AUTH , HOME PAGE AND ONBOARDING

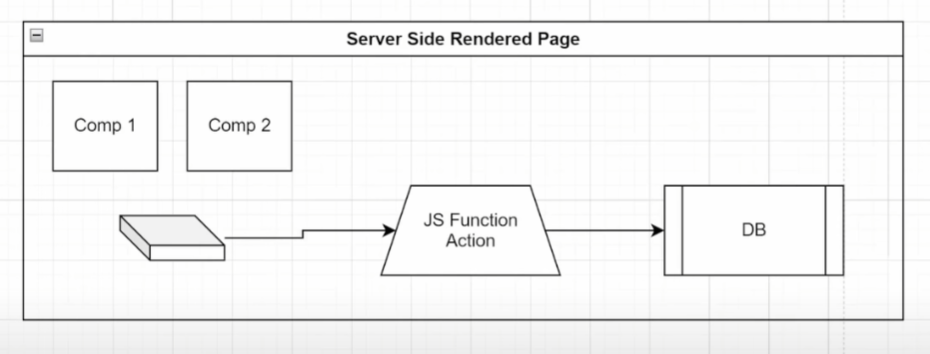
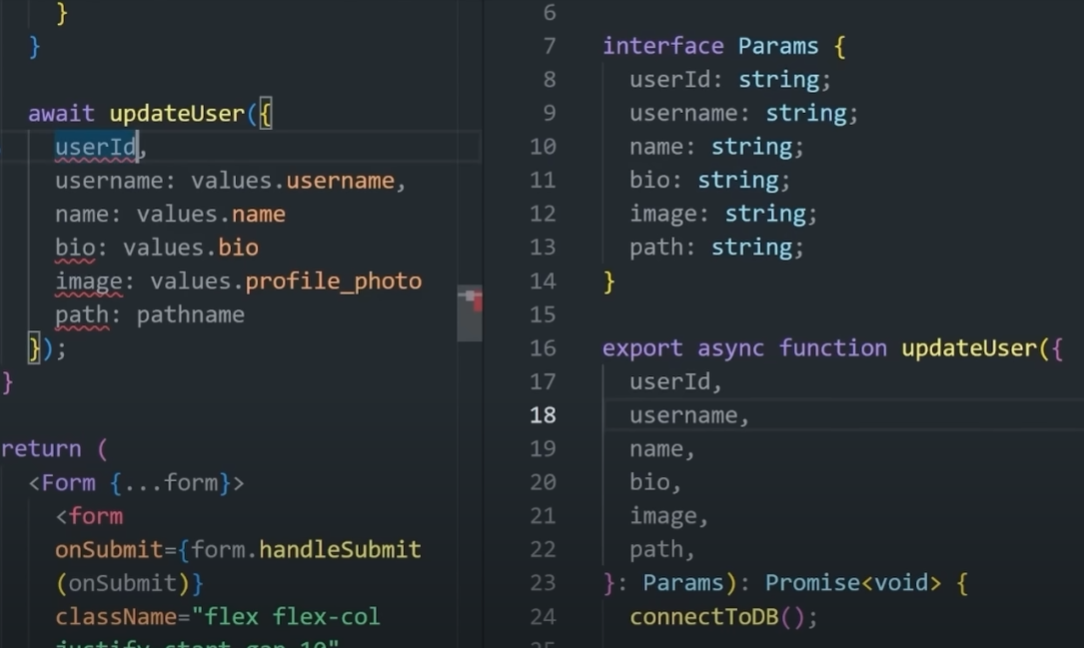
* SEO- search engine optimisation nextjs allows simpler cleaner and efficient seo
* We import fonts from google fonts and in nextjs we separately declare fonts to a const
* We can use clerk components only under clerkprovider and if we use it without it it myt not show wrror but moving to that directory it will show
* Auth to display
  + Layout in auth is wat we get when app is loaded initially
* Root
  + Layout is wat we get after passing thru auth
  + And shd mention afterSingOutUrl= directory/path as “/”
* If updates are not shown try to remove cache in next and run dev again
* In the app folder we only put the file we want the next js to render on the home page that’s y we create components folder outside
* Go thru tailwind website for what specific clases do
* Always link comp shd be there inside the nav tag ,link with href\*\*
* For favicon in next u hsd always have it in app folder and in public
* If that’s doesn’t work then go to ur meta data and create icons call logo icon from assets that’s works well
* Clerk’s advantage of having signedIn and SignedOut components makes work very easier as instead creating Isuserloggedin fn and write the if else code for it can be done using single signedIn comp
* If we are using a non client side render component in next such as userouter pr usepathname then we have to mention “use client” in the top of the text to mention is this is not a client side component only then it works well without error becoz we are using a hook
* And using pathname we can create a isActive const to check is if the path is on the current path or the path going thru this path which helps us to highlight the current active component
* We cant use the component signOutCallback in next15 instead use   
  onClick={() => { // Handle sign-out and navigate to sign-in page router.push('/sign-in'); }}  
  to go to signin page after sign out
* Import dark from clerk/themes for adapting darker themes
* className flex makes the components to look flowy and arrange properly
* shadcn for designed components which are exly accessible customizable and os so we are using them for our forms, they designed forms using react build hook and zod
* zod- typescript first schema validation with static type interface, to create schema for diff fields
* make sure favicon Icon is ONLY IN PUBLIC NOT ANYWHERE ELSE coz next reads for favicon in public folder automatically
* w-full takes the full width of the container
* Your AccountProfile component is set up with a React Hook Form and includes a feature for uploading a profile picture and other user details. The issue with the Invalid src prop error is caused by the <Image /> component trying to render an external image URL (field.value from user?.image), which requires proper configuration in your next.config.js.
  + While accessing external image url we need to modify our next.config.ts for all other remote patterns too
* serverActions true s=is necessary to work with mongoose ans next15 doesn’t support serverActions true directly rather it needs to follow specific structure, i.e just serverAction: {} there is no need to enable true and serverComponentsExternalPackages: is changed to serverExternalPackages: in next15
* for forms use react hook forms zod and shadcn
* using fieldChange we update the field in react hook form
* user.image in react hook represents the value of the image
* if u wanna access useuploadthing fileUrl then in next15 u have to access is as serverData.fileUrl

BACKEND

* standard way of developing websites

A screenshot of a computer

Description automatically generated

* but in the latest nextjs we no longer need nodejs server with backend routes ,considering we already rendering everything on the server side itself 
* note that backend following the above structure will not be routes but will be js actions
* and in every action function don’t forget to mention “user server” as we are using server actions
* MongoDB creds:
  + Username : pranavseelam1305,pranavseelam
  + Pwd : n3WQ11eOyb6Hx1St, JCCWFnaISYttOUZ3
* Upsert: update is existing row and insert if new row {db operation}
* Revalidatepath() functions helps to revalidate while updating cache data i.r revalidate data associate with specific path
* When we await for updateUser it need the parameters to be ordered in the same format and it myt mixup the data in DB so we put all those parametes in a single object and also destructure the updateUser fn and creating a interface params and defining the type of the values in the function,we are destructuring by name,
* For other account iYZDzApWjvHwno0M
* A screen shot of a computer program

  Description automatically generated
* The above is the necessary things to create a basic form