Name: Aarati Bhailal Dhuriyabans

Assignment :NotesApp using ReactJs:

Github: https://github.com/AaratiDhuriyabans/NotesApp

Cd notesapp

Npx create-react-app notesapp

App.js:

import React, { useState } from 'react';

import NavBar from './Components/NavBar';

import Form from './Components/Form';

import Notes from './Components/Notes';

import EditModal from './Components/EditModal';

export default function App()

{

  const [title,setTitle]=useState("");

  const [description,setDescription]=useState("");

  const [notes,setnotes]=useState(getNotesFromLs);

  const [editid,seteditid]=useState("");

  localStorage.setItem("notes",JSON.stringify(notes))

  console.log(notes);

  return(

   <>

    <div>

      <EditModal editid={editid} notes={notes} setnotes={setnotes}/>

      <NavBar/>

      <Form  title={title} setTitle={setTitle} description={description} setDescription={setDescription} notes={notes} setnotes={setnotes} />

      <div className="container justify-content-center">

        <div className="row">

          <h1 className='mb-3'>My Notes</h1>

          <div className="col-md-10" >

            {

              notes.length===0?

              <div className="card">

              <div className="card-header">

                Featured

              </div>

              <div className="card-body">

                <h5 className="card-title">Message</h5>

                <p className="card-text">No notes are Available</p>

                </div>

            </div>: notes.map((element)=>{

              return(

                <Notes element={element} key={element.id} notes={notes} setnotes={setnotes} seteditid={seteditid}/>

              )

            })        }

          </div>

        </div>

      </div>

    </div>

   </>

    )

    function getNotesFromLs()

    {

      const note=JSON.parse(localStorage.getItem("notes"));

      if(note)

      {

        return note

      }

      else{

        return [];

      }

    }

}

Index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

//import './index.css';

import App from './App';

//import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

//reportWebVitals();

Index.html:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

    <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

    <!--

      manifest.json provides metadata used when your web app is installed on a

      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

    -->

    <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

    <!--

      Notice the use of %PUBLIC\_URL% in the tags above.

      It will be replaced with the URL of the `public` folder during the build.

      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

      work correctly both with client-side routing and a non-root public URL.

      Learn how to configure a non-root public URL by running `npm run build`.

    -->

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

    <!--

      This HTML file is a template.

      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.

      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.

      To create a production bundle, use `npm run build` or `yarn build`.

    -->

  </body>

</html>

NavBar.js:

import React  from "react";

export default function NavBar()

{

    return(<>

    <nav className="navbar navbar-expand-lg text-bg-primary p-3 ">

  <div className="container-fluid">

    <a className="navbar-brand" href="#">NotesApp</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 className="collapse navbar-collapse" id="navbarSupportedContent">

      <ul className="navbar-nav me-auto mb-2 mb-lg-0">

        <li className="nav-item">

          <a className="nav-link active" aria-current="page" href="#">Home</a>

        </li>

        <li className="nav-item">

          <a className="nav-link" href="#">About</a>

        </li>

        </ul>

      <form className="d-flex" role="search">

        <input className="form-control me-2" type="search" placeholder="Search" aria-label="Search"></input>

         <button className="btn btn-outline-success" type="submit">Search</button>

      </form>

    </div>

  </div>

</nav>

    </>)

}

Form.js:

import React from "react";

export default function Form({title,setTitle,description,setDescription,notes,setnotes})

{

  const inputHandler=(e)=>{

    if(e.target.id==="title")

    {

      setTitle(e.target.value)

    }

    else{

      setDescription(e.target.value)

    }

    console.log(title,description);

  }

  const addNotesHandler=(e)=>{

    e.preventDefault();

   if(title!=="" && description!=="")

   { setnotes((note)=>{

    return(

      [...note,{

        title:title,

        description:description,

        id:new Date().getTime()

      }]

    )

  })

 }

    setTitle("");

    setDescription("");

  }

    return(<>

  <div className="container my-3">

    <div className="row justify-content-center">

        <div className="col-md-10">

        <form style={{border:"2px solid #ced4da",borderRadius:"10px",padding:"10px"}}>

  <div className="mb-3">

    <label for="title" className="form-label">Enter Title\*:</label>

    <input type="text" className="form-control" id="title" placeholder="Title"  value={title} onChange={inputHandler}/>

    </div>

  <div className="mb-3">

    <label for="desc" className="form-label">Enter Description\*:</label>

    <textarea name="desc" id="desc"  rows={"10"} className="form-control"  placeholder="Description" value={description} onChange={inputHandler}></textarea>

  </div>

   <button type="submit" className="btn btn-success" onClick={addNotesHandler}>Add Note</button>

</form>

        </div>

    </div>

  </div>

     </>)

}

Edit Modal:

import React from "react";

export default function EditModal({editid,notes,setnotes})

{

  const updateHandler=()=>{

   const updateNotes= notes.map((el)=>{

//console.log(el);

if(editid===el.id)

{

  return ({...el,

    title:document.getElementById("etitle").value,

    description:document.getElementById("edesc").value

  })

}

else{

  return el

}

    })

    console.log(updateNotes);

    setnotes(updateNotes);

  }

    return(

        <>

        {/\*<!-- Button trigger modal -->\*/}

{/\*<!-- Modal -->\*/}

<div className="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

  <div className="modal-dialog">

    <div className="modal-content">

      <div className="modal-header">

        <h1 className="modal-title fs-5" id="exampleModalLabel">Edit Notes</h1>

         </div>

      <div className="modal-body">

      <form>

  <div className="mb-3">

    <label for="etitle" className="form-label">Title\*:</label>

    <input type="text" className="form-control" id="etitle" placeholder="Title" />

    </div>

  <div className="mb-3">

    <label for="edesc" className="form-label">Description</label>

    <textarea name="desc" id="edesc"  rows={"10"} className="form-control"  placeholder="Description"></textarea>

  </div>

</form>

      </div>

      <div className="modal-footer">

        <button type="button" className="btn btn-primary" data-bs-dismiss="modal" onClick={updateHandler}>Edit Note</button>

        </div>

    </div>

  </div>

</div>

        </>

    )

}

Notes.js:

import React from "react";

export default  function Notes({element,notes,setnotes,seteditid})

{

  const removeHandler=(id)=>{

    //console.log(notes);//id

    const newNotes = notes.filter((emnt) => {

      return emnt.id !== id;

    });

      setnotes(newNotes);

    //  seteditid(id)

  }

  const EditHandler=(id)=>{

     // console.log(id);

     seteditid(id);

     notes.filter((el)=>{

      if(el.id===id)

      {

document.getElementById("etitle").value=el.title;

document.getElementById("edesc").value=el.description;

      }

     })

  }

    return(

        <>

        <div className="card">

  <div className="card-header">

    Featured

  </div>

  <div className="card-body" style={{textTransform:"capitalize"}}>

    <h5 className="card-title">{element.title}</h5>

    <p className="card-text">{element.description}</p>

    <button type="button" className="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal"

    onClick={()=>{

      EditHandler(element.id)

    }}

    >

  Edit

</button>

    <button className="btn btn-danger mx-3" onClick={()=>{

      removeHandler(element.id)

    }}> Delete</button>

  </div>

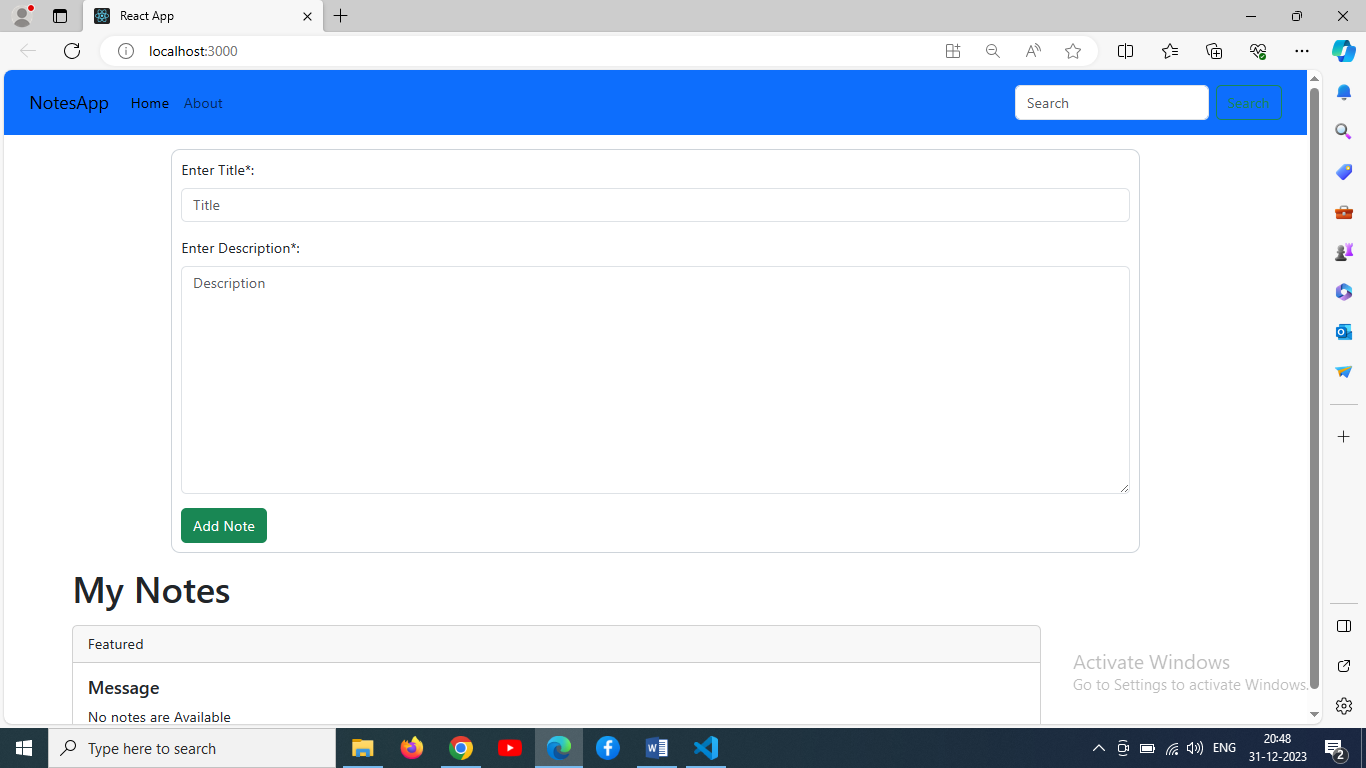
</div>

        </>

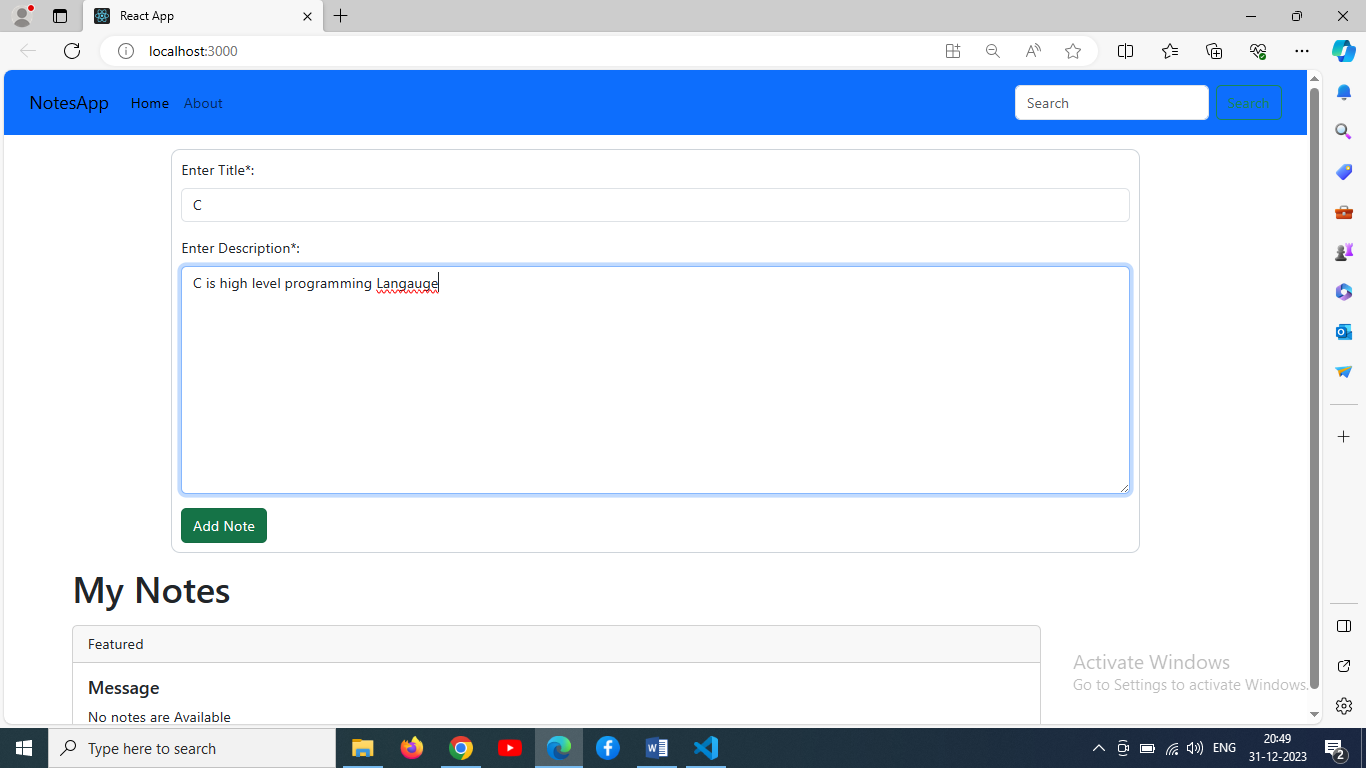
    )

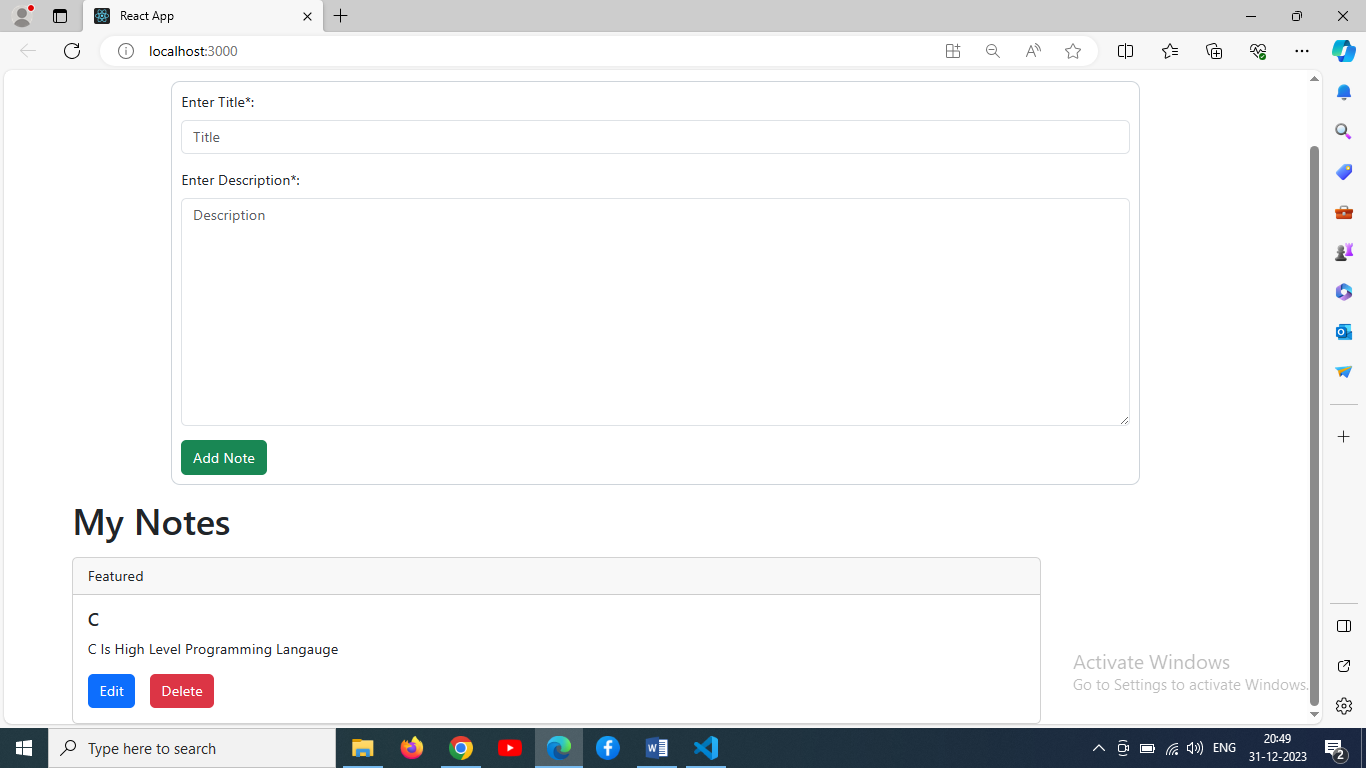
}

Output:

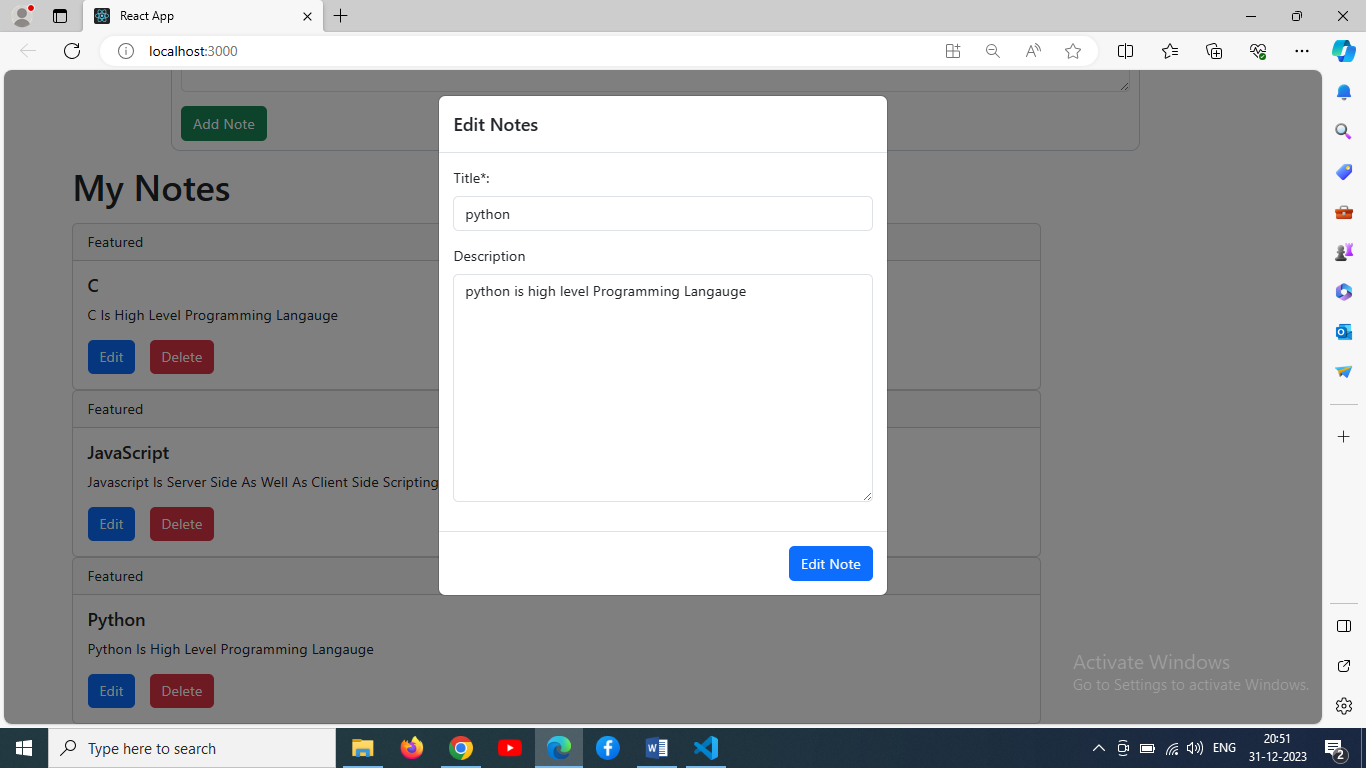


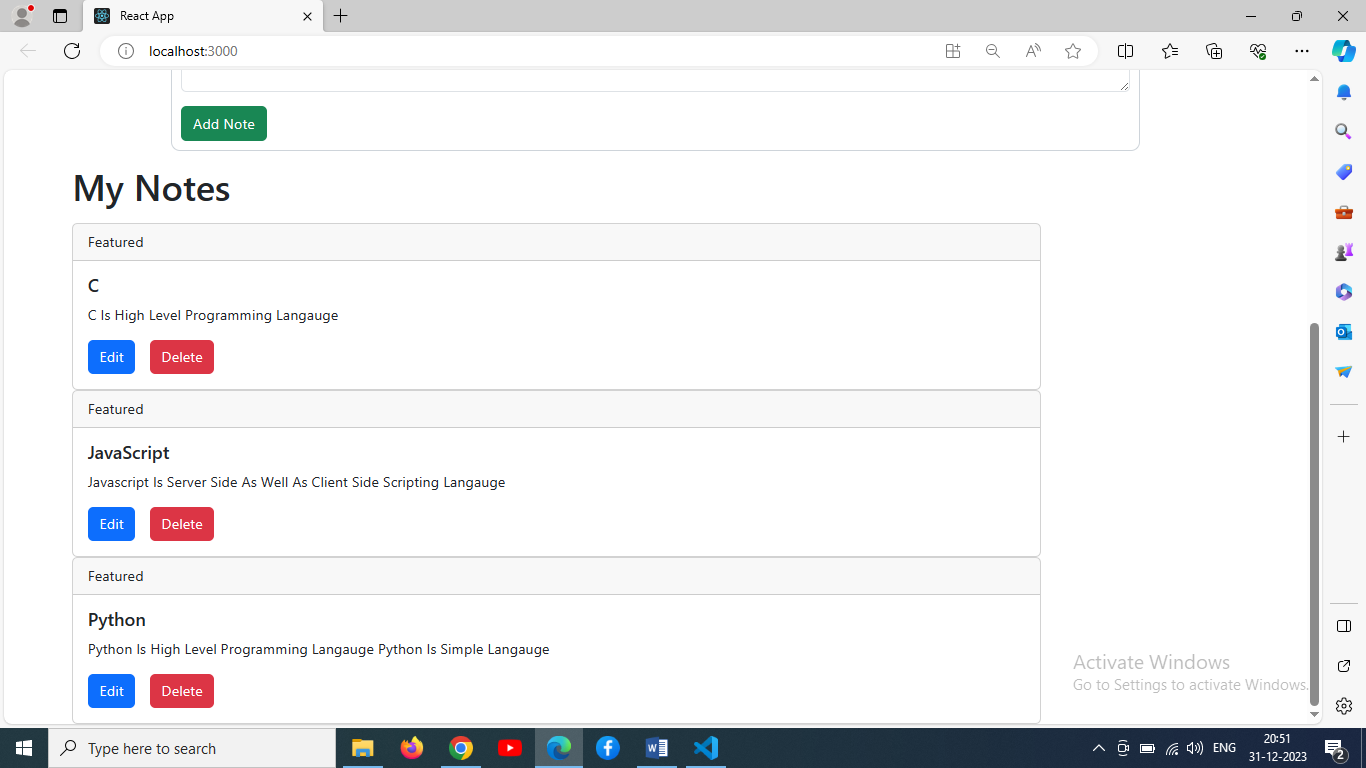
Adding some note:





EditNotes:





After removing one notes llike javascript:

