Job Portal

Here I am making a full stack job portal website using clerk

Signin to clerk and get the key and keey in .env file

And then modify the main.jsx file

import { StrictMode } from 'react'

import { createRoot } from 'react-dom/client'

import './index.css'

import App from './App.jsx'

import {BrowserRouter} from 'react-router-dom'

import { AppContextProvider } from './context/AppContext.jsx'

import {ClerkProvider} from '@clerk/clerk-react'

const PUBLISHABLE\_KEY = import.meta.env.VITE\_CLERK\_PUBLISHABLE\_KEY

createRoot(document.getElementById('root')).render(

  <ClerkProvider publishableKey={PUBLISHABLE\_KEY} afterSignOutUrl='/'>

  <BrowserRouter>

  <AppContextProvider>

    <App />

  </AppContextProvider>

  </BrowserRouter>,

  </ClerkProvider>

)

import React from 'react'

import { assets } from '../assets/assets'

import { useClerk,UserButton,useUser } from '@clerk/clerk-react'

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

function Navbar() {

const {openSignIn}=useClerk()

*//check user is login or not. method of ckerk*

const {user}=useUser()

  return (

    <div className='shadow py-4'>

     <div className='container px-42xl:px-20 mx-auto flex justify-between items-center '>

        <img src={assets.logo} alt="" />

        {user? <div className='flex items-center gap-3'>

          <Link to='/application'> Applied Jobs</Link>

          <p>|</p>

          <p>Hii,{user.firstName+" " +user.lastName}</p>

          <UserButton/>

        </div> :<div className='flex gap-5 mr- 10 mx-sm:text-xs'>

            <button className='text-gray-600'>Recreuiter Login</button>

            <button onClick={e=>openSignIn()} className='bg-blue-600 text-white px-6 sm:px-9 py-2 rounded-full '> Login</button>

        </div>}

     </div>

    </div>

  )

}

export default Navbar

  <div className='bg-gradient-to-r from-blue-800 to-purple-950 text-white py-16 text-center mx-2 rounded-xl'>

  <p dangerouslySetInnerHTML={{\_\_html:job.dedcription.slice(0,150)}}></p>

{\_\_html:job.description.slice(0,150}