React Js

Intial :

import './App.css';

function App() {

  const name="Yaswanth";

  const uName=<h1>yash</h1>;

  const age=<h1>21</h1>;

  const email="bodduyaswanth9@gmail.com"

  const user=(

    <div>

      {name}

      {age}

      {email}

    </div>

  )

  return (

    <div className="App">

      {name}

      {uName}

      <h1>{name}</h1>

      {user}

      {user}

    </div>

  );

}

export default App;

Components:

File: App.js

import './App.css';

function App() {

  return (

    <div className="App">

    <GetDetails />

    <GetDetails />

    </div>

  );

}

const GetDetails = ()=>{

  return(

    <div>

      <h1>Yaswanth</h1>

      <h2>21</h2>

      <h1>bodduyaswanth@gmail.com</h1>

    </div>

  )

}

export default App;

Props:

import './App.css';

function App() {

  return (

    <div className="App">

    <GetDetails name="Yaswanth" age={21} email="bodduyaswanth@gmail.com"/>

    <GetDetails name="syam" age={20} email="syam@gmail.com"/>

    </div>

  );

}

const GetDetails = (props)=>{

  return(

    <div>

      <h1>{props.name}</h1>

      <h2>{props.age}</h2>

      <h1>{props.email}</h1>

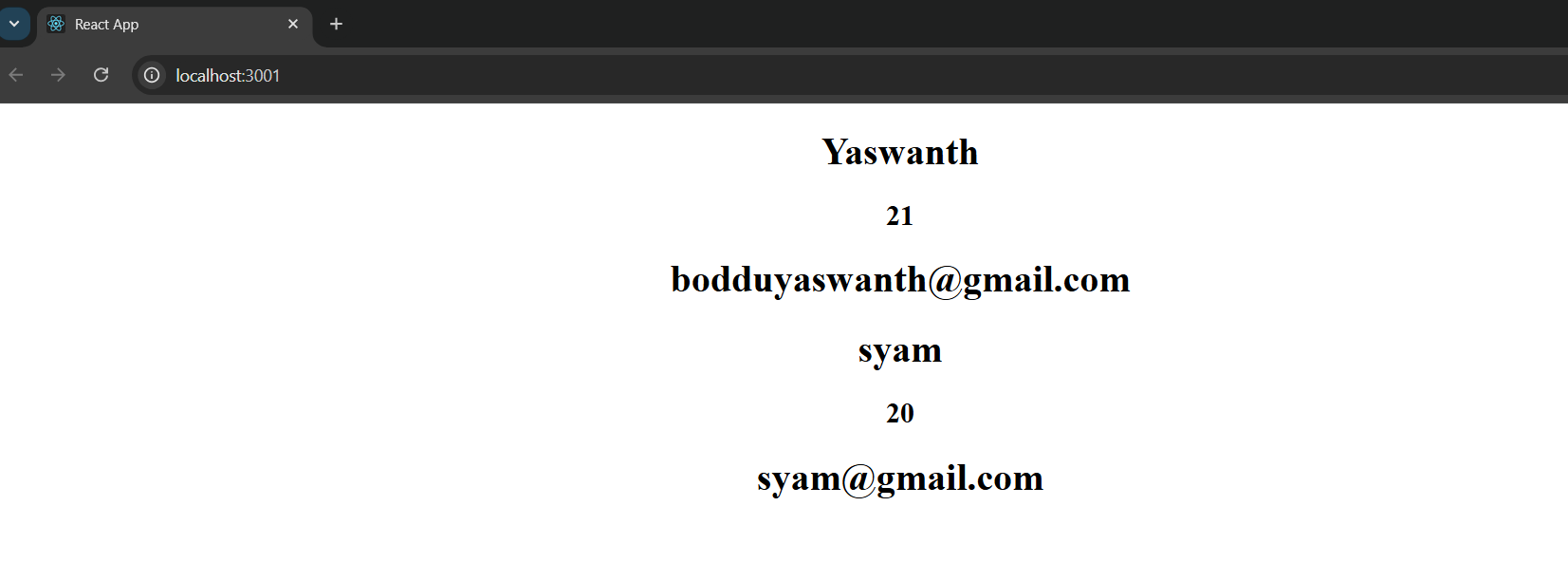
    </div>

  )

}

export default App;

Output:



Props – Example :

import './App.css';

function App() {

  return (

    <div className="App">

      <Job salary={90000} position="senior SDE" company="Amazon"/>

      <Job salary={12000} position="Junior SDE" company="Google"/>

      <Job salary={10000} position="project Manager" company="Netflix"/>

    </div>

  );

}

const Job=(props)=>{

  return(

    <div>

      <h1>{props.salary}</h1>

      <h1>{props.position}</h1>

      <h1>{props.company}</h1>

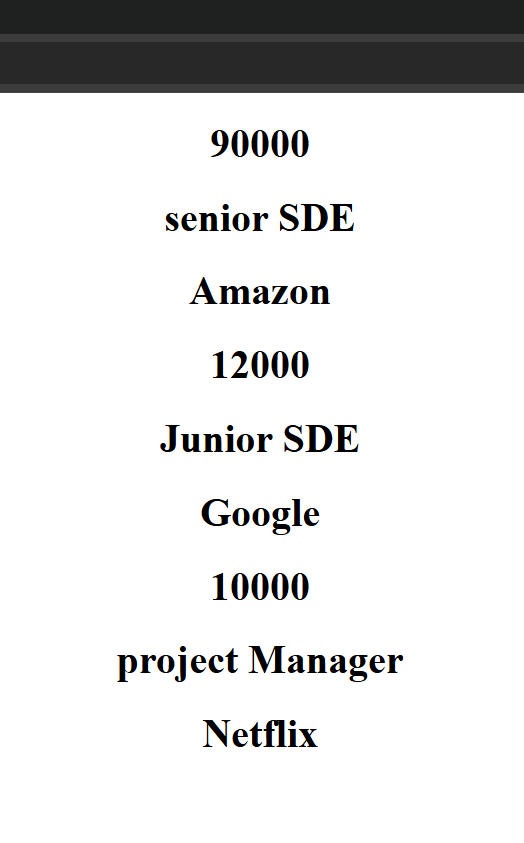
    </div>

  )

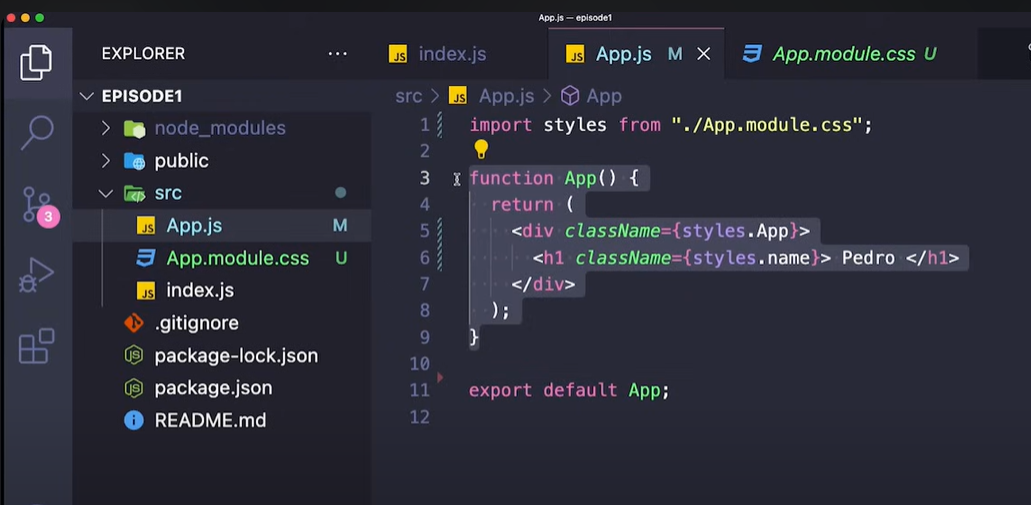
}

export default App;

Output:



Css Styles In ReactJs:



Ternary Operator:

import './App.css';

function App() {

  const age=18

  return (

    <div className="App">

      {age>=18 ? <h1>Under Age</h1> :<h1>Over Age</h1>}

    </div>

  );

}

export default App;

Different Operations by using Ternary Operator:

import './App.css';

function App() {

  const age=18

  const isGreen=true

  const isButton=false

  return (

    <div className="App">

      {age>=18 ? <h1>Under Age</h1> :<h1>Over Age</h1>}

      <h1 style={{color: isGreen ? "green" : "red"}}>This is color</h1>

      {isButton && <button>This is a Button</button>}

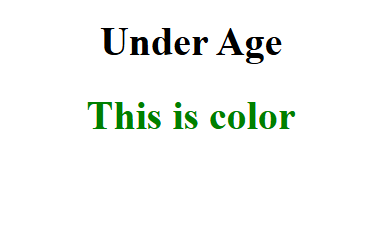
    </div>

  );

}

export default App;

Output:



Basic Array List in the ReactJs and fetch values by using map function:

import './App.css';

function App() {

  const names=["yaswanth","srinu","subbu","sai","saida","syam","sivagopi"]

  return (

    <div className="App">

      {names.map((name,key)=>{

        return <h1 key={key}>{name}</h1>

      })}

    </div>

  );

}

Basic Object in the ReactJs and fetch values by using map function:

import './App.css';

function App() {

  const users=[

    {name:"yaswanth", age:21},

    {name:"srinu" , age:20},

    {name:"subbu", age:19}

  ]

  return (

    <div className="App">

      {users.map((user,key)=>{

        return <User name={user.name} age={user.age}/>

      })}

    </div>

  );

}

const User=(props)=>{

  return(

    <div>

      {props.name}

      {props.age}

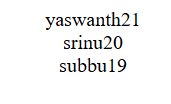
    </div>

  )

}

export default App;

Output:



Example:

import './App.css';

function App() {

const sports=[

{name:"KLR", profession:"cricket"},

{name:"Ronaldo", profession:"FootBall"},

{name:"Michel Jordan", profession:"BasketBall"}

]

return (

<div className="App">

{sports.map((person,key)=>{

return <Mens name={person.name} profession={person.profession}/>

})}

</div>

);

}

const Mens=(props)=>{

return(

<div>{props.name} {props.profession}</div>

)

}

export default App;

Example Props :

import './App.css';

function App() {

  const planets=[

    {name:"Mars", isGasPlanet:false},

    {name:"Earth", isGasPlanet:false},

    {name:"Jupiter",isGasPlanet:true},

    {name:"Venus",isGasPlanet:false},

    {name:"Neptune",isGasPlanet:true},

    {name:"Uranus",isGasPlanet:true}

  ];

  return (

    <div className="App">

      {planets.map((plant,key)=>{

        return <Planet name={plant.isGasPlanet && plant.name }/>

      })}

    </div>

  );

}

const Planet=(props)=>{

  return(

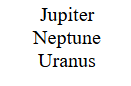
    <div>{props.name}</div>

  )

}

export default App;

Output:



States in React, useState Hook:

Example – 1:

import './App.css';

import {useState} from "react"

function App() {

  const [age,setAge]=useState(0);

  const increaseAge=()=>{

    setAge(age+1);

  };

  const decreaseAge=()=>{

    setAge(age-1);

  }

  const resetAge=()=>{

    setAge(0);

  }

  return (

    <div className="App">

      {age}

      <button onClick={increaseAge}>Increment</button>

      <button onClick={decreaseAge}>Decrease</button>

      <button onClick={resetAge}>Reset</button>

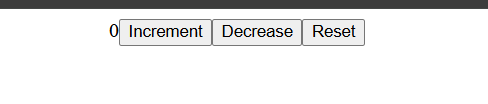
    </div>

  );

}

export default App;

Output:



Example – 2: Same text appeared when you entered the text in input box.

import './App.css';

import {useState} from "react"

function App() {

  const [inputValue,setInputValue]=useState("");

  const handleInputChange=(event)=>{

    setInputValue(event.target.value)

  }

  return (

    <div className="App">

    <input type='text' onChange={handleInputChange}/>

    {inputValue}

    </div>

  );

}

export default App;

Output:



Example – 3: show text and hide based on the button by using use state

import './App.css';

import {useState} from "react"

function App() {

  const [showText,setShowText]=useState(true);

  return (

    <div className="App">

    <button onClick={()=>{setShowText(!showText)}}>show/hide</button>

    {showText && <h1>This is KLR</h1>}

    </div>

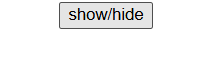
  );

}

export default App;

Output:





Example – 4 : Change the text color by clicking the button:

import './App.css';

import {useState} from "react"

function App() {

  const [textColor,setTextColor]=useState("black");

  return (

    <div className="App">

      <button onClick={()=>{

        setTextColor(textColor === "black" ? "red" : "black");

      }}>show/Hidden</button>

      <h1 style={{color:textColor}}>This is KLR</h1>

    </div>

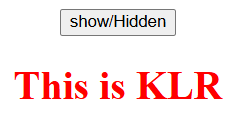
  );

}

export default App;

Output:





Example – 5 (Task) Increasing and decreasing using button:

import './App.css';

import {useState} from "react"

function App() {

  const [count,setCount]=useState(0);

  return (

    <div className="App">

      <button onClick={()=>{setCount(count+1)}}>Increase</button>

      <button onClick={()=>{setCount(count-1)}}>Decrease</button>

      <button onClick={()=>{setCount(0)}}>set to zero</button>

      {count}

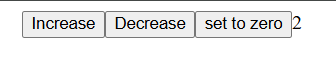
    </div>

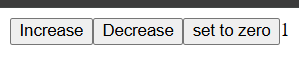
  );

}

export default App;

Output:





CURD in REACT:

TO DO LIST:

import './App.css';

import { useState } from 'react';

function App() {

  const [todoList, setTodoList]=useState([]);

  const [newTask, setNewTask]=useState("");

  const handleChange=(event)=>{

    setNewTask(event.target.value)

  }

  const addTask=()=>{

    const task={

      id: todoList.length === 0 ? 1 : todoList[todoList.length-1].id + 1,

      taskName:newTask,

    }

    setTodoList([...todoList,task])

  }

  const deleteTask=(id)=>{

    setTodoList(todoList.filter((task)=> task.id!== id))

  }

  return (

    <div className="App">

      <div className="addTask">

        <input onChange={handleChange}/>

        <button onClick={addTask}>Add Task</button>

      </div>

      <div className='list'>

        {todoList.map((task)=>{

          return( <div>

                    <h1>{task.taskName}</h1>

                    <button onClick={() => deleteTask(task.id)}>X</button>

                </div>

                );

        })}

      </div>

    </div>

  );

}

export default App;

Fetching Data from API:

import './App.css';

function App() {

  fetch("https://catfact.ninja/fact").then((res)=> res.json()).then((data)=> console.log(data))

  return (

    <div className="App">

      <button>Genarate Cat Fact</button>

      <p></p>

    </div>

  );

}

export default App;

Example – 2 (Using axios)

App.js

import './App.css';

import Axios from 'axios'

import { useState, useEffect } from 'react';

function App() {

  const [catFact, setCatFact] = useState("");

  const fetchData=()=>{

    Axios.get("https://catfact.ninja/fact").then((res)=> setCatFact(res.data.fact))

  }

  useEffect(()=>{

    fetchData();

  },[])

  fetch("https://catfact.ninja/fact").then((res)=> res.json()).then((data)=> console.log(data))

  return (

    <div className="App">

      <button onClick={fetchData}>Genarate Cat Fact</button>

      <p>{catFact}</p>

    </div>

  );

}

export default App;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(

  // <React.StrictMode>

    <App />

  // </React.StrictMode>

);

Example – 3:

API : api.agify.io/?name=yaswanth

// Fetch Data from api axios

import './App.css';

import Axios from 'axios';

import { useState,useEffect } from 'react';

function App() {

  const [name, setName]=useState("");

  const [predictedAge, setPredictedAge] = useState(null);

  const fetchData=()=>{

    Axios.get(`https://api.agify.io/?name=${name}`).then((res)=>setPredictedAge(res.data))

  }

  return (

    <div className="App">

      <input placeholder='Enter name' onChange={(e)=>setName(e.target.value)}/>

      <button onClick={fetchData}>Predict Age</button>

      <h1>Name: {predictedAge?.name}</h1>

      <h1>Age: {predictedAge?.age}</h1>

      <h1>Count: {predictedAge?.count}</h1>

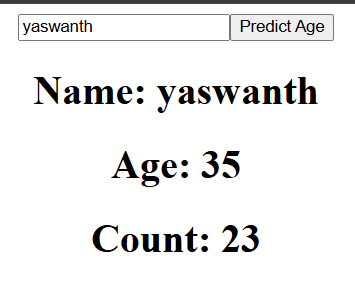
    </div>

  );

}

export default App;

Output:



React-Router-Dom:

Install - > npm install react-router-dom

App.js

import './App.css';

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

import { Home } from './pages/Home';

import { Contact } from './pages/Contact';

import { Menu } from './pages/Menu';

function App() {

  return (

    <div className="App">

      <Router>

        <h1>NavBar</h1>

        <Link to="/">Home</Link>

        <Link to="/menu">Menu</Link>

        <Link to="/contact">Contact</Link>

        <Routes>

          <Route path="/" element={<Home />}/>

          <Route path="/menu" element={<Menu />}/>

          <Route path="/contact" element={<Contact />}/>

          <Route path="\*" element={<h1>Page Not Found</h1>}/>

        </Routes>

      </Router>

    </div>

  );

}

Home.js

export const Home=()=>{

    return <h1>This is Home Page</h1>

}

Menu.js

export const Menu=()=>{

    return <h1>This is Menu Page</h1>

}

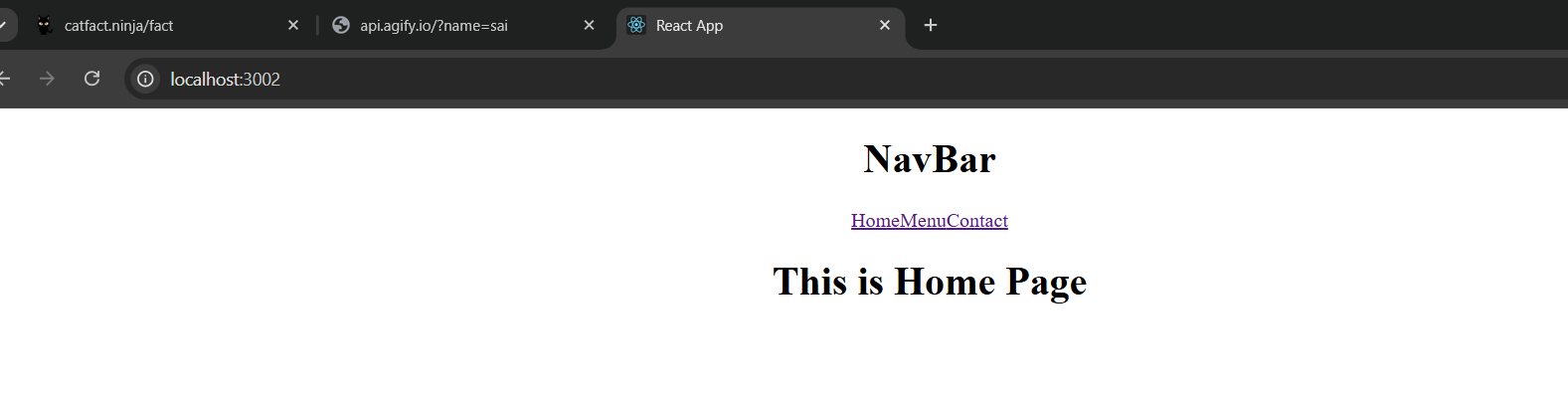
Contact.js

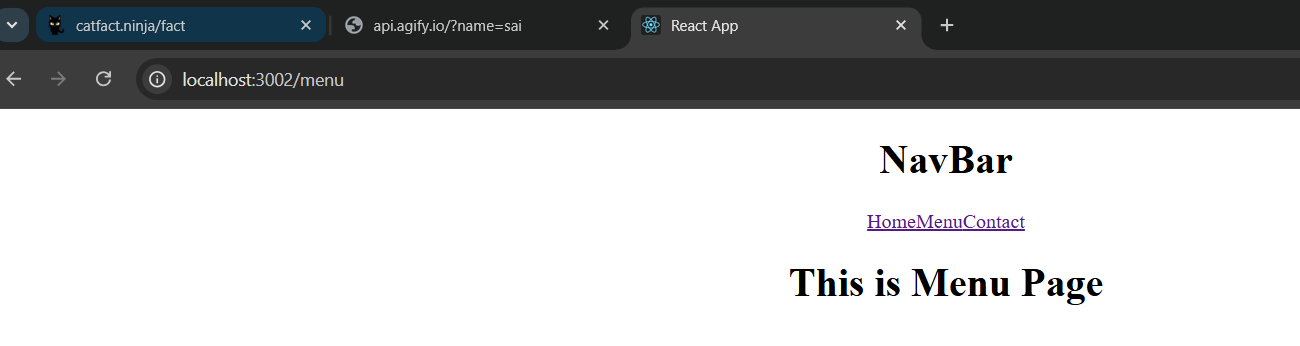
export const Contact=()=>{

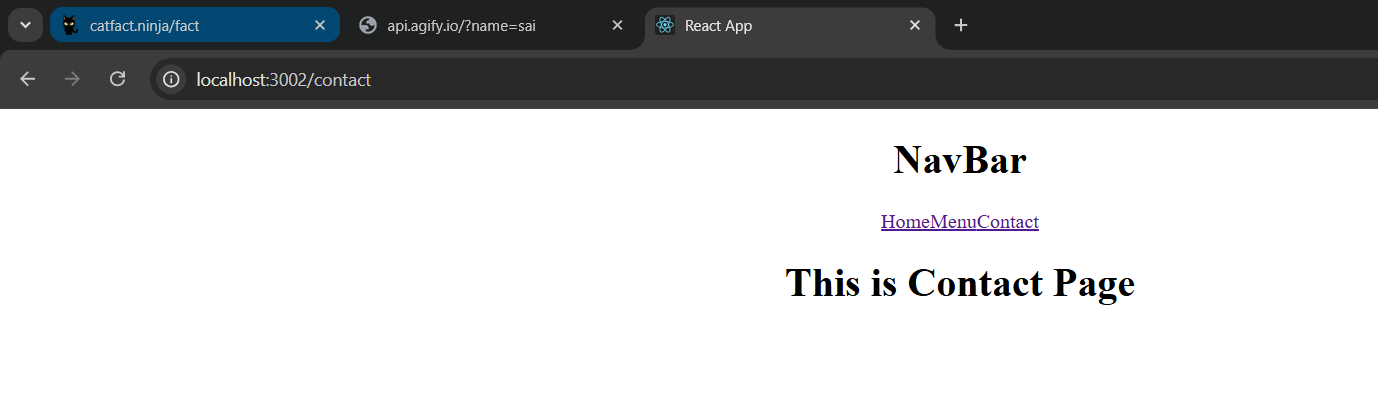
    return <h1>This is Contact Page</h1>

}

Output:







UseContext Hook Example:

App.js

import './App.css';

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

import { Home } from './pages/Home';

import { Profile } from './pages/Profile';

import { Contact } from './pages/Contact';

import { Navbar } from './Navbar';

import { createContext, useState } from 'react';

export const AppContext=createContext();

function App() {

  const [userName,setUserName]=useState("yaswanth");

  return (

    <div className="App">

      <AppContext.Provider value={{userName,setUserName}}>

      <Router>

        <Navbar />

        <Routes>

          <Route path="/" element={<Home />}/>

          <Route path="/profile" element={<Profile />} />

          <Route path="/contact" element={<Contact />} />

        </Routes>

      </Router>

      </AppContext.Provider>

    </div>

  );

}

export default App;

Home.js

import { useContext } from "react"

import { AppContext } from "../App"

export const Home=()=>{

    const {userName}=useContext(AppContext)

    return(

        <h1>This is Home Page . And the user name is {userName}</h1>

    )

}

Contact.js

export const Contact=()=>{

    return(

        <h1>This is Contact Page</h1>

    )

}

Profile.js

import { useContext } from "react"

import { AppContext } from "../App"

import { ChangeProfile } from "../ChangeProfile"

export const Profile=()=>{

    const {userName}=useContext(AppContext)

    return(

        <div>

            <h1>Profile and the user name is {userName}</h1>

            <ChangeProfile />

        </div>

    )

}

ChangeProfile.js

import { AppContext } from "./App";

import { useContext, useState } from "react";

export const ChangeProfile=()=>{

    const [newUserName,setNewUserName]=useState("");

    const {setUserName}=useContext(AppContext)

    return(

        <div>

            <input onChange={(e)=> setNewUserName(e.target.value)}/>

            <button onClick={()=>setUserName(newUserName)}>Change</button>

        </div>

    )

}

Index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(

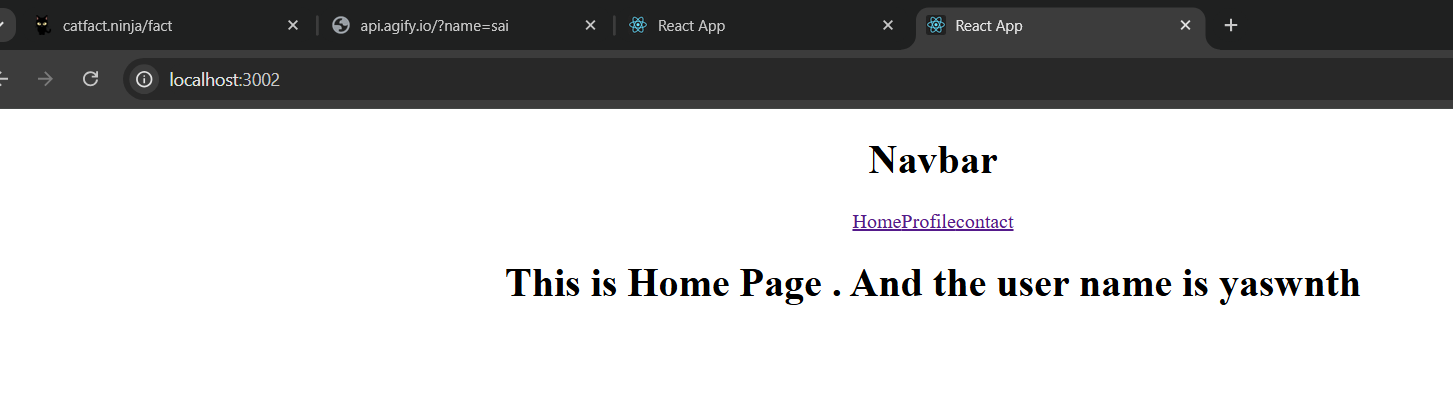
  <React.StrictMode>

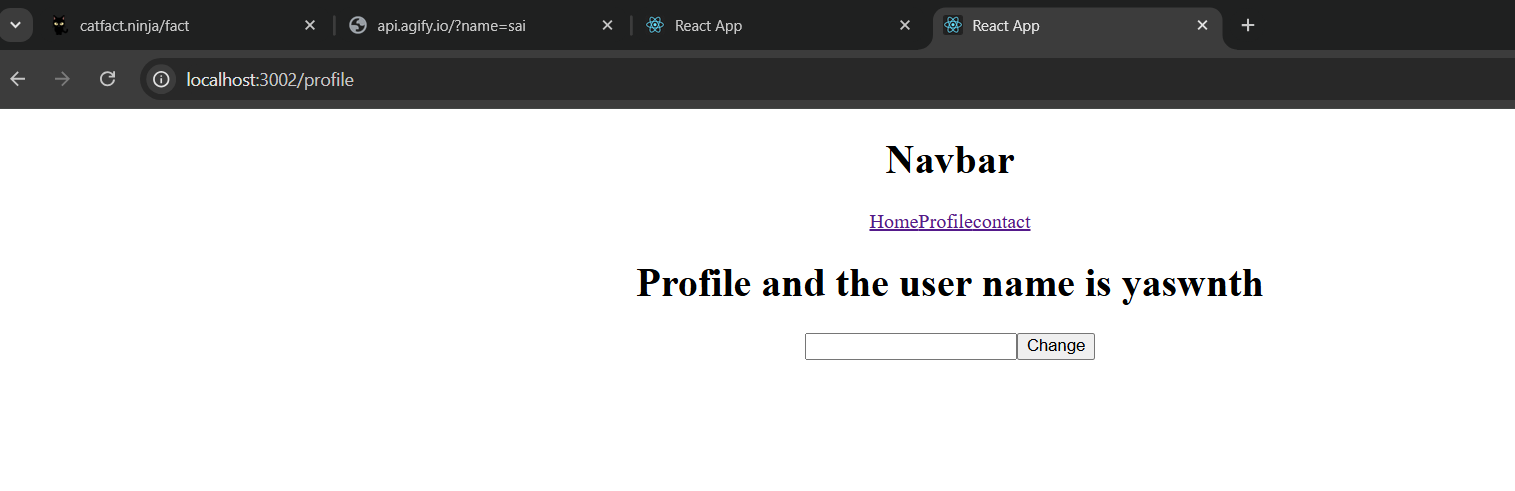
    <App />

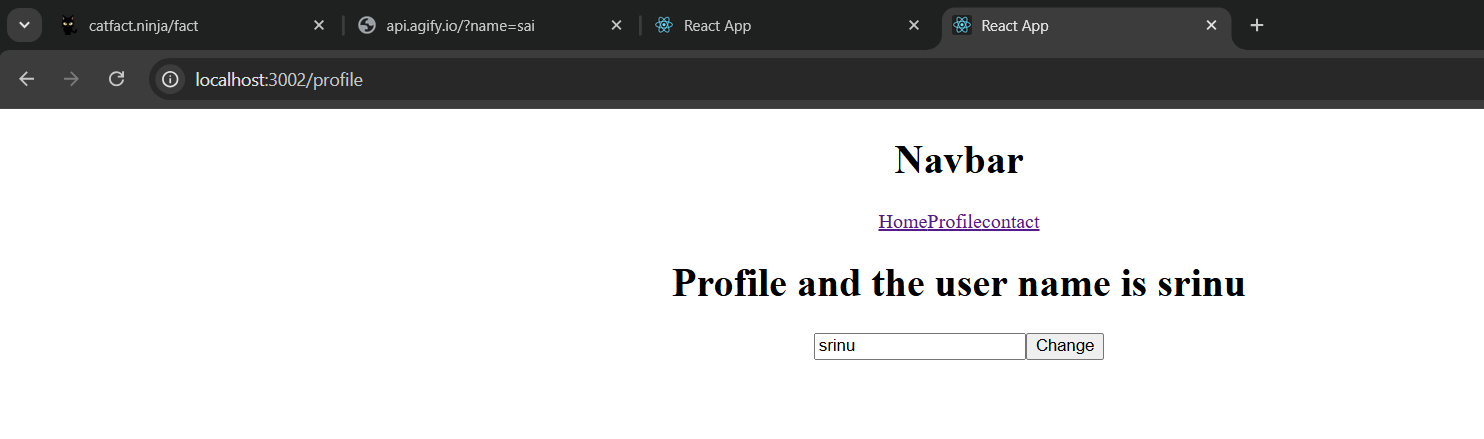
  </React.StrictMode>

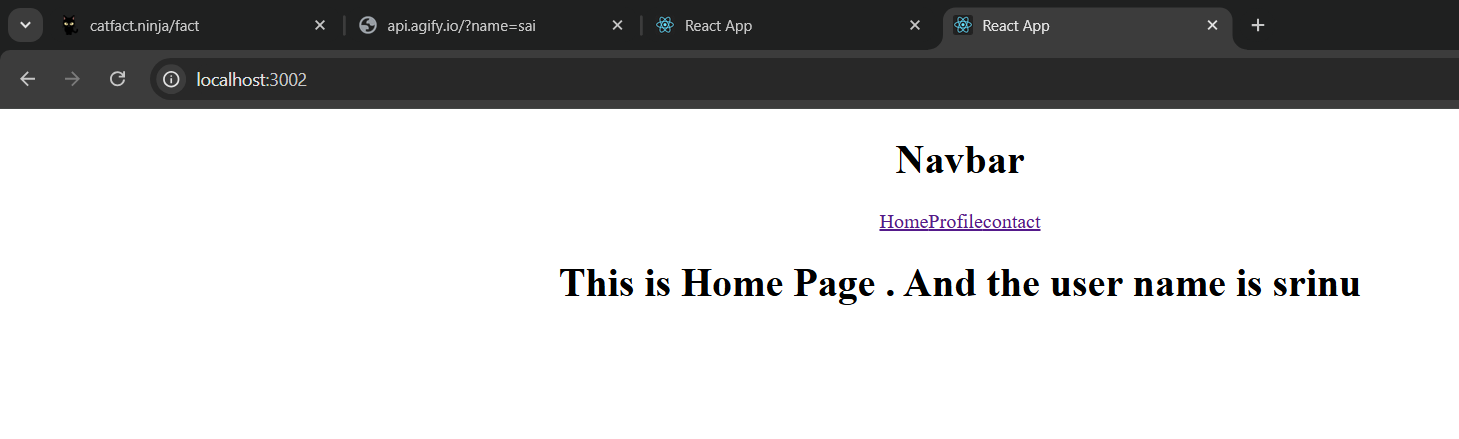
);

Output:









React Query:

Install - > npm install @tanstack/react-query

App.js

import './App.css';

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

import { Home } from './pages/Home';

import { Profile } from './pages/Profile';

import { Contact } from './pages/Contact';

import { Navbar } from './Navbar';

import {QueryClient, QueryClientProvider} from "@tanstack/react-query"

function App() {

  const client=new QueryClient();

  return (

    <div className="App">

      <QueryClientProvider client={client}>

      <Router>

        <Navbar />

        <Routes>

          <Route path="/" element={<Home />}/>

          <Route path="/profile" element={<Profile />}/>

          <Route path="/contact" element={<Contact />}/>

        </Routes>

      </Router>

      </QueryClientProvider>

    </div>

  );

}

export default App;

Navbar.js

import {Link} from 'react-router-dom';

export const Navbar=()=>{

    return(

        <div>

            <h1>Navbar</h1>

            <Link to="/">Home</Link>

            <Link to="/profile">Profile</Link>

            <Link to="/contact">Contact</Link>

        </div>

    )

}

Home.js

import {useQuery} from '@tanstack/react-query';

import Axios from 'axios';

export const Home=()=>{

    const {data, isLoading,isError,refetch} = useQuery(["cat"] , ()=>{

        return Axios.get("https://catfact.ninja/fact").then((res)=>res.data)

    });

    if(isError){

        return <h1>Sorry, There is an error</h1>

    }

    if(isLoading){

        return <h1>Loding.....</h1>

    }

    return(

        <h1>This is Home Page <p>{data.fact}</p>

        <button onClick={refetch}>Update</button>

        </h1>

    )

}

Profile.js

export const Profile=()=>{

    return(

        <h1>This is Profile Page</h1>

    )

}

Context.js

export const Contact=()=>{

    return(

        <h1>This is Contact Page</h1>

    )

}

React forms:

Install Requried Files for this project :

npm install react-hook-form

npm install yup

npm install @hookform/resolvers

App.js

import './App.css';

import { Form } from './components/Form';

function App() {

  return (

    <div className="App">

      <Form />

    </div>

  );

}

export default App;

Form.js

import {useForm} from 'react-hook-form';

import \* as yup from "yup";

import {yupResolver} from '@hookform/resolvers/yup'

export const Form=()=>{

const scheme=yup.object().shape({

        fullName: yup.string().required("This is required Field"),

        email:yup.string().email().required(),

        age:yup.number().positive().integer().min(5).required(),

        password:yup.string().min(4).max(20).required(),

        confirmPassword:yup.string().oneOf([yup.ref("password"),null],"password Don't Match").required(),

    })

    const {register, handleSubmit, formState : {errors}} = useForm({

        resolver: yupResolver(scheme),

    });

    const onSubmit=(data)=>{

        console.log(data);

    }

    return(

        <form onSubmit={handleSubmit(onSubmit)}>

            <input type="text" placeholder="Full Name.." {...register("fullName")}/>

            <p>{errors.fullName?.message}</p>

            <input type="text" placeholder="Email..." {...register("email")}/>

            <p>{errors.email?.message}</p>

            <input type="number" placeholder="Age.." {...register("age")}/>

            <p>{errors.age?.message}</p>

            <input type="password" placeholder="Password.." {...register("password")}/>

            <p>{errors.password?.message}</p>

            <input type="password" placeholder="confirm password.." {...register("confirmPassword")}/>

            <p>{errors.confirmPassword?.message}</p>

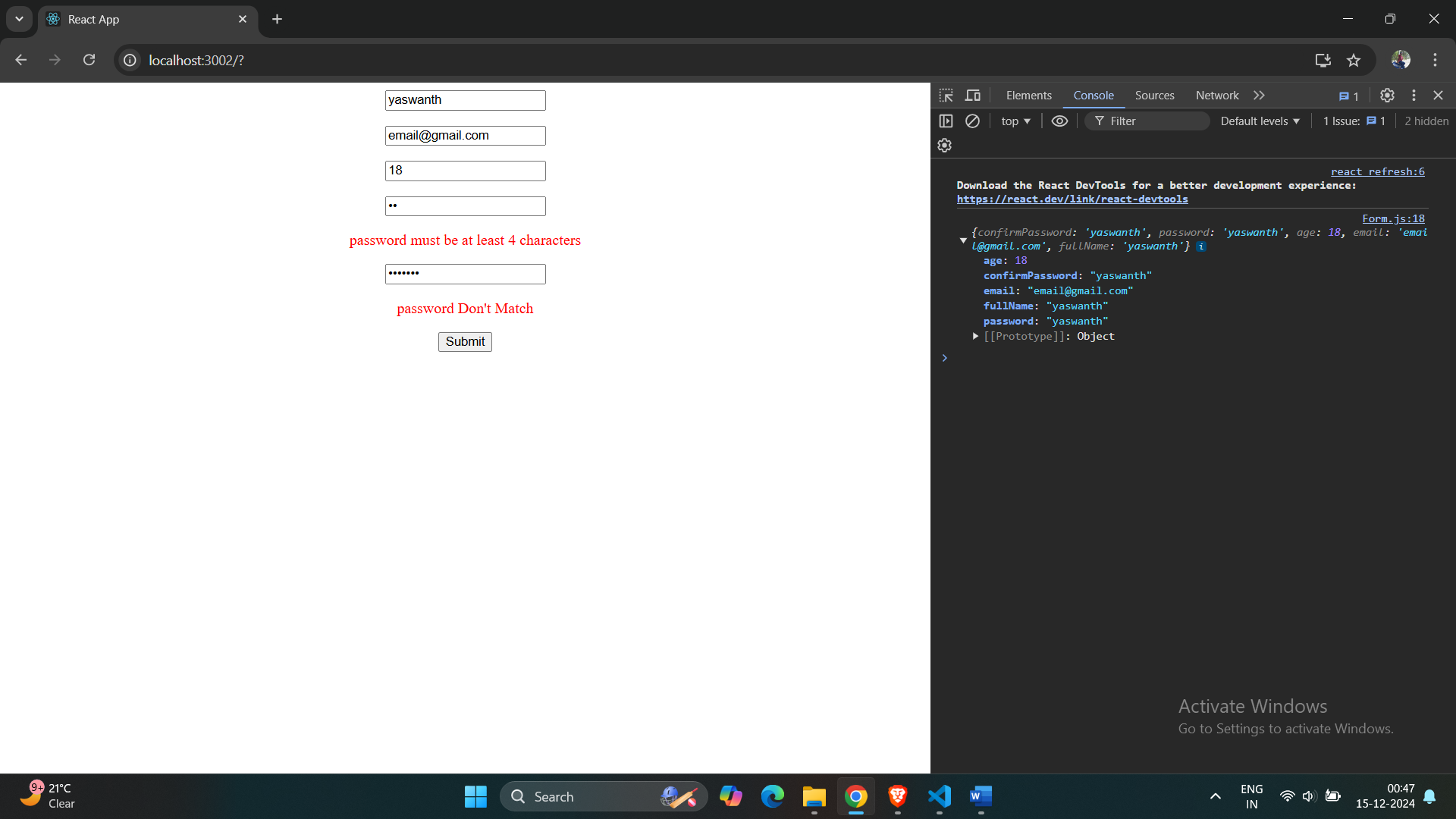
            <input type="submit" />

        </form>

    );

};

Output:



Custom Hooks :

App.js

import { useState } from 'react';

import './App.css';

import { useToggle } from './useToggle';

import { useCount } from './useCount';

function App() {

  const [isVisible1, toggle1] =useToggle();

  const {state, toggle} = useToggle(); // this is object type . so custoom hook also return the object {} => In this it is not worked

  const {count,increase,decrease,restart}=useCount();

  return (

    <div className="App">

      <button onClick={toggle1}>

        {isVisible1 ? "hide" : "show"}

      </button>

    {isVisible1 && <h1>This is hidden Text</h1>}

    <button onClick={toggle}>

        {state ? "hide" : "show"}

      </button>

    {state && <h1>This is hidden Text</h1>}

    {/\* Custom Hook increase, Decrease and restrat \*/}

    <div>

        <h1>{count}</h1>

        <button onClick={increase}>Increase</button>

        <button onClick={decrease}>Decrease</button>

        <button onClick={restart}>Restart</button>

    </div>

    </div>

  );

}

export default App;

useCount.js

import { useState } from "react"

export const useCount=(initial = 0)=>{

    const [count, setCount] = useState(initial);

    const increase = ()=>{

        setCount((prev)=>prev+1)

    }

    const decrease = ()=>{

        setCount((prev) => prev-1);

    }

    const restart=()=>{

        setCount(0);

    }

    return {count, increase, decrease, restart};

}

useToggle.js:

import { useState } from "react"

export const useToggle=(initial = false)=>{

    const [state, setState] = useState(initial);

    const toggle=()=>{

        setState((prev)=>!prev);

    };

    return [state,toggle]

}

Index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(

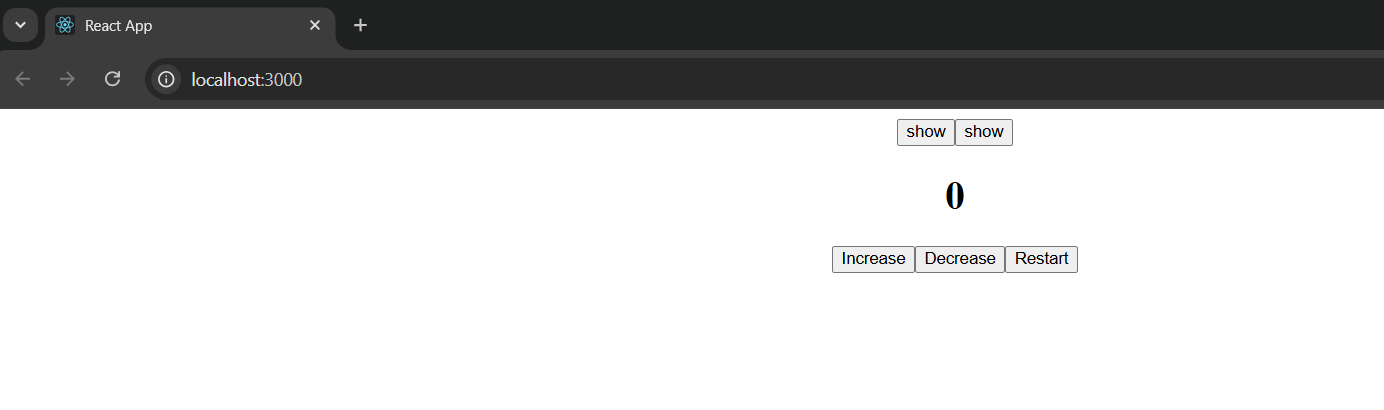
  <React.StrictMode>

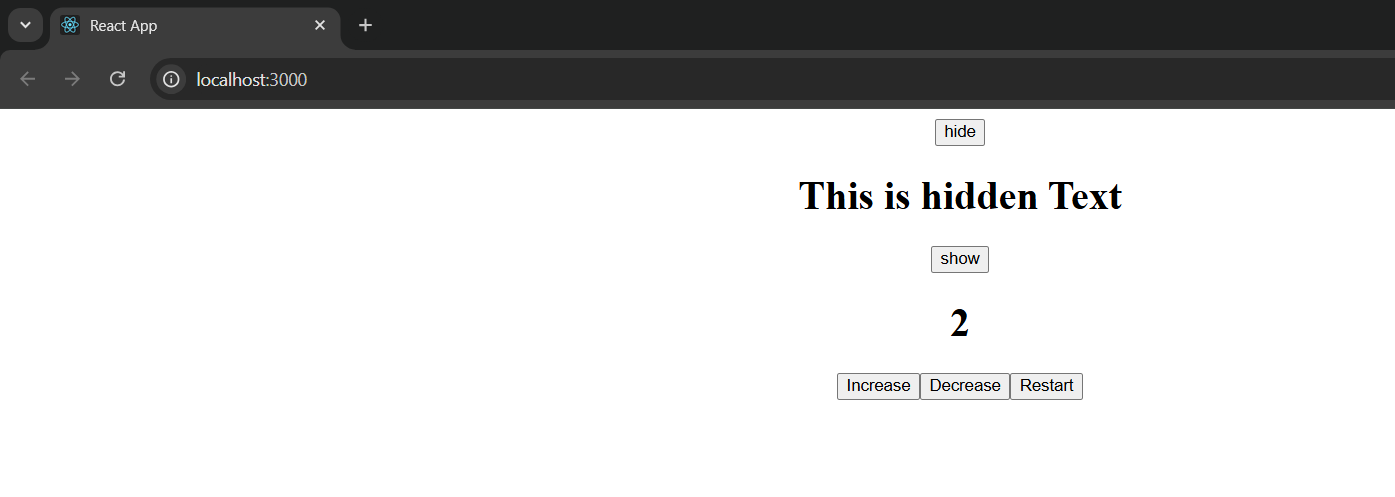
    <App />

  </React.StrictMode>

);

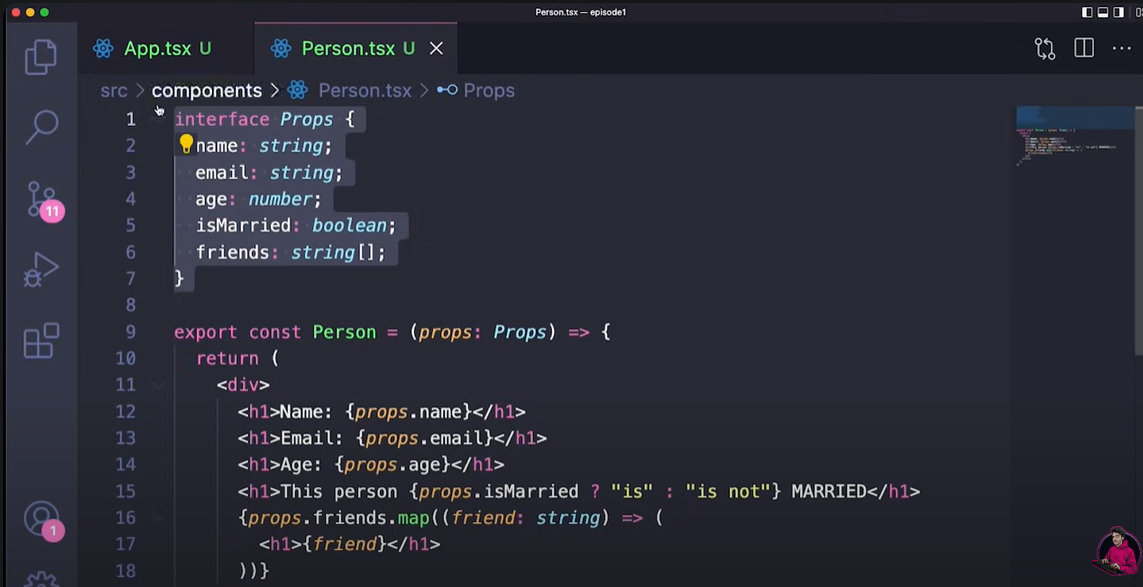
Output:

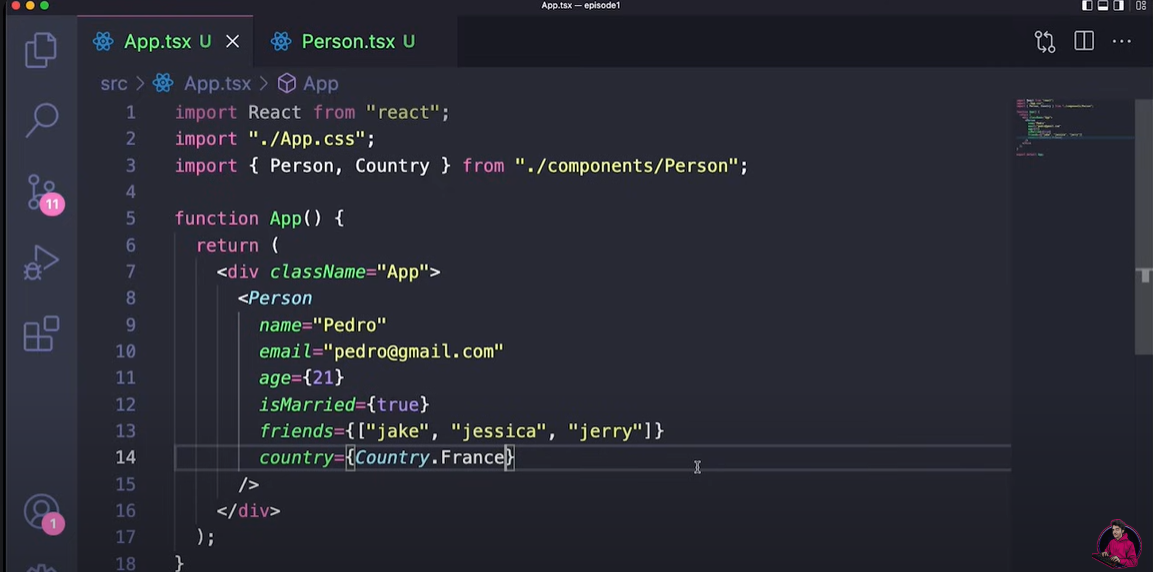


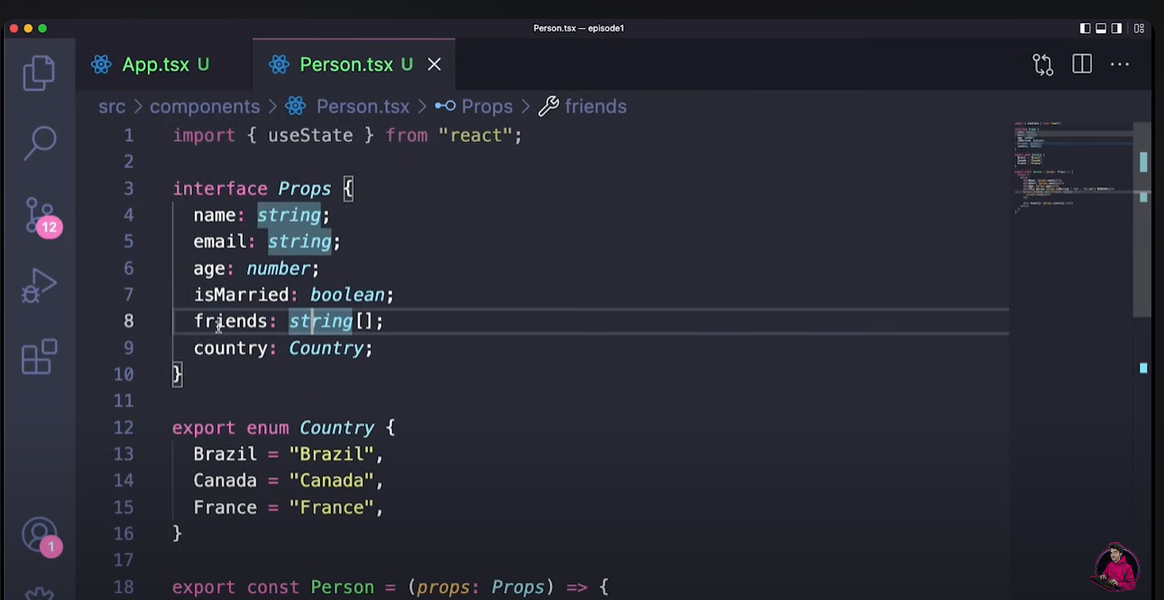


TypeScript :

npx create-react-app . –template typescript







Redux – Tool in React :

npx create-react-app . –template typescript

npm install @reduxjs/toolkit

npm install react-redux

App.tsx:

import './App.css';

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

import { Home } from './pages/Home';

import { Login } from './pages/Login';

import { Contact } from './pages/Contact';

import {Provider} from 'react-redux';

import {store} from './store'

function App() {

  return (

    <div className="App">

      <Provider store={store}>

      <Router>

        <Link to="/">Home</Link>

        <Link to="/login">Login</Link>

        <Link to="/contact">Contact</Link>

        <Routes>

          <Route path="/" element={<Home />}/>

          <Route path="/login" element={<Login />} />

          <Route path="/contact" element={<Contact />} />

        </Routes>

      </Router>

      </Provider>

    </div>

  );

}

export default App;

store.ts

import {configureStore, createSlice} from "@reduxjs/toolkit";

const initialState={value: {userName: ""}};

const userSlice = createSlice({

    name: "user",

    initialState,

    reducers :{

        login: (state,action) =>{

            state.value=action.payload;

        },

        logout: (state) =>{

            state.value=initialState.value;

        }

    }

})

export const {login,logout} = userSlice.actions;

export const store=configureStore({

    reducer:{

        user:userSlice.reducer

    }

})

Home.tsx:

import {useDispatch, useSelector} from 'react-redux'

export const Home=()=>{

    const username=useSelector((state: any) => state.user.value.username);

    return(

        <h1>This is Home Page {username}</h1>

    )

}

Login.tsx:

import { useState } from "react"

import {login,logout} from '../store';

import {useDispatch, useSelector} from 'react-redux'

export const Login=()=>{

    const [newUserName, setNewUserName] = useState("");

    const dispatch=useDispatch();

    const username=useSelector((state: any) => state.user.value.username);

    return(

        <div>

            {/\* <h1>This is Login Page</h1> \*/}

            {username}

            <input onChange={(e)=>setNewUserName(e.target.value)}/>

            <button onClick={()=> dispatch(login({username: newUserName}))}>Login</button>

            <button onClick={()=>dispatch(logout())}>Logout</button>

        </div>

    )

}

Contact.tsx:

export const Contact=()=>{

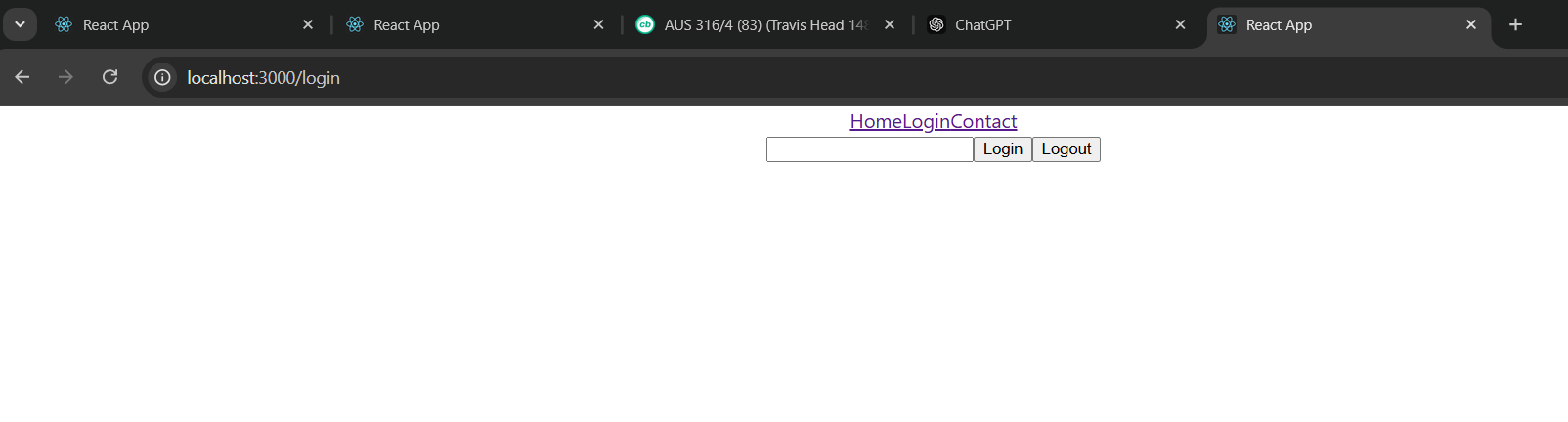
    return(

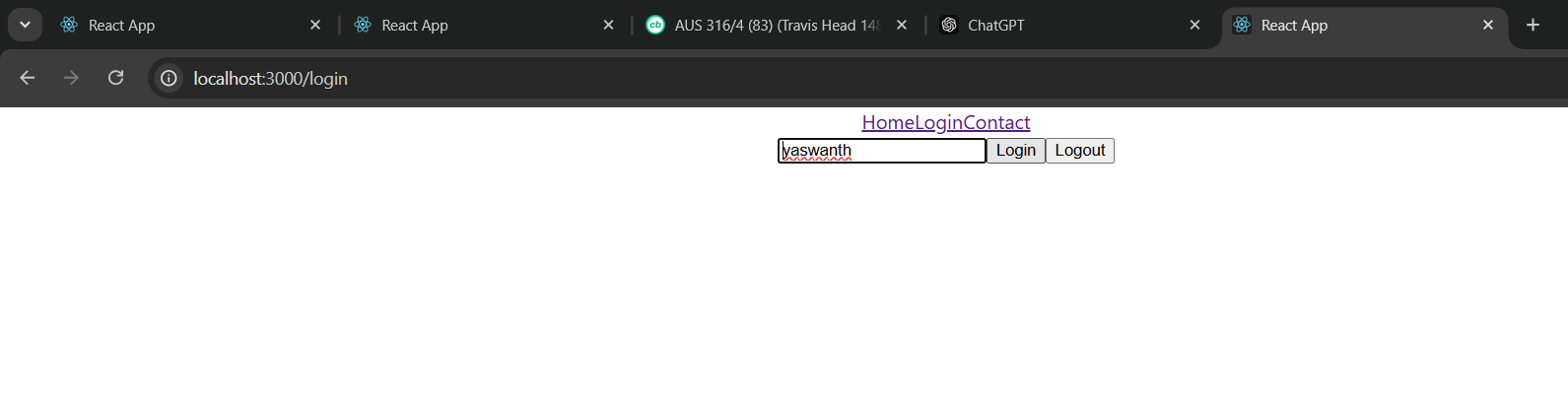
        <h1>This is Contact Page</h1>

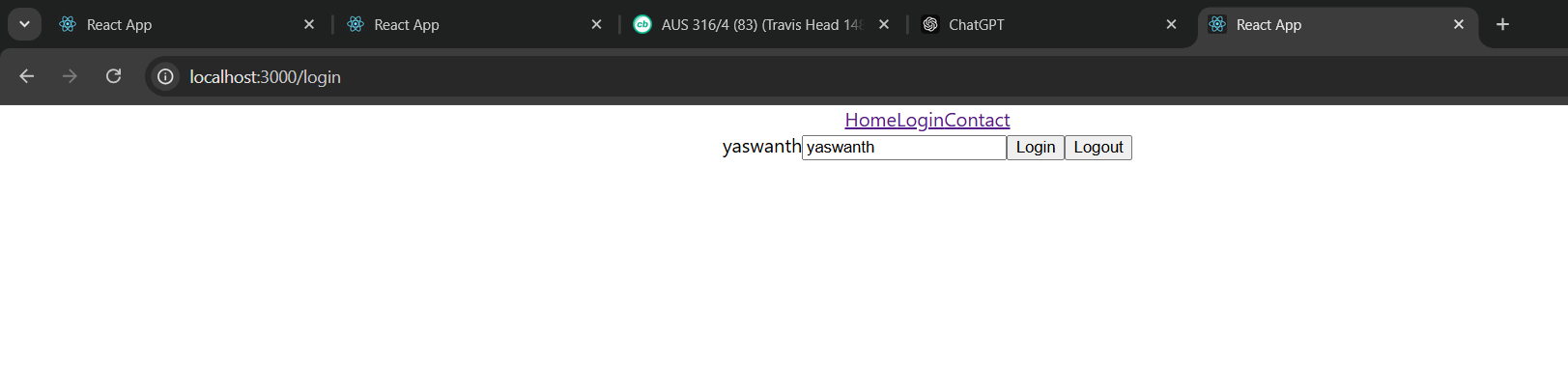
    )

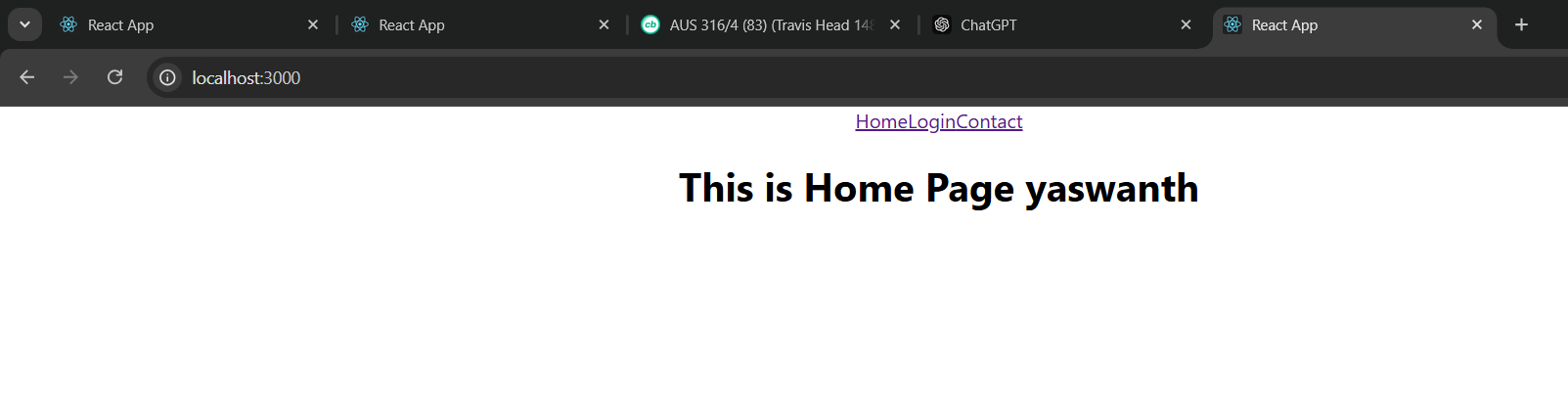
}

Output:

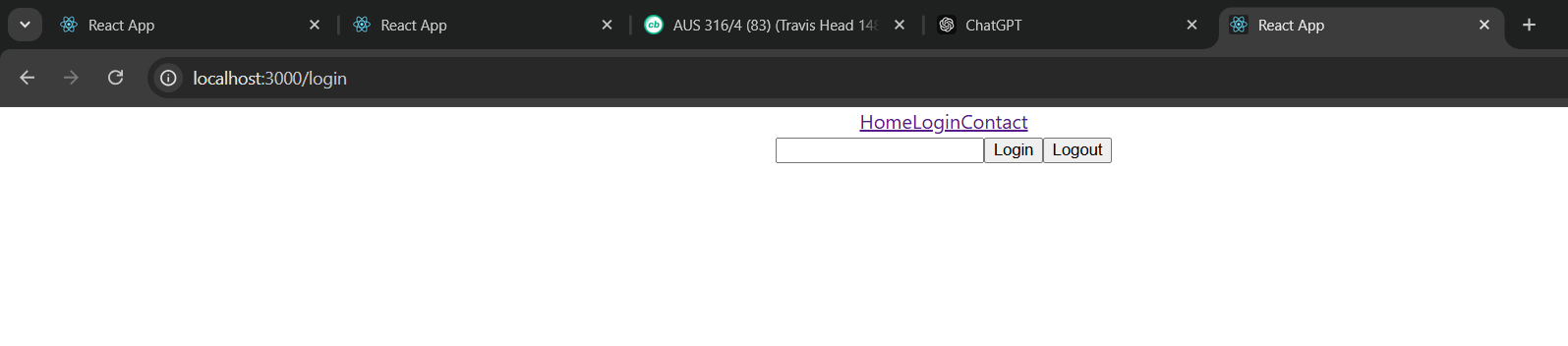








Going back to login page and click logout:



Social- Media – app using react js:

Login Authentication:

App.tsx:

import './App.css';

import React from 'react';

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

import { Main } from './pages/main';

import { Login } from './pages/login';

import { Navbar } from './components/navbar';

function App() {

  return (

    <div className="App">

      <Router>

        <Navbar />

        <Routes>

          <Route path="/" element={<Main />}/>

          <Route path="/login" element={<Login />} />

          <Route path="\*" element={"Page Not Found"}/>

        </Routes>

      </Router>

    </div>

  );

}

export default App;

firebase.ts:

// Import the functions you need from the SDKs you need

import { initializeApp } from "firebase/app";

import {getAuth, GoogleAuthProvider} from 'firebase/auth'

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration

const firebaseConfig = {

  apiKey: "AIzaSyCU73e-DABUEczcGK0BRUktYLuuIFCM1vY",

  authDomain: "social-media-react-8ad64.firebaseapp.com",

  projectId: "social-media-react-8ad64",

  storageBucket: "social-media-react-8ad64.firebasestorage.app",

  messagingSenderId: "413837802202",

  appId: "1:413837802202:web:99e354994cdd8f8a1043e3"

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

export const auth=getAuth(app);

export const provider=new GoogleAuthProvider();

navbar.tsx:

import { Link } from "react-router-dom";

import { auth } from '../config/firebase';

import { useAuthState } from "react-firebase-hooks/auth";

import {signOut} from "firebase/auth"

export const Navbar = () => {

    const [user] = useAuthState(auth);

    const signUserOut=async()=>{

        await signOut(auth);

    }

    return (

        <div className="navbar">

            <h2>NavBar</h2>

            <Link to="/">Home</Link>

            <Link to="/login">Login</Link>

            <div className="user-info">

                {user && (

                    <>

                <p>{user?.displayName}</p>

                <img src={user?.photoURL || ""} alt="User" />

                <button onClick={signUserOut}>Logout</button>

                </>

                )

            }

            </div>

        </div>

    );

};

Login.tsx:

import {auth,provider} from '../config/firebase';

import {signInWithPopup} from 'firebase/auth';

import {useNavigate} from 'react-router-dom';

export const Login=()=>{

    const navigate=useNavigate();

    const signInWithGoogle=async ()=>{

        const result= await signInWithPopup(auth,provider);

        console.log(result);

        navigate("/");

    }

    return(

        <div>

        <h1>Login Page</h1>

        <p>Sign in With Google TO continue</p>

        <button onClick={signInWithGoogle}>Sign in with Google</button>

        </div>

    )

}

main.tsx:

import {auth} from "../config/firebase";

import {useAuthState} from "react-firebase-hooks/auth"

export const Main=()=>{

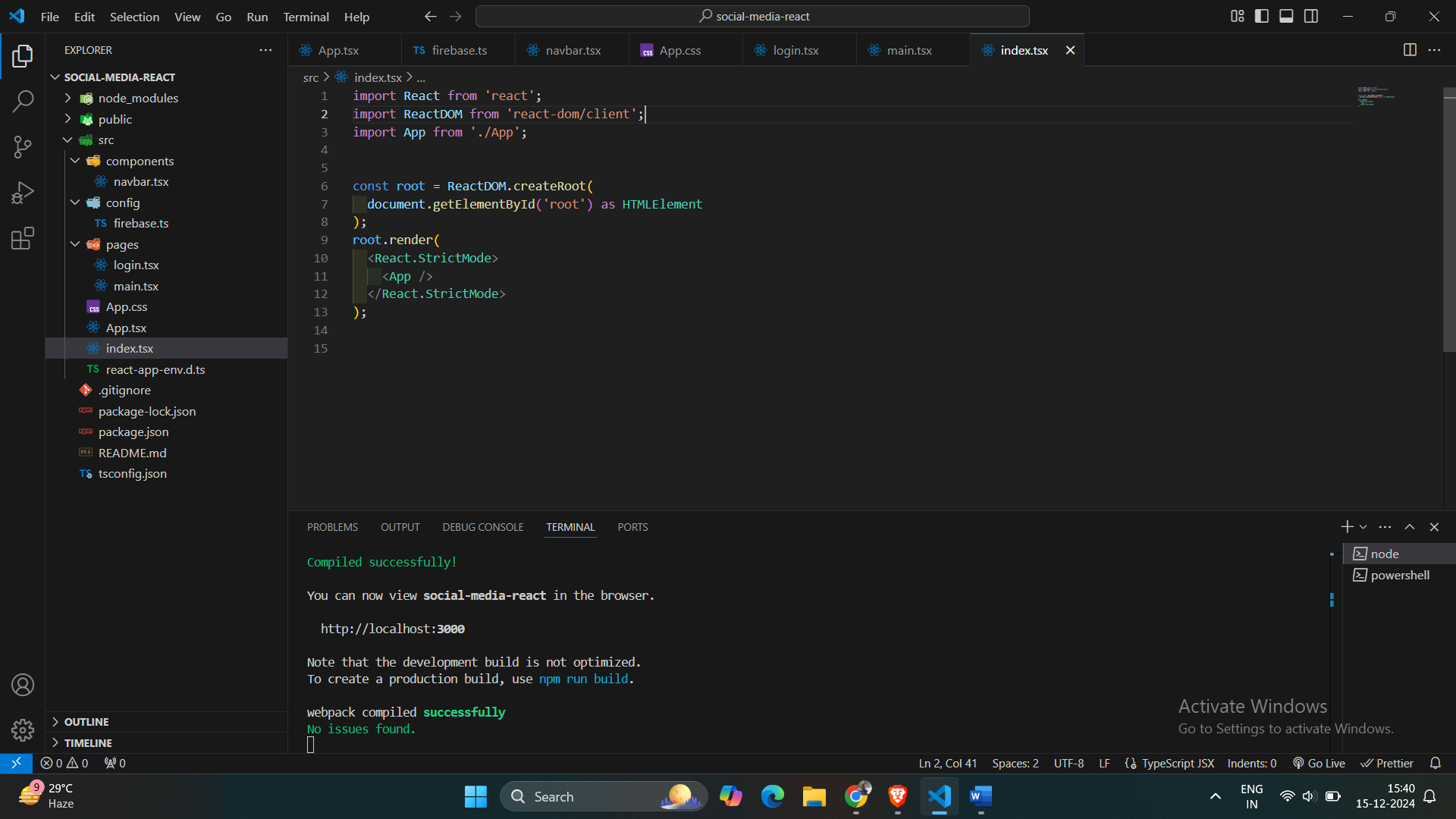
    const [user]=useAuthState(auth);

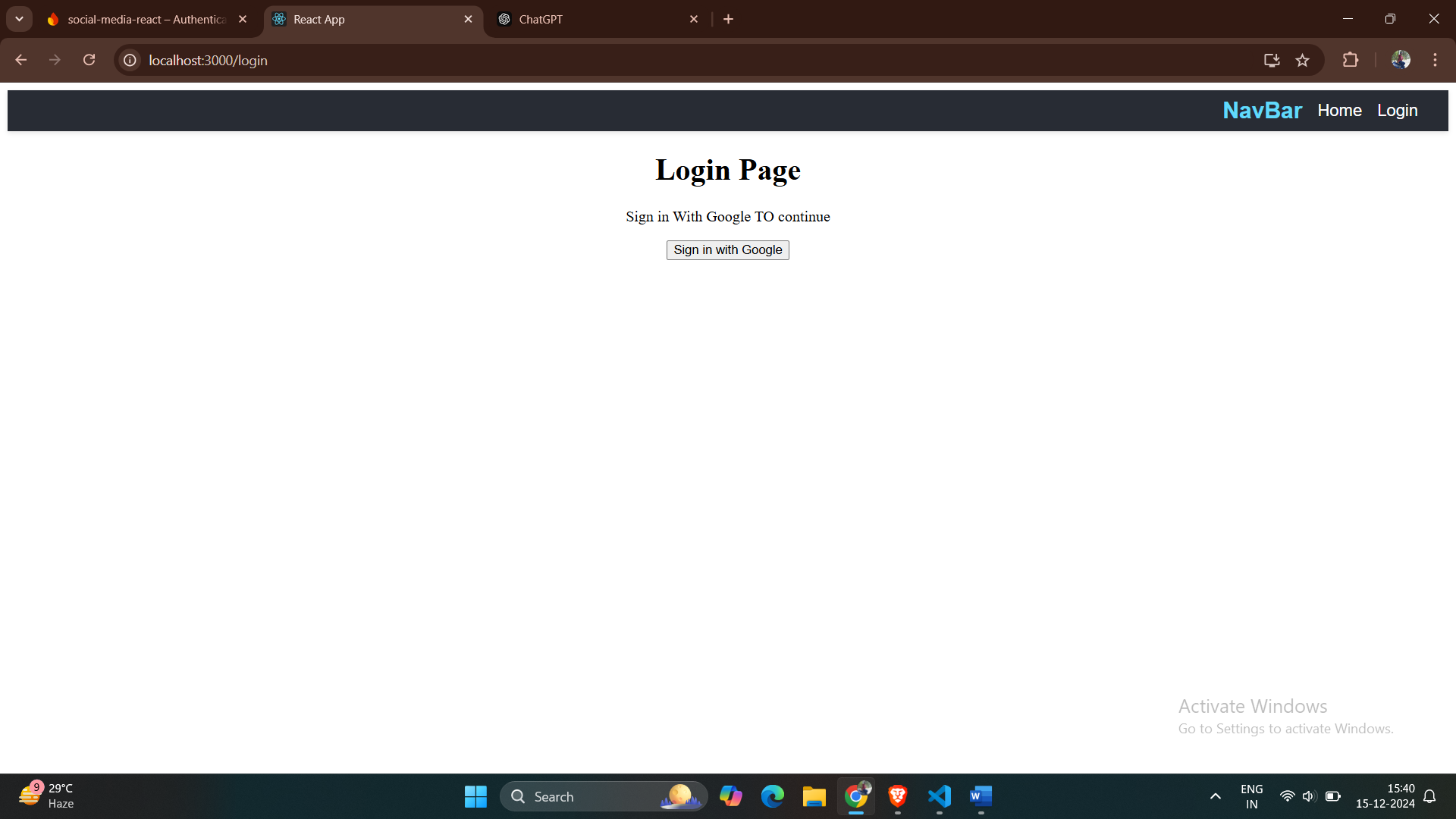
    return(

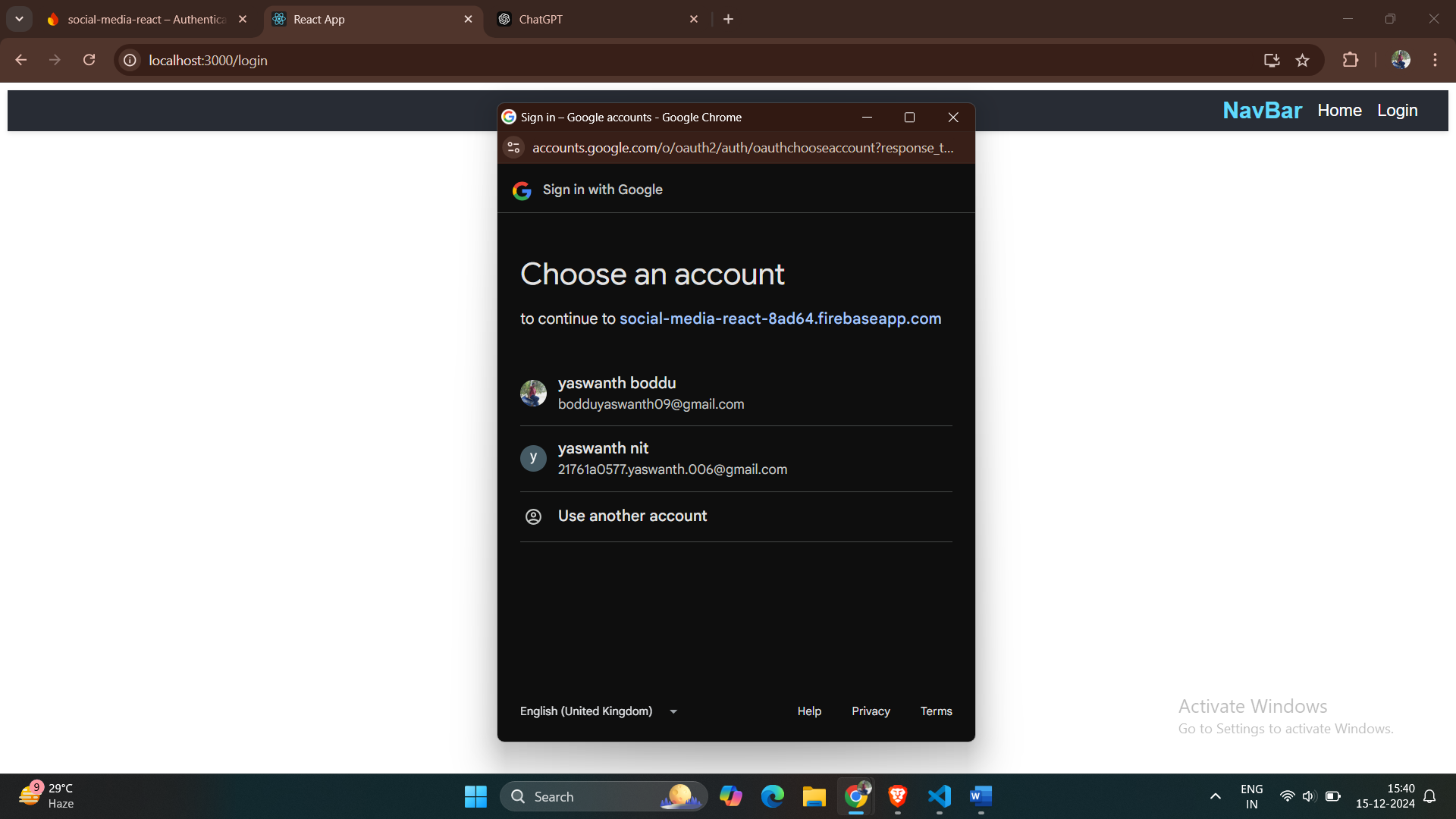
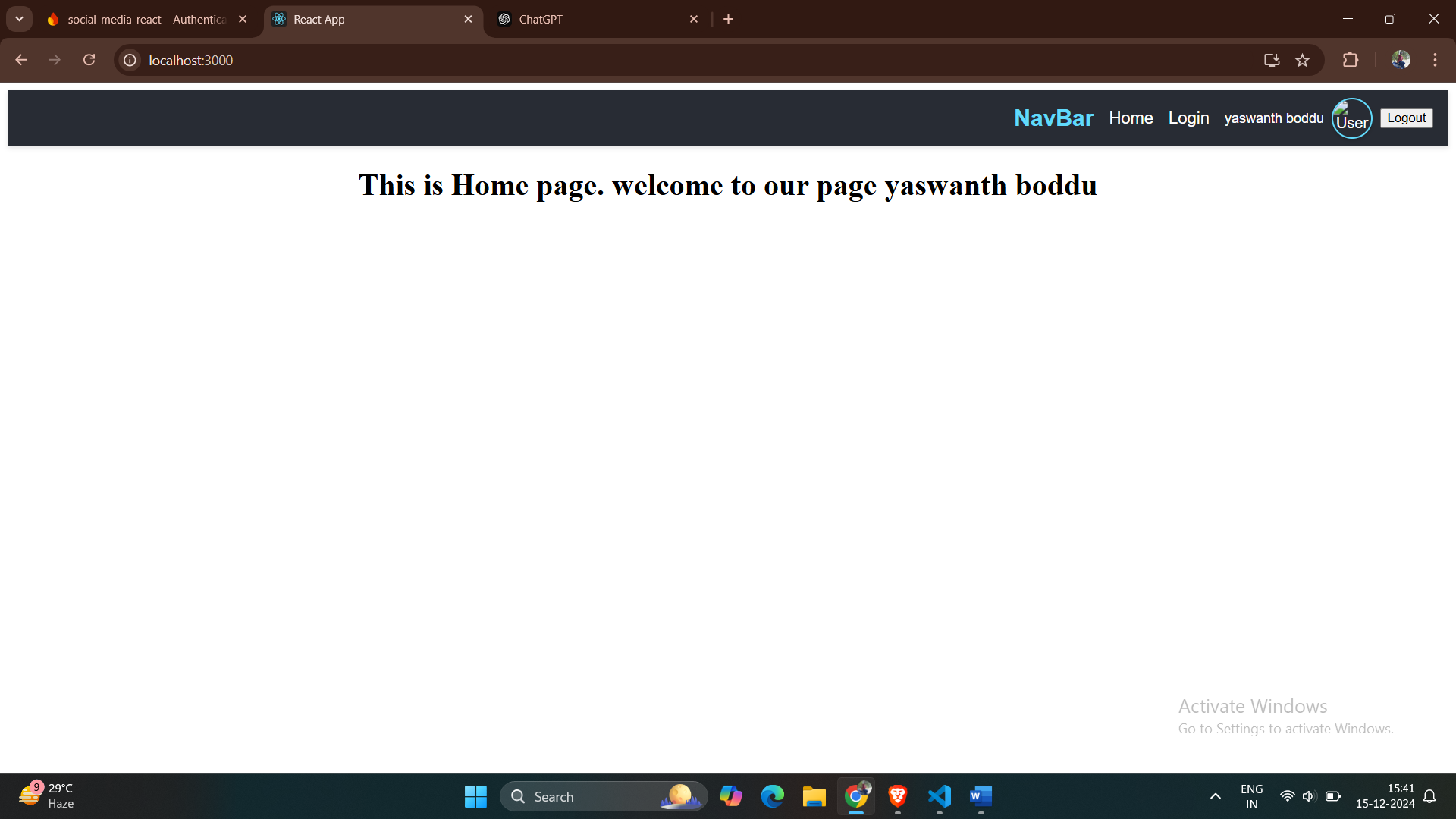
        <h1>This is Home page. welcome to our page {user?.displayName}</h1>

    )

}





Post Data From the ReactApp to firebase.google.com

App.tsx:

import './App.css';

import React from 'react';

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

import { Main } from './pages/main';

import { Login } from './pages/login';

import { Navbar } from './components/navbar';

import { CreatePost } from './pages/create-post/create-post';

function App() {

  return (

    <div className="App">

      <Router>

        <Navbar />

        <Routes>

          <Route path="/" element={<Main />}/>

          <Route path="/login" element={<Login />} />

          <Route path="/createpost" element={<CreatePost />} />

          <Route path="\*" element={"Page Not Found"}/>

        </Routes>

      </Router>

    </div>

  );

}

export default App;

create-post.tsx:

import { CreateForm } from "./create-form"

export const CreatePost=()=>{

    return(

        <div>

        <CreateForm />

        </div>

    )

}

create-form.tsx:

import {useForm} from 'react-hook-form';

import \* as yup from "yup";

import {yupResolver} from '@hookform/resolvers/yup';

import { addDoc, collection } from 'firebase/firestore';

import { auth, db } from '../../config/firebase';

import { title } from 'process';

import { useAuthState } from 'react-firebase-hooks/auth';

import { useNavigate } from 'react-router-dom';

interface CreateFormData{

    title: string,

    description: string

}

export const CreateForm=()=>{

    const [user] =useAuthState(auth);

    const navigate=useNavigate();

    const schema=yup.object().shape({

        title: yup.string().required("You must add the title"),

        description: yup.string().required("You must add the description"),

    })

    const {register, handleSubmit, formState:{errors}}=useForm<CreateFormData>({

        resolver:yupResolver(schema),

    })

    const postRef = collection(db,"posts")

    const onCreatePost=async (data:CreateFormData)=>{

        await addDoc(postRef,{

            title: data.title,

            description: data.description, // ...data

            username: user?.displayName,

            userId: user?.uid

        })

        navigate("/")

    }

    return(

        <form onSubmit={handleSubmit(onCreatePost)}>

            <input placeholder='title...' {...register("title")}/>

            <p style={{color:"red"}}>{errors.title?.message}</p>

            <textarea placeholder='Description...' {...register("description")}/>

            <p style={{color:"red"}}> {errors.description?.message}</p>

            <input type='submit' />

        </form>

    )

}

firebase.ts:

// Import the functions you need from the SDKs you need

import { initializeApp } from "firebase/app";

import {getAuth, GoogleAuthProvider} from 'firebase/auth'

import {getFirestore} from 'firebase/firestore'

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration

const firebaseConfig = {

  apiKey: "AIzaSyCU73e-DABUEczcGK0BRUktYLuuIFCM1vY",

  authDomain: "social-media-react-8ad64.firebaseapp.com",

  projectId: "social-media-react-8ad64",

  storageBucket: "social-media-react-8ad64.firebasestorage.app",

  messagingSenderId: "413837802202",

  appId: "1:413837802202:web:99e354994cdd8f8a1043e3"

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

export const auth=getAuth(app);

export const provider=new GoogleAuthProvider();

export const db=getFirestore(app)

navbar.tsx:

import { Link } from "react-router-dom";

import { auth } from '../config/firebase';

import { useAuthState } from "react-firebase-hooks/auth";

import {signOut} from "firebase/auth"

import { useNavigate } from "react-router-dom";

export const Navbar = () => {

    const [user] = useAuthState(auth);

    const navigate=useNavigate();

    const signUserOut=async()=>{

        await signOut(auth);

        navigate("/");

    }

    return (

        <div className="navbar">

            <h2>NavBar</h2>

            <Link to="/">Home</Link>

            {!user ? ( <Link to="/login">Login</Link> ): ( <Link to="/createpost">Create Post</Link>) }

            <div className="user-info">

                {user && (

                    <>

                <p>{user?.displayName}</p>

                <img src={user?.photoURL || ""} alt="User" />

                <button onClick={signUserOut}>Logout</button>

                </>

                )

            }

            </div>

        </div>

    );

};

login.tsx:

import {auth,provider} from '../config/firebase';

import {signInWithPopup} from 'firebase/auth';

import {useNavigate} from 'react-router-dom';

export const Login=()=>{

    const navigate=useNavigate();

    const signInWithGoogle=async ()=>{

        const result= await signInWithPopup(auth,provider);

        console.log(result);

        navigate("/");

    }

    return(

        <div>

        <h1>Login Page</h1>

        <p>Sign in With Google TO continue</p>

        <button onClick={signInWithGoogle}>Sign in with Google</button>

        </div>

    )

}

main.tsx:

import {auth} from "../config/firebase";

import {useAuthState} from "react-firebase-hooks/auth"

export const Main=()=>{

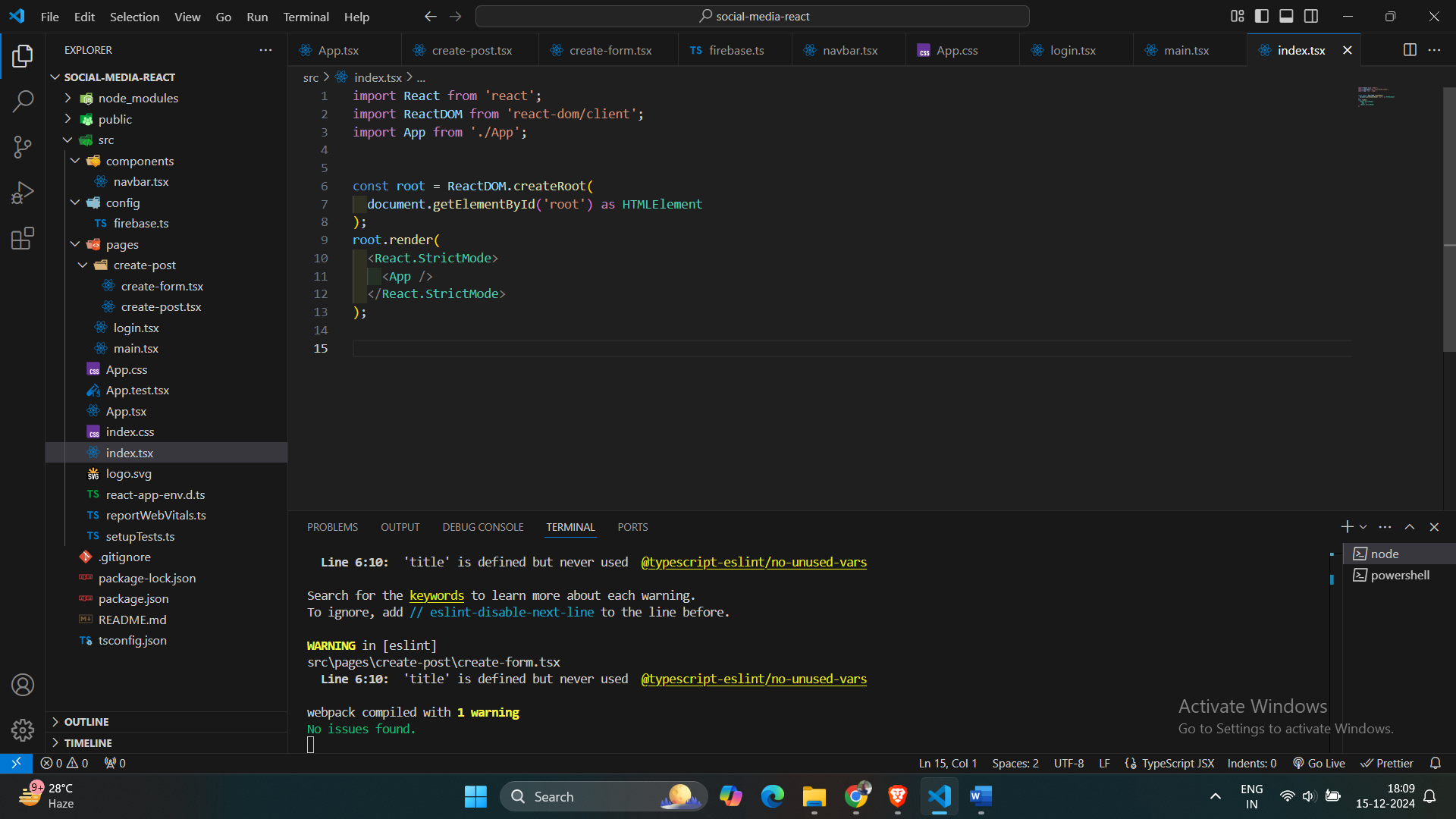
    const [user]=useAuthState(auth);

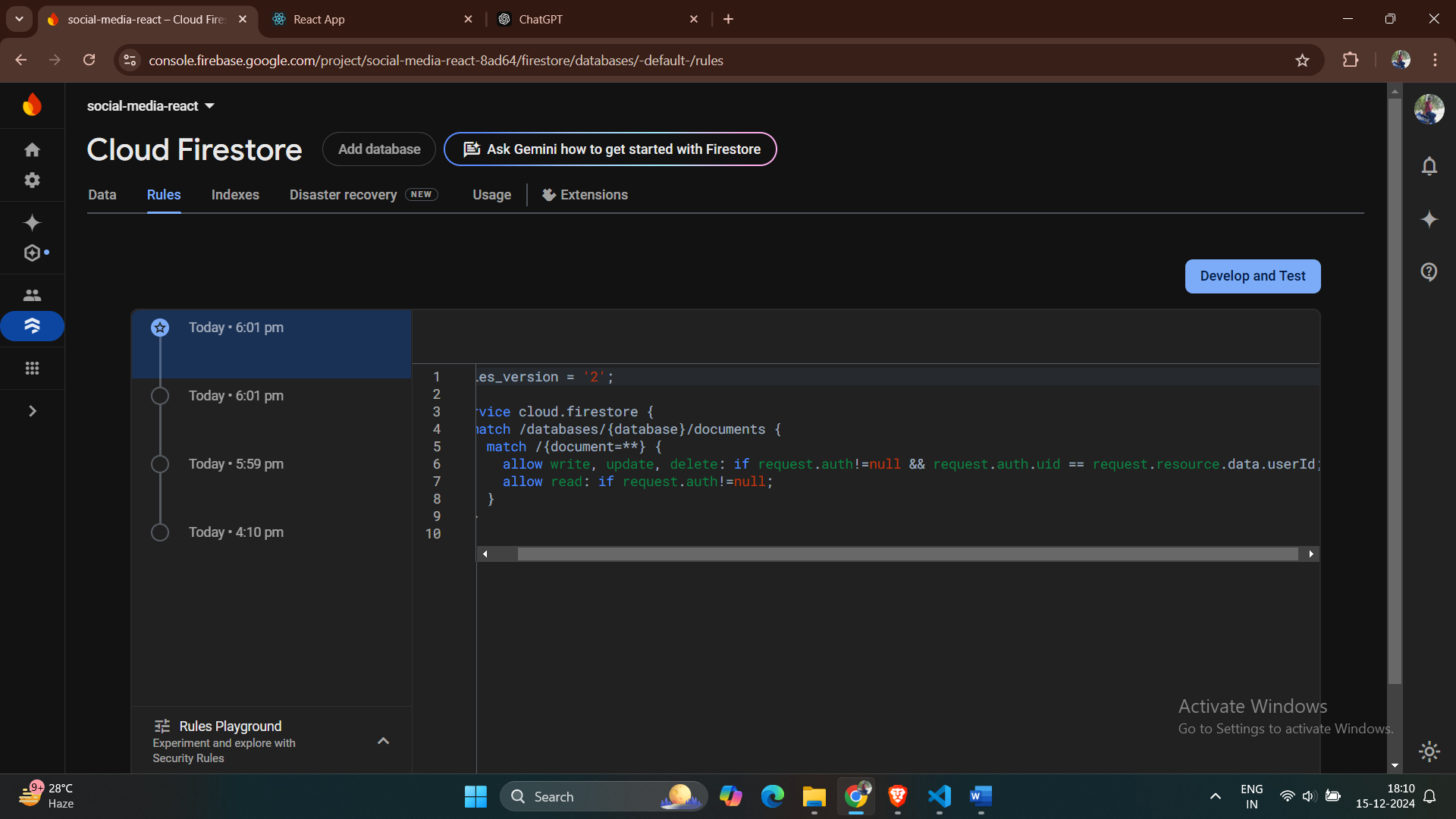
    return(

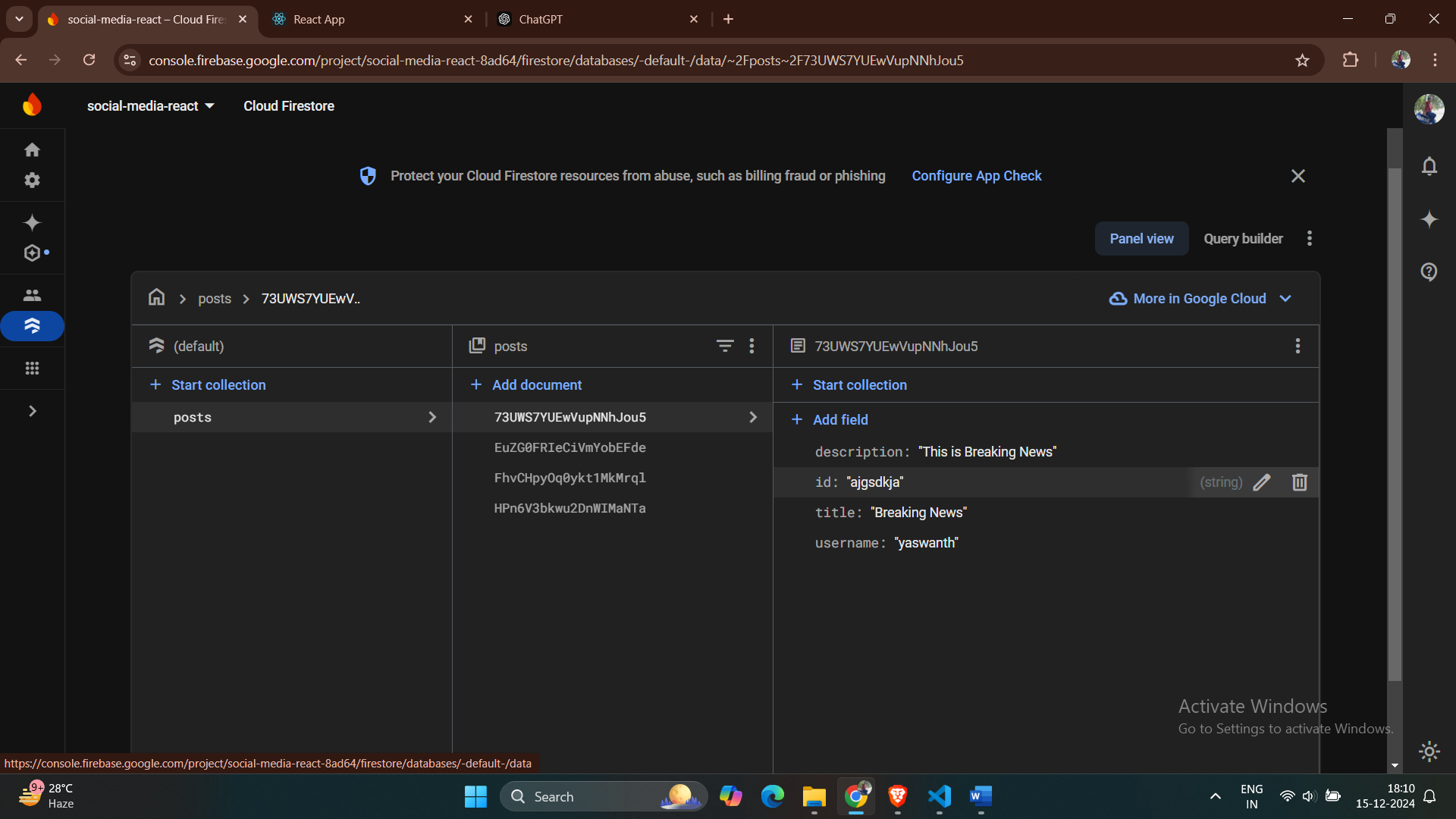
        <h1>This is Home page. welcome to our page {user?.displayName}</h1>

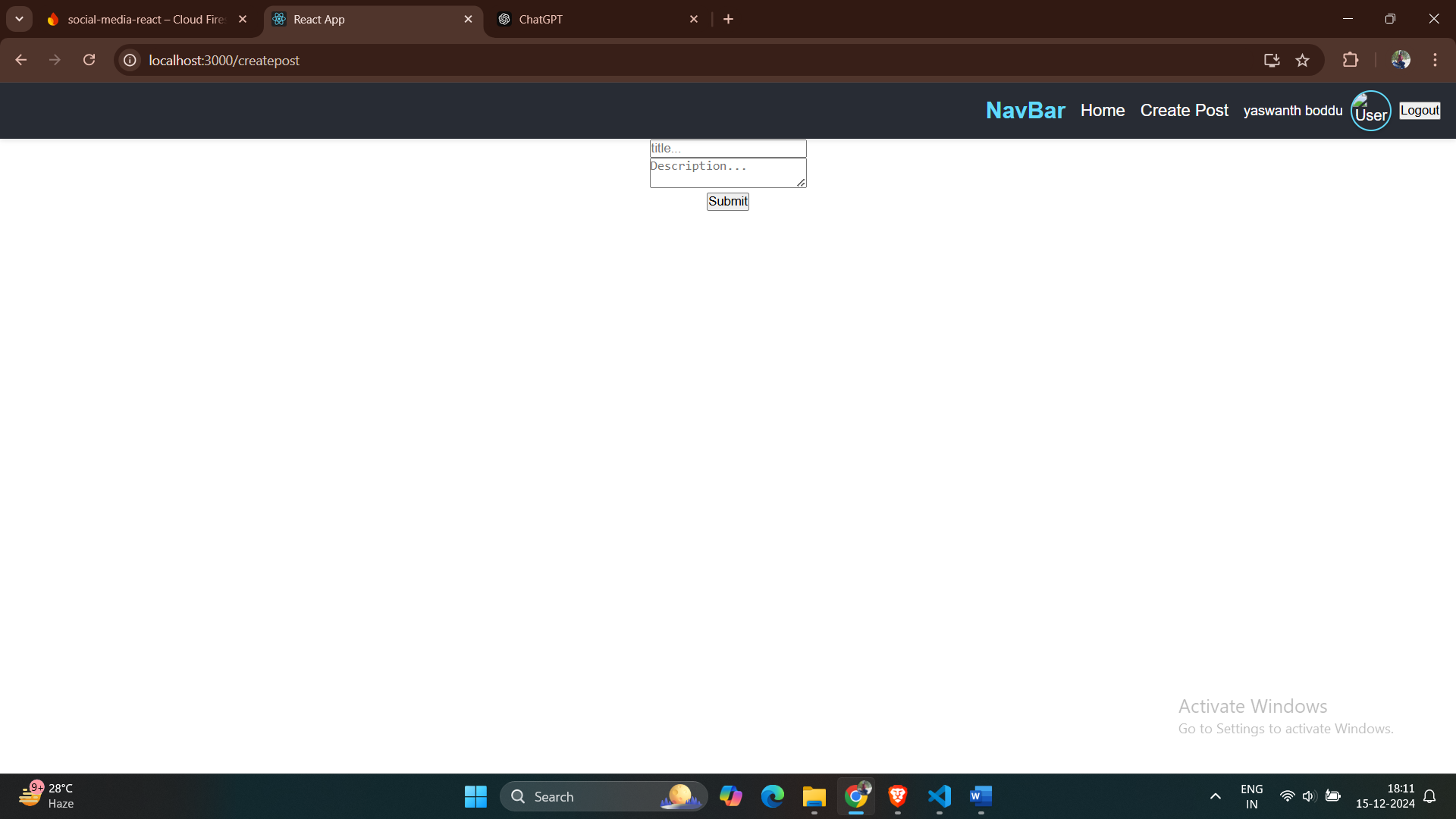
    )

}









npm i @mui/material @mui/icons-material react-to-print

npm i react-barcode

npm i @emotion/react @emotion/styled