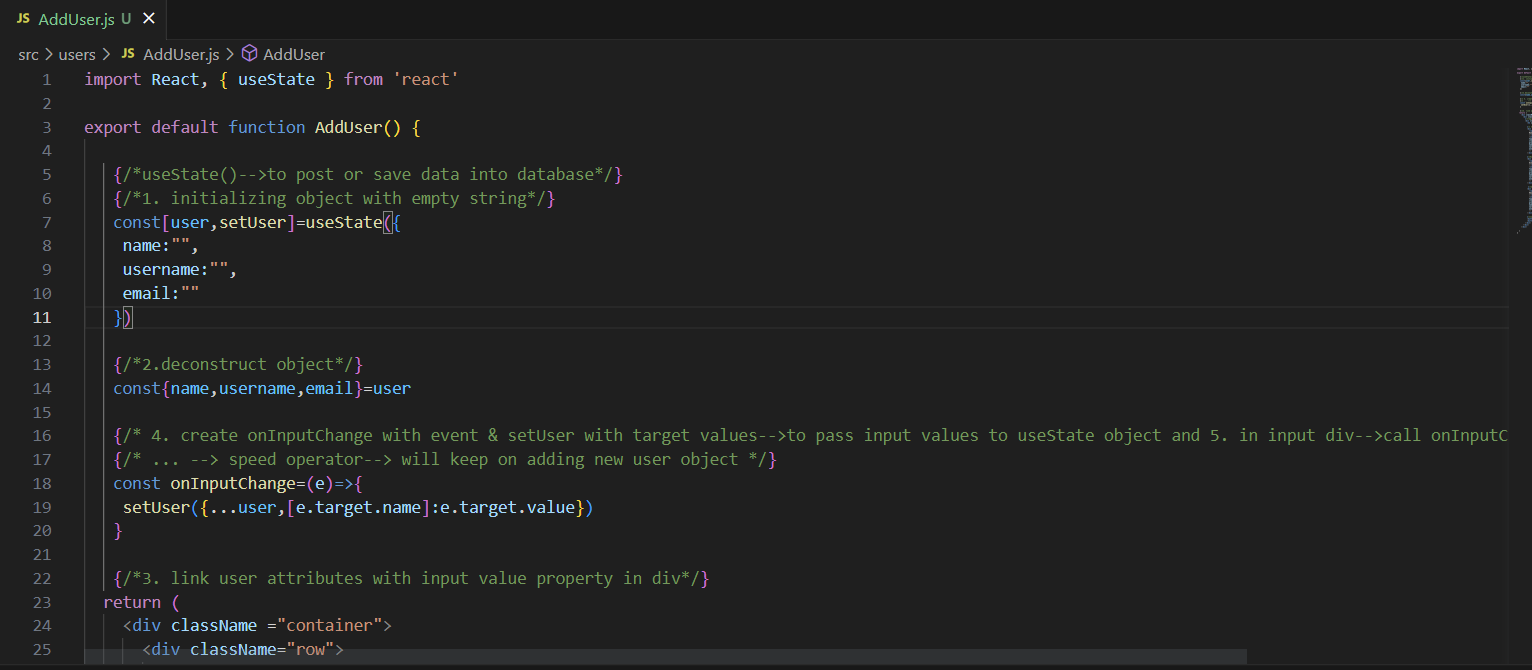
-------------------

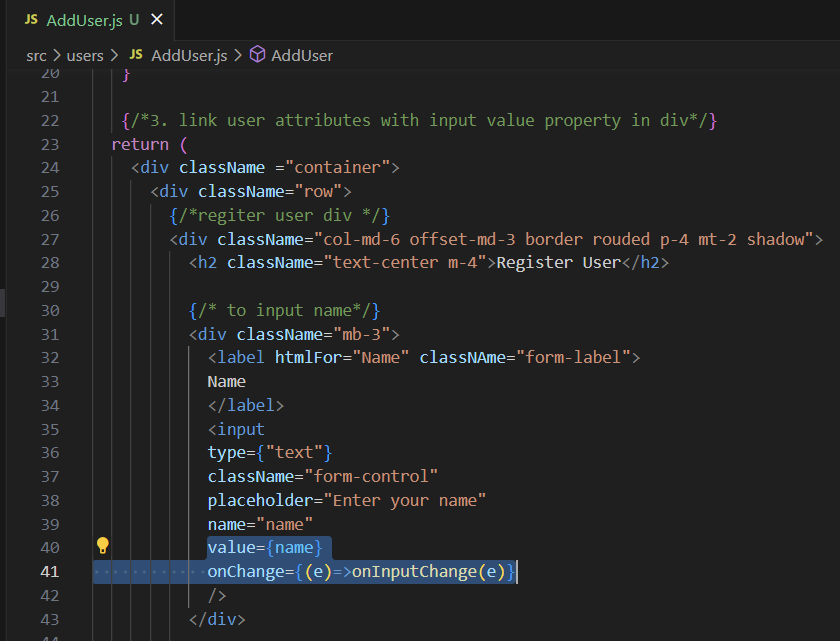








To post data into db  


****

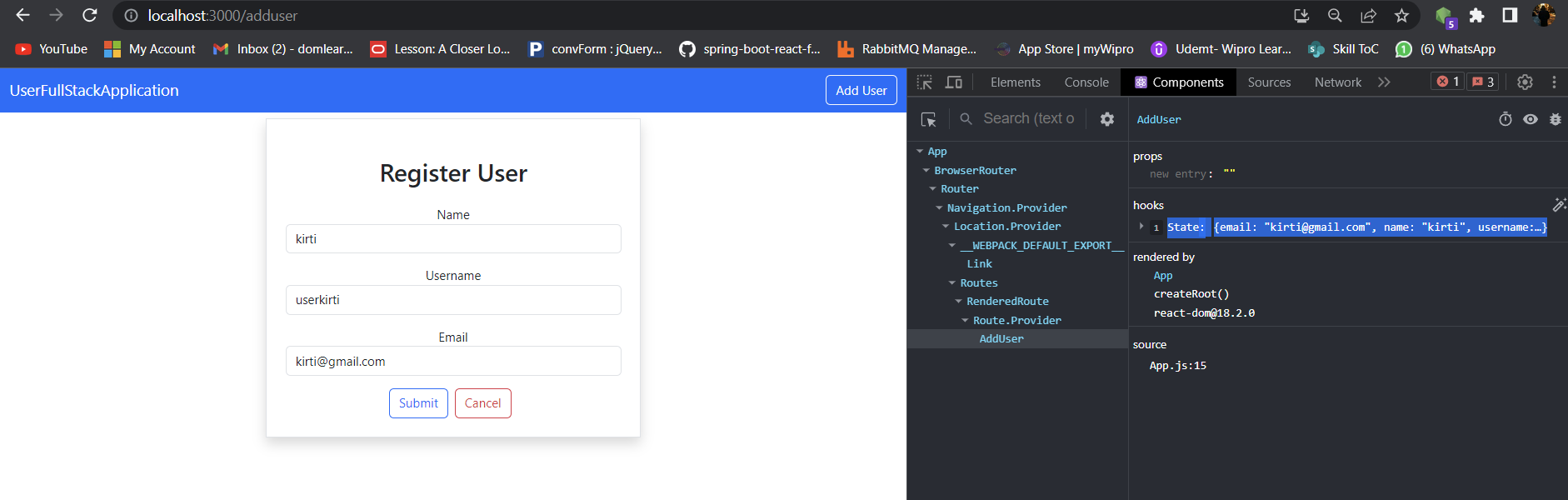
**WILL SEE OUTPUT🡪**

**U CAN EVEN INSTALL REACT DEVOLOPER TOOL IN CROME EXTENSION**

**CROME🡪 INSPECT🡪 IN ARROW🡪 COMPONENT🡪WE CAN SEE AddUser component which we created earlier**

**When u enter details in input**

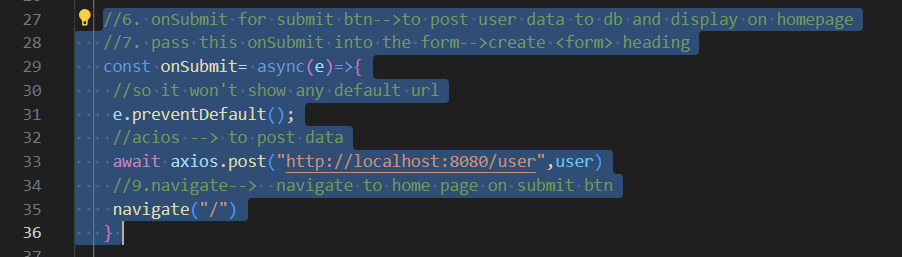
**THOS IS HOW WE CAN STORES INFORMATION IN STATE**

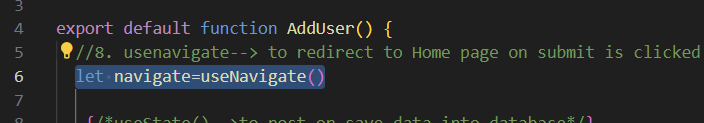
****

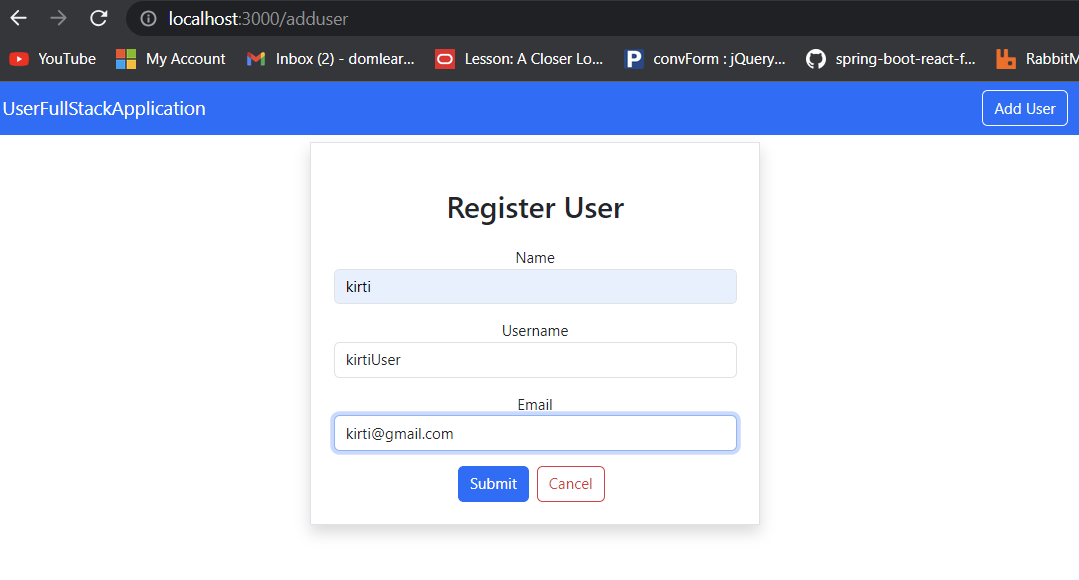


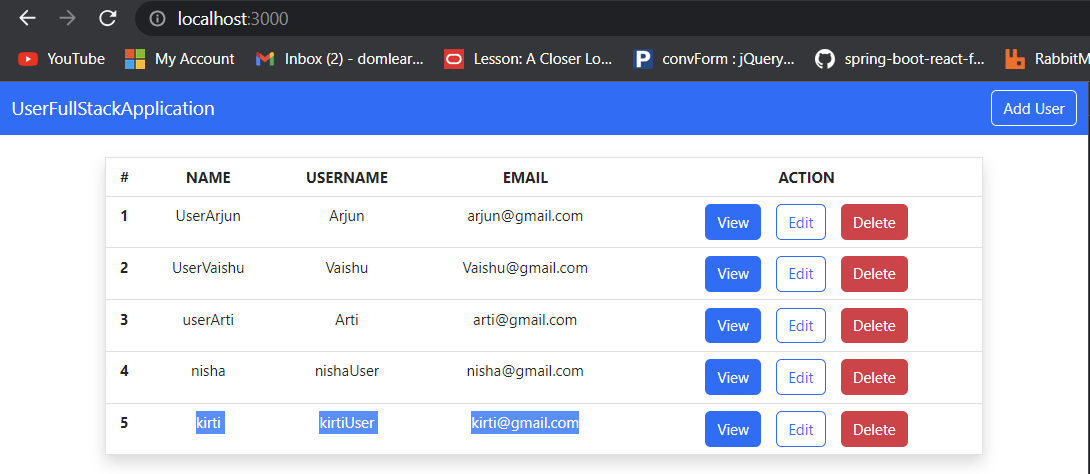
ON SUBMIT🡪 DATA SHOULD BE POSTED INTO DATABASE AND DISPLAY ON HOMEPAGE

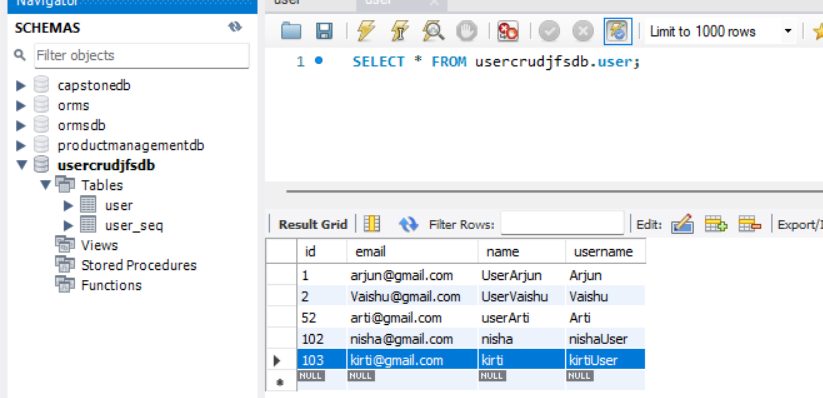
* CREATE onSubmit in AddUser.js







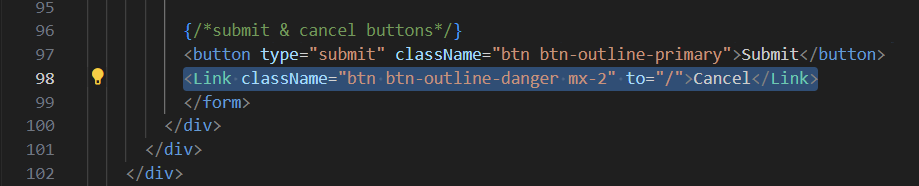




NEXT🡪WORK ON CANCEL BUTTON

In AddUSer.js

Change cancel <button> to a <Link> and give path

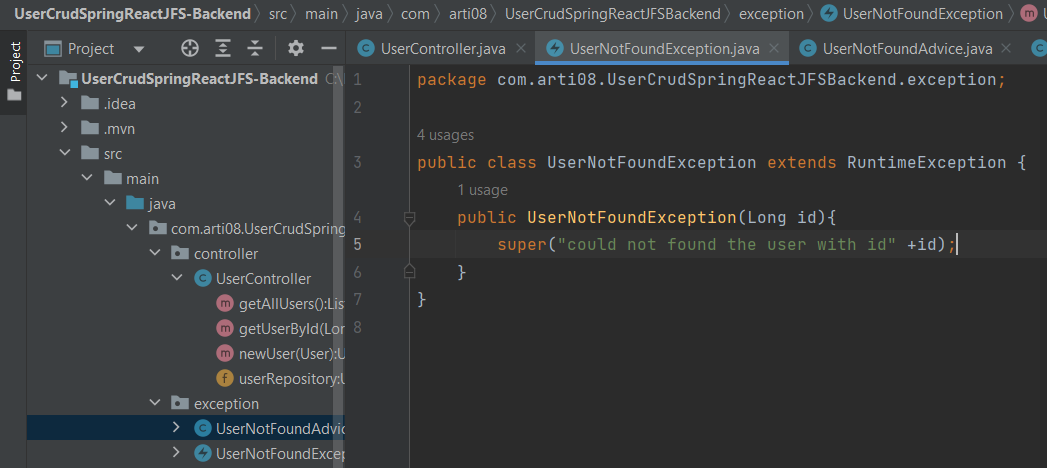


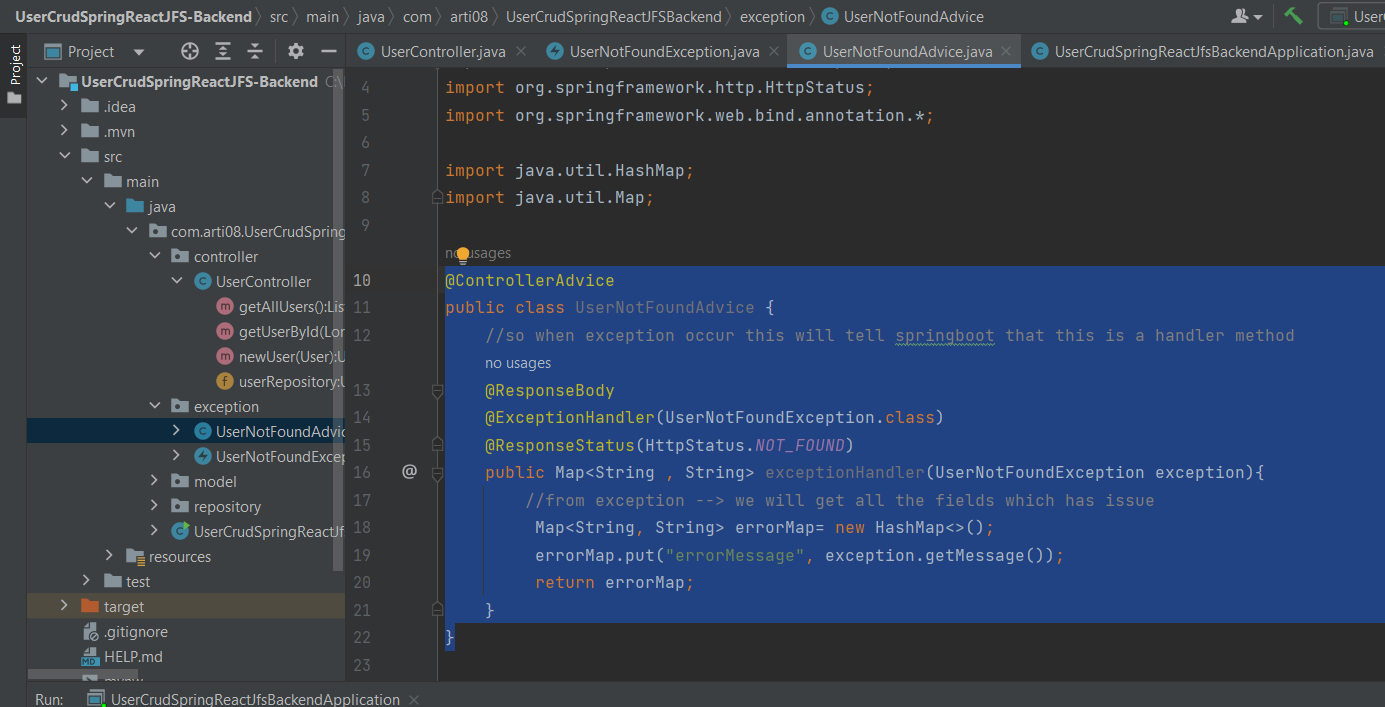
-----------------------------------------------------------------------------



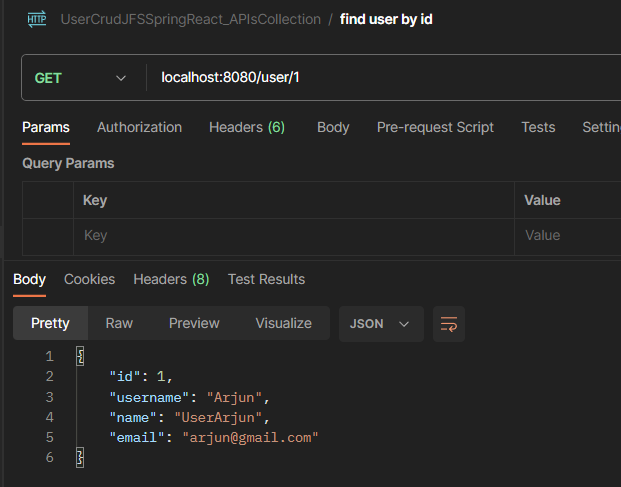
EDIT USER🡪

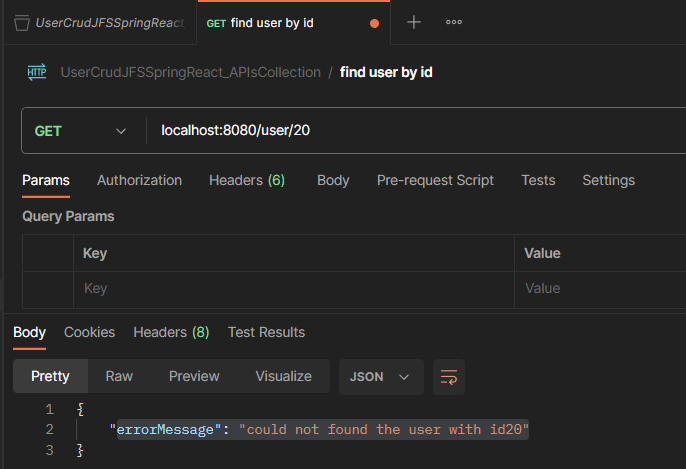
1. Writing code in springboot



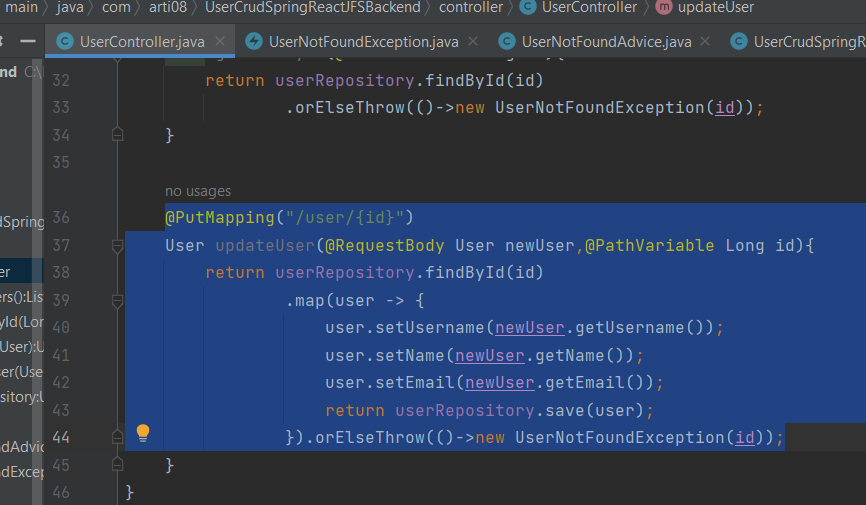


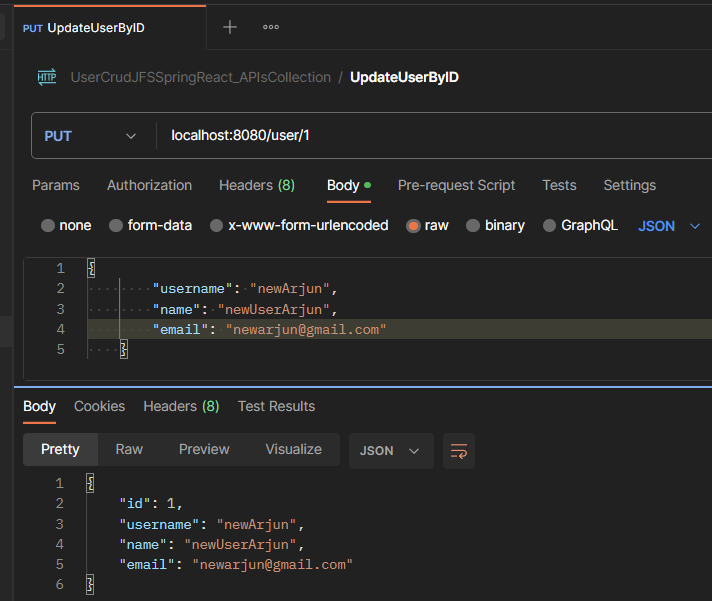
Check on postman

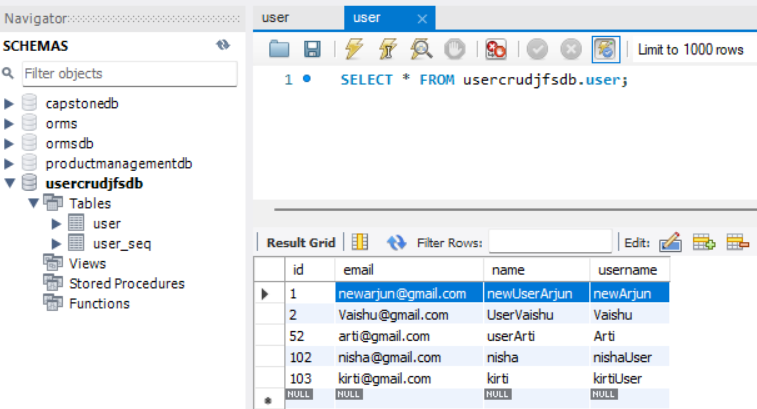






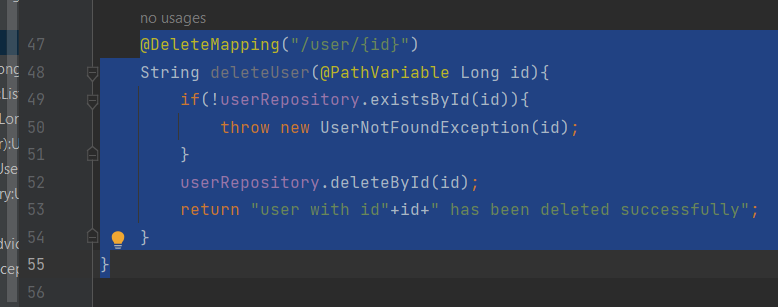


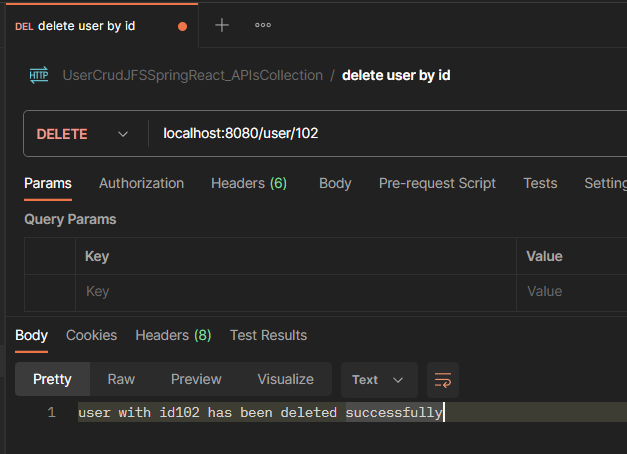


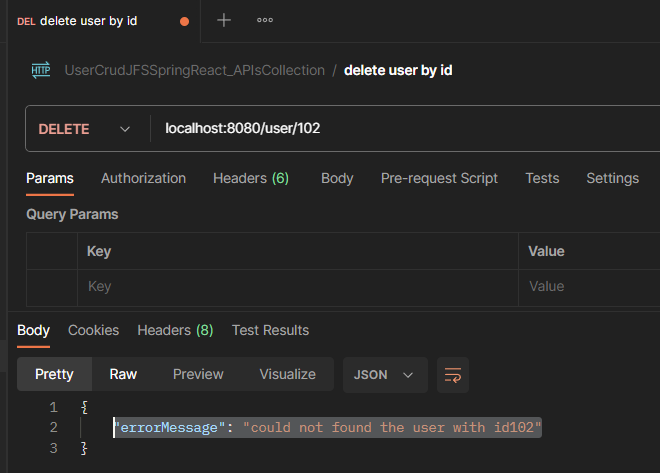


---------------------------------









-----------------------------------------------

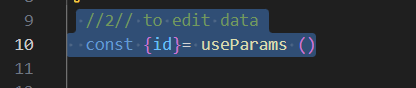
--------------------------------------------------------------------------------------

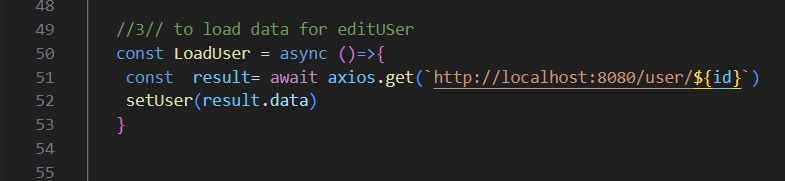
In frontewnd🡪 vscode

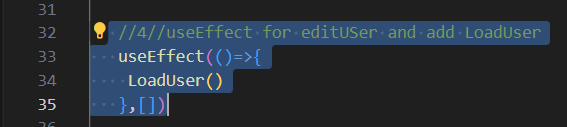


1. Copy paste all code from AddUser to EditUser🡪 change AddUser to EditUser

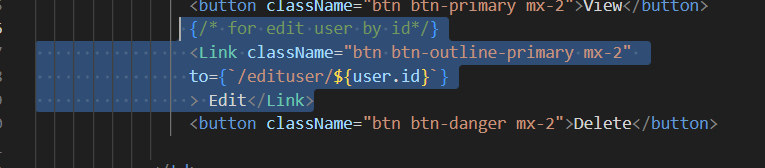




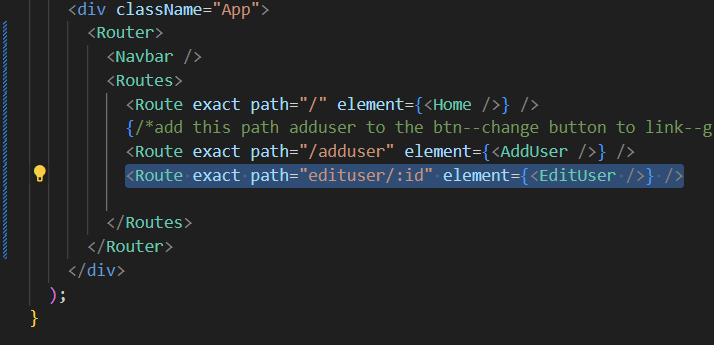


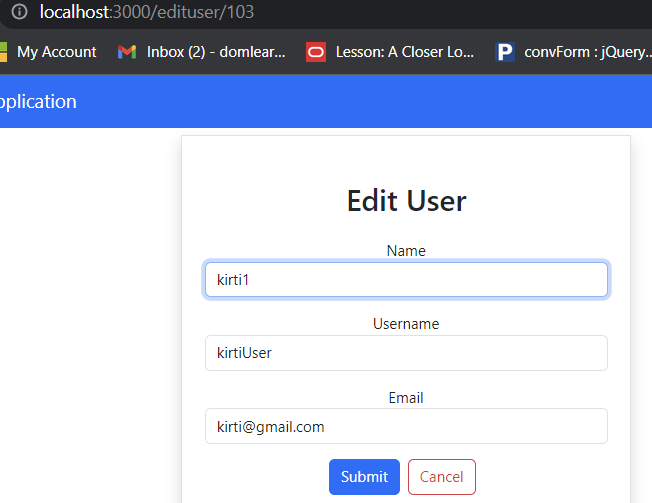


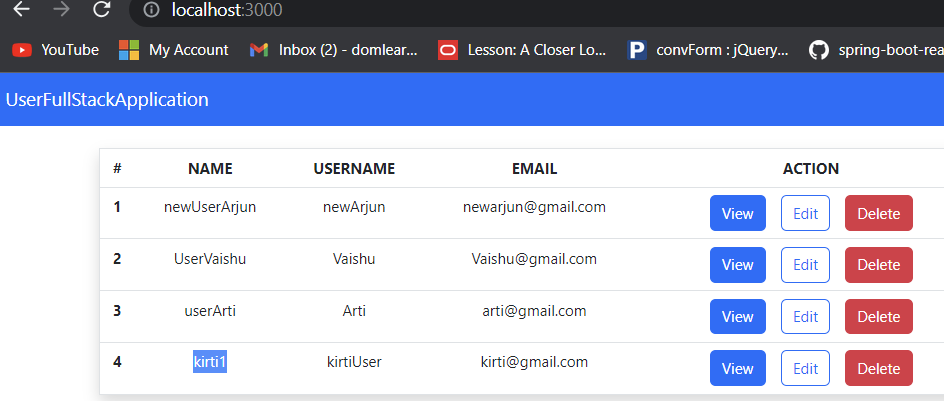
1. In home.js, change edit <button> to <Link> and give path with id



1. In app.js Add edituser to route





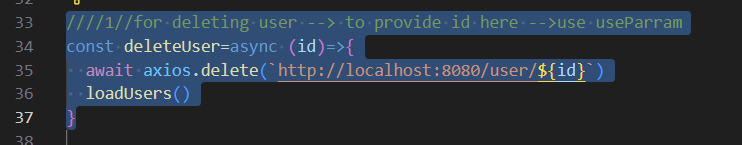


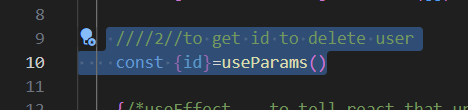
-----------------------------------------------

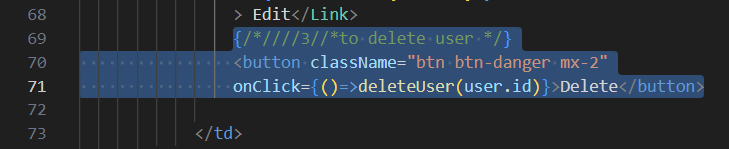
-------------------------------------------

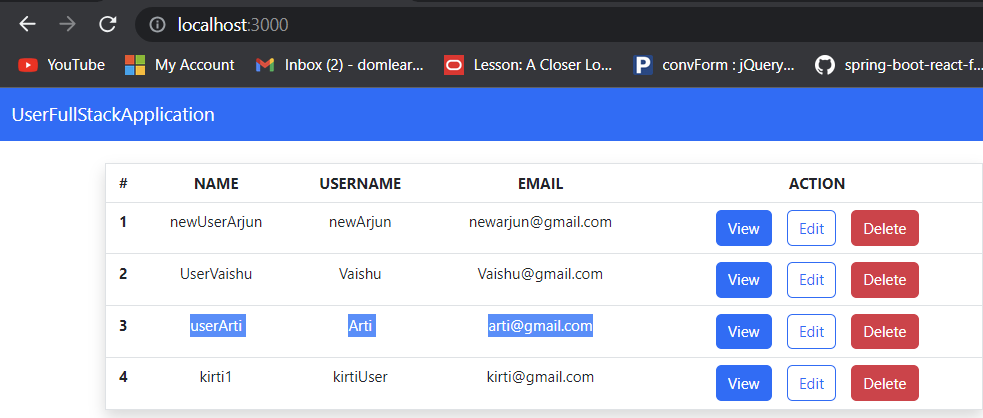


In home.js

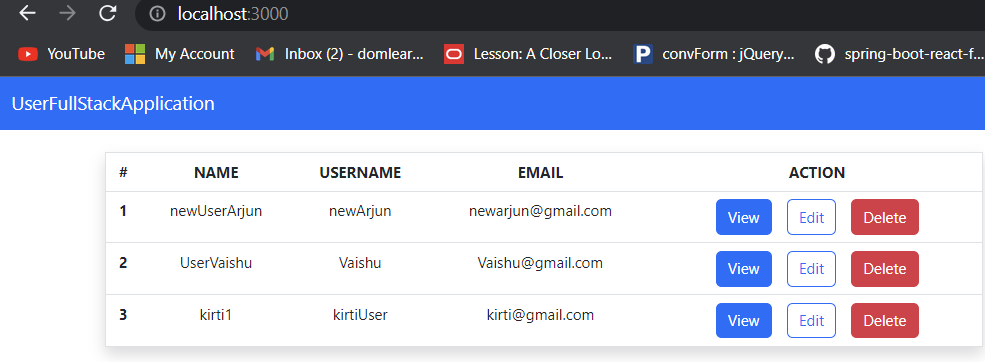


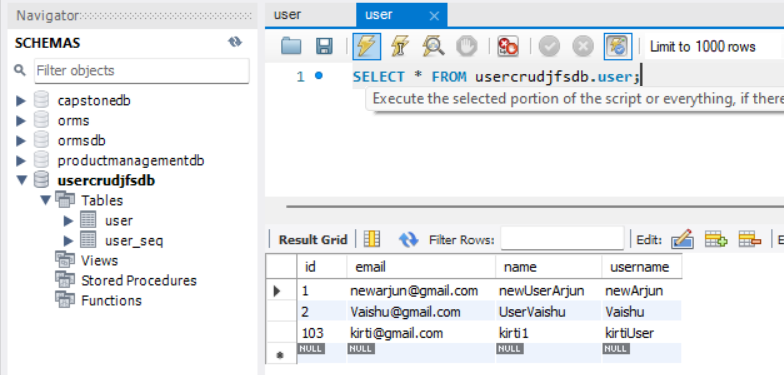






Delete user with id 3

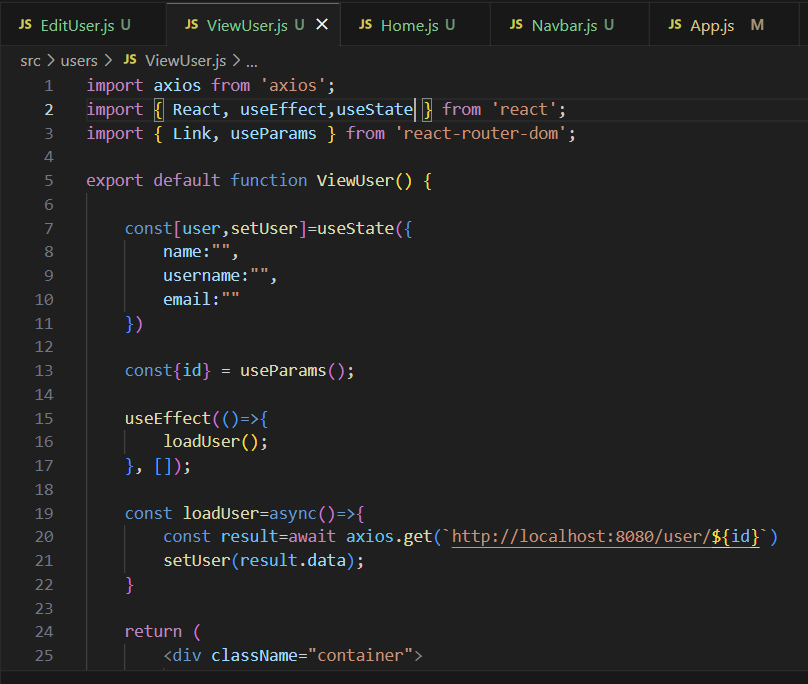


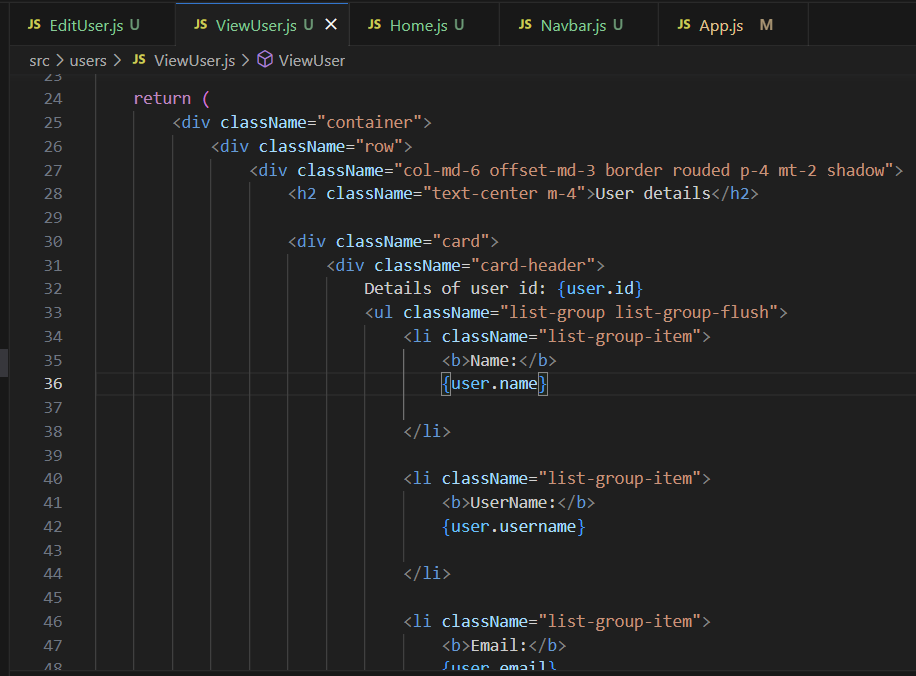


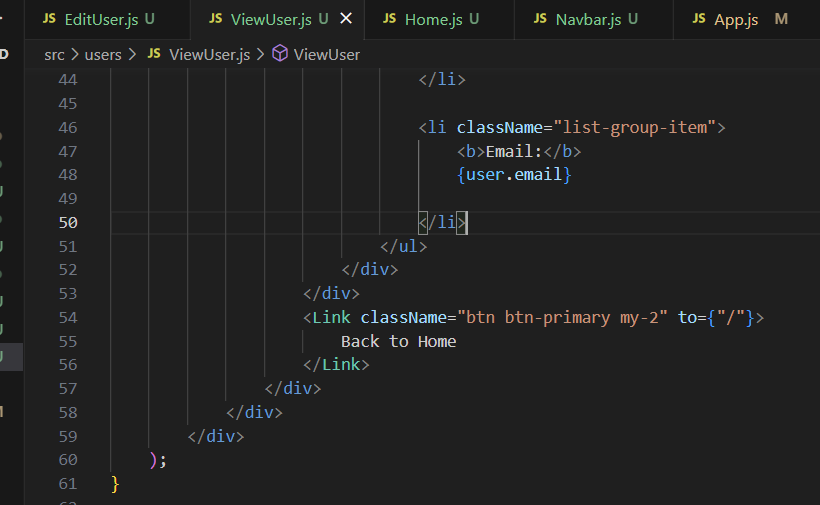
-------------------------------------------------------------

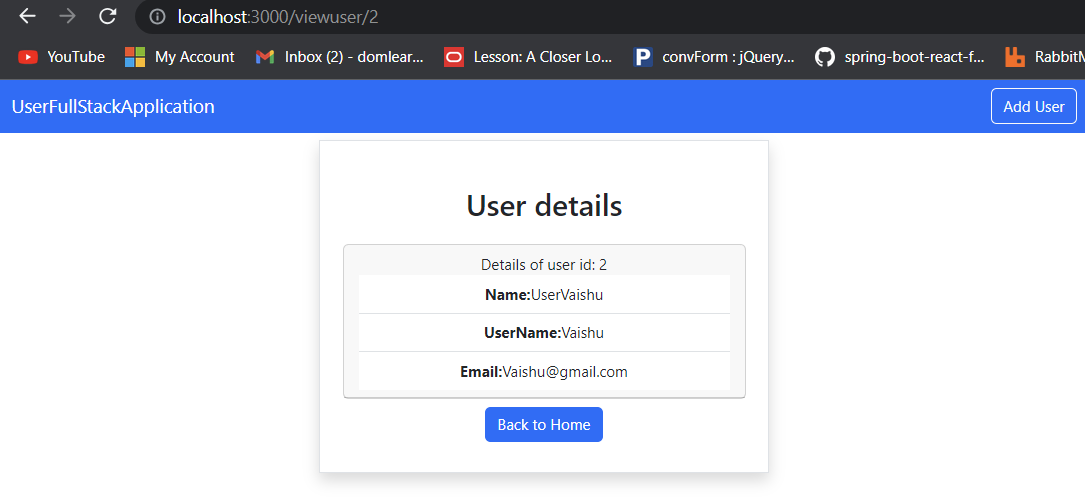


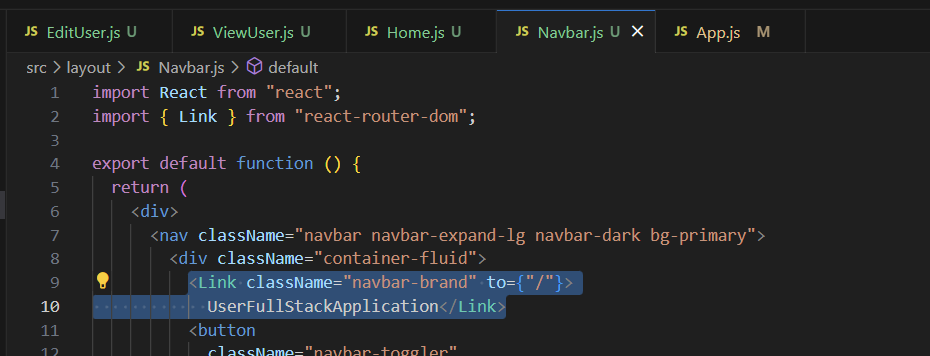
VIEW USER DETAILS:











-------------------------------------------------

-------------------------------------------------------------------------------