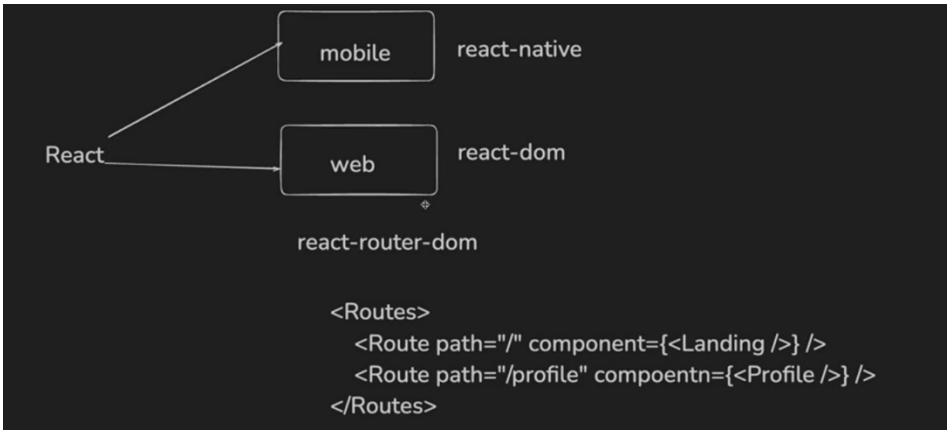
Remix / Next js

Next js contains both frontend and backend for a project as like the react only has frontend

NextJS was a framework that was introduced because of some minor inconviniences in React

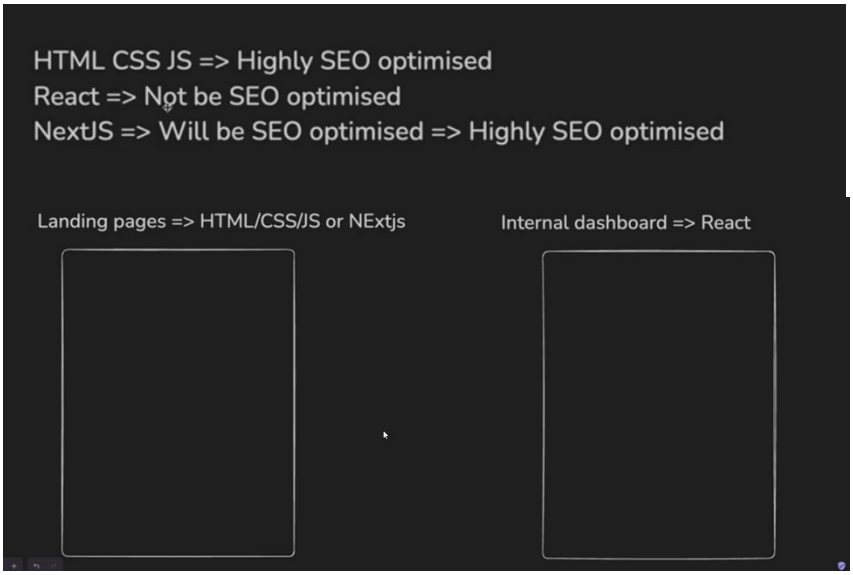
1. In a React project, you have to maintain a separate Backend project for your API routes



2. React does not provide out of the box routing (you have to use react-router-dom)

In React we need like react-native for mobile version, react-dom for web version and react-router-dom for the routing and all

But nextjs does this all without any lib



3.React is not SEO Optimised (Search engine optimization)

not exactly true today because of React Server components

we'll discuss soon why

that's why some companies make the frontend page in html,css ,js or nextjs and other pages in react

eg: neon.tech

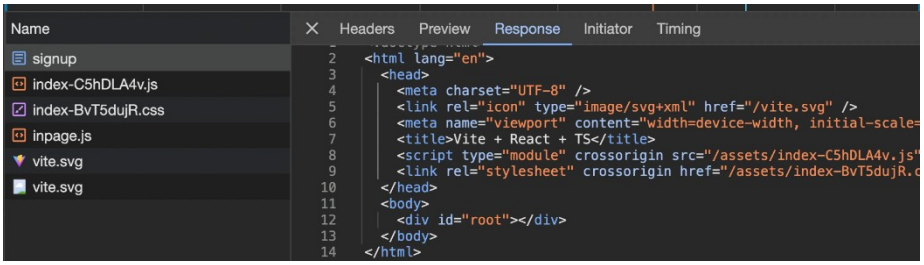
4. Waterfalling problem

Google engine also have something like ranking of the pages its like firstly they crawl the internet and findings which pages are good and which are bad and to find what all the pages are doing and if any page is mentioned in top pages it will get ranked up using ranking algo

Google/Bing has a bunch of crawlers that hit websites and figure out what the website does.

It ranks it on Google based on the HTML it gets back

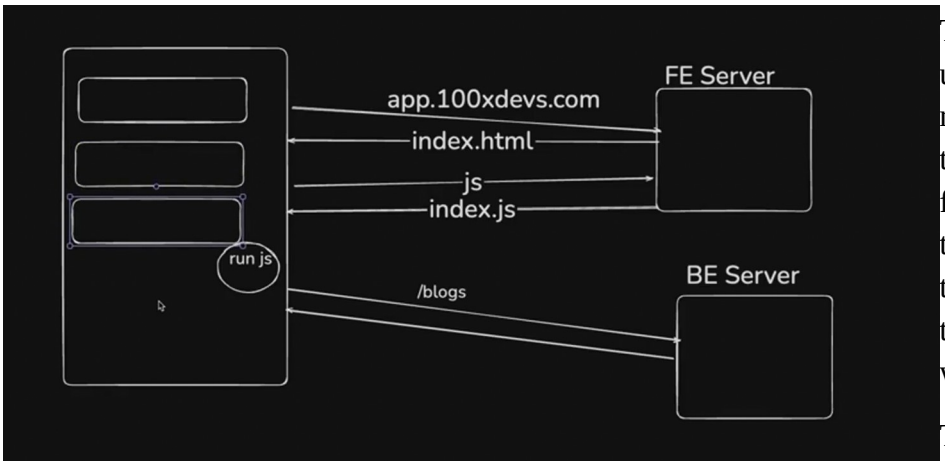
The crawlers DONT usually run your JS and render your page to see the final output.



But the problem is react index.html files does not have the code it just refers to the tsx file so that's why it does not get ranked in the list

Googlebot has no idea on what the project is. It only sees Vite + React + TS in the original HTML response.

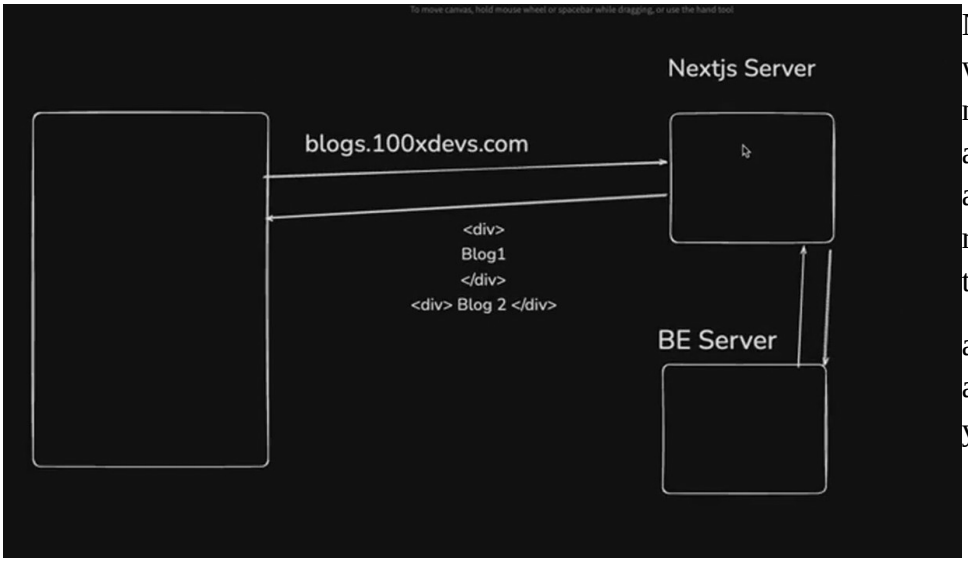
Ofcourse when the JS file loads eventually, things get rendered but the Googlebot doesn't discover this content very well.



This is called **Client Side Rendering** used by react in which when the user run a website firstly hits the fe server then it returns index.html then it asks for js file and then it return index.js and then it runs the run.js and then it hits the backend server and then it get back the blogs part and then shows on the website this is the procedure

This above shit that goes on looks like **waterfall** and the be server is gets hitted as the last that's the problem

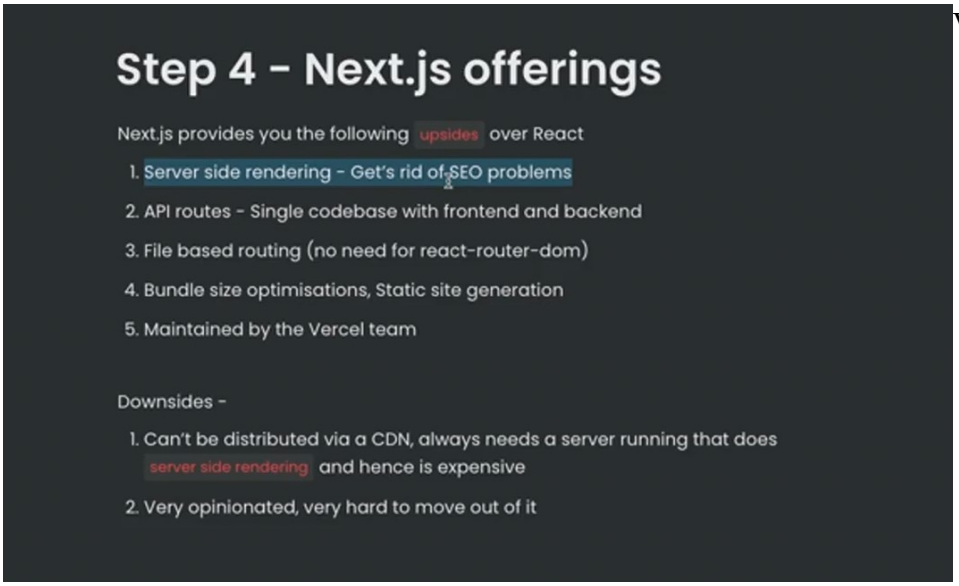
next js does is server side rendering in next js we can do both fe and be



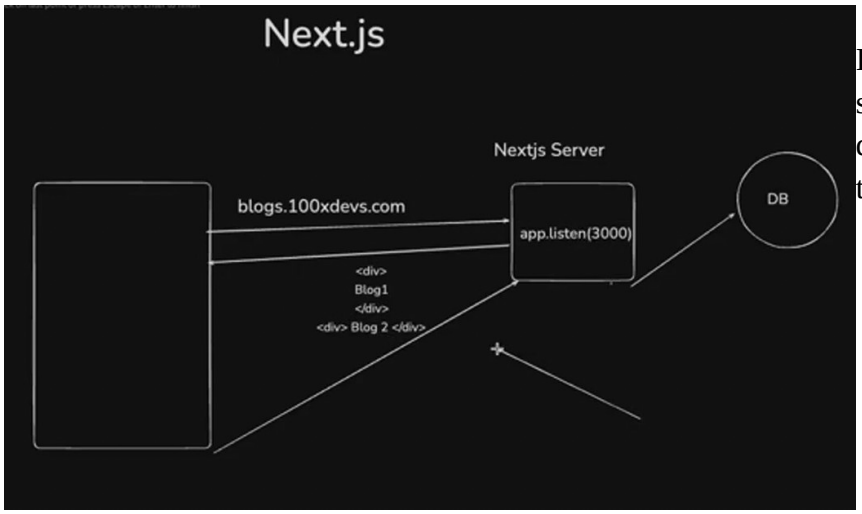
Now in next js what happens is whenever the user hits the url the next js server talks to the be server and it returns asks for the blogs and all and convert that into html and returns the whole data in html form to the user

and agar pheli req mai hii sara html are rha hai toh seo optimised hoga yeh toh bhot jada

in react its very cheap to run in production as after the project is made in npm run build we just do is converting the project into html css and js but in the case of next js actually a server is running and working

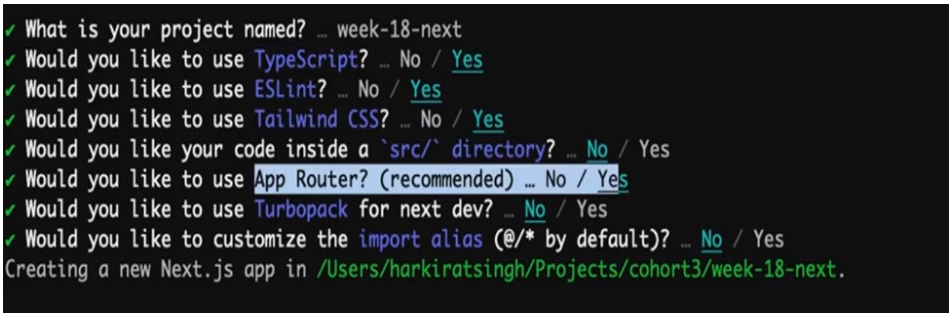


VERCEL Rocks



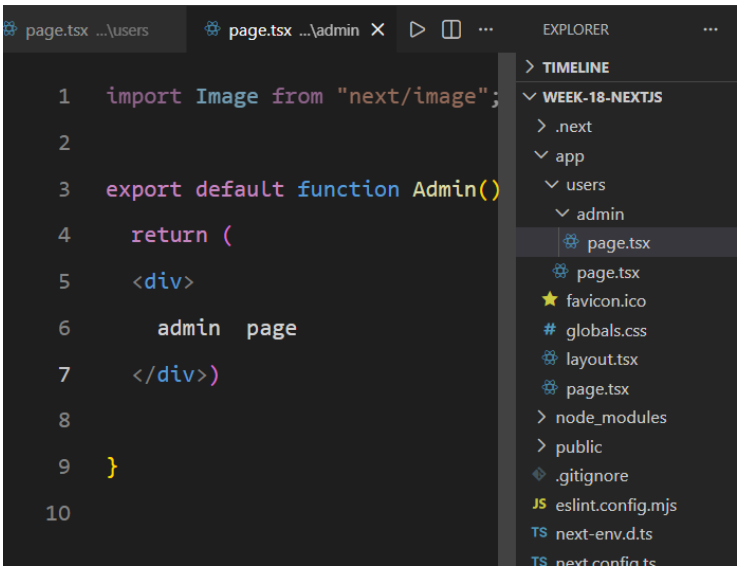
In this when we hit the next js server it runs some server type code access the data from db and then returns the data this is main thing to remember

Setting up a nextjs project



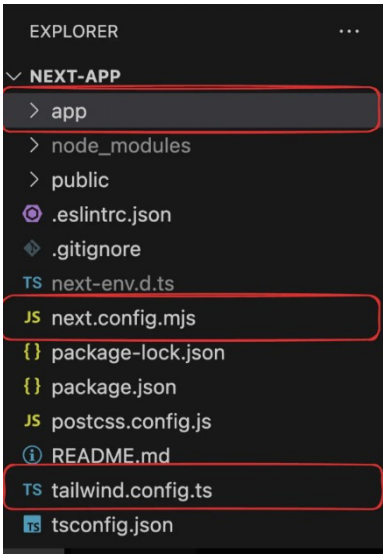
npx create-next-app@latest then use . to create in that folder only

Next js has file based routing means you make a folder a route is created **damnnnn(react router damn is not required)**



see in this http://localhost:3000/users/admin in this url we can see admin page text written

File structure



next.config.mjs - Nextjs configuration file

tailwind.config.js - Tailwind configuration file

app - Contains all your code/components/layouts/routes/apis

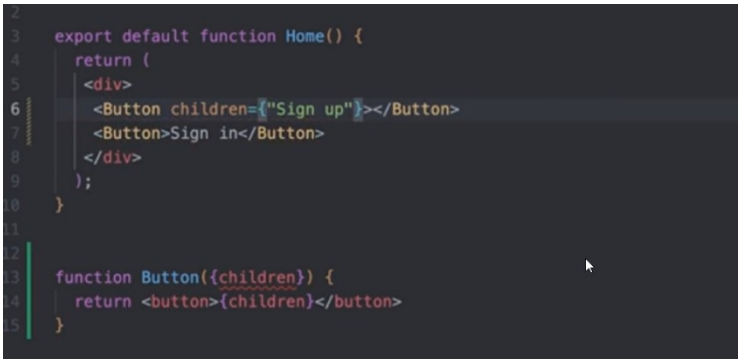
Earlier

Now

```tsx
function App() {

  return (
    <>
      <BrowserRouter>
        <Routes>
          <Route path="/signup" element={<Signup />} />
          <Route path="/signin" element={<Signin />} />
          <Route path="/blog/:id" element={<Blog />} />
        </Routes>
      </BrowserRouter>
    </>
  )
}
```

Next js prefetch the buttons or the links

How does children works in react remember?



```tsx
export default function Home() {
  return (
   <div>
    <Button children={"Sign up"}></Button>
    <Button>Sign in</Button>
   </div>
  );
}


function Button({children}) {
  return <button>{children}</button>
}
```

created a children below and what ever you pass in the button it will used as the children (this is the children prop)

In layout.tsx the metadata is the webpage top name side of the fav icon

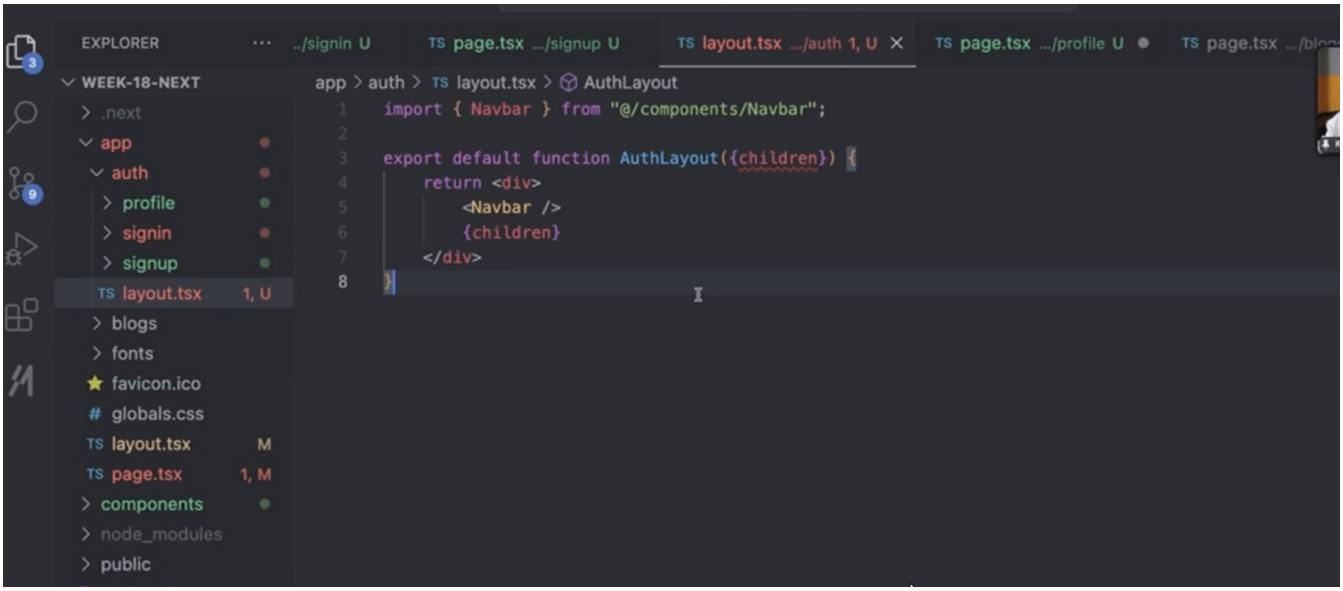
Layout.tsx is imp shit

```tsx
app > layout.tsx > RootLayout
    src: "./fonts/GeistMonoVF.woff",
    variable: "--font-geist-mono",
    weight: "100 900",
  });

  export const metadata: Metadata = {
    title: "Todo application",
    description: "Generated by create next app",
  };

  export default function RootLayout({
    children,
  }: Readonly<{
    children: React.ReactNode;
  }>) {
    return (
      <html lang="en">
        <body
          className={`${geistSans.variable} $
          {geistMono.variable} antialiased`}
        >
          <div className="border-b p-4">Quizio</div>
          {children}
        </body>
      </html>
    );
  }
```

So what is the use of layout.tsx?

In this what happens is like its like a outer structure as you can see it defines the font the nav bar and then the children what is children here?

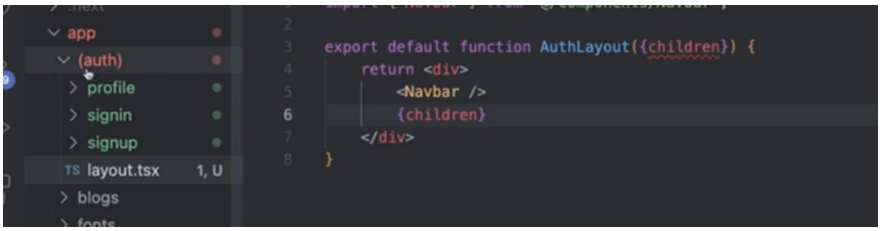
Its like for every page the nav bar will exist but the below data will change so the children is like the home page , about page, contact page etc and the layout.tsx is a wrapper around that



Better usage of layout.tsx is like

Imagine like there is auth folder which contains like profile, signin, signup ok and we want to have nav bar on all on them so what we can do instead of importing the navbar component in each file we can do like just make a layout.tsx file in the auth folder and above the children add a navbar compo and hurray the code repetition decreases and everytime you add any new route in the auth folder it will contain the nav bar

The outer layout.tsx is the main file and its runs like firstly the navbar and then all the components



if you wrap the auth inside the parenthesis then you can directly run like

localhost:3000/profile ignoring auth

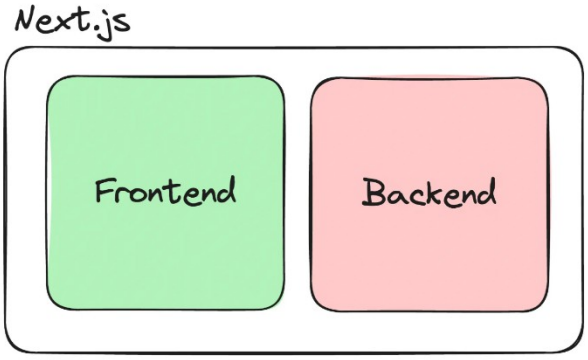
Somethings known as HYDRATION ERROR = when you are rendering something and the data on client and render side becomes different then this error happens

Some companies do these type of things like make the front page in html,css, and js and other all pages in react or front page also in next js

CORS ISSUE OCCURS WHEN THE FRONTEND AND BACKEND ARE ON DIFFERENT ROUTES WHICH MEANS DIFFERENT ORIGINS BUT IN NEXT JS THE ORIGIN IS SAME UNTILL OR UNLESS YOU ARE USING ANY EXTERNAL BACKEND (CROSS ORIGIN RESOURCE SHARING)

Backends in Next.js

Next.js is a full stack framework



This means the same process can handle frontend and backend code

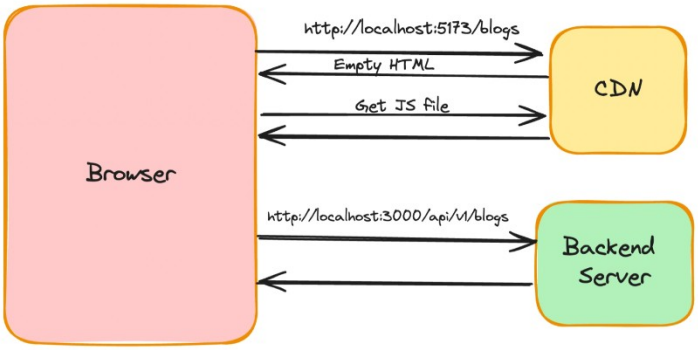
Why?

Single codebase for all your codebase

No cors issues, single domain name for your FE and BE

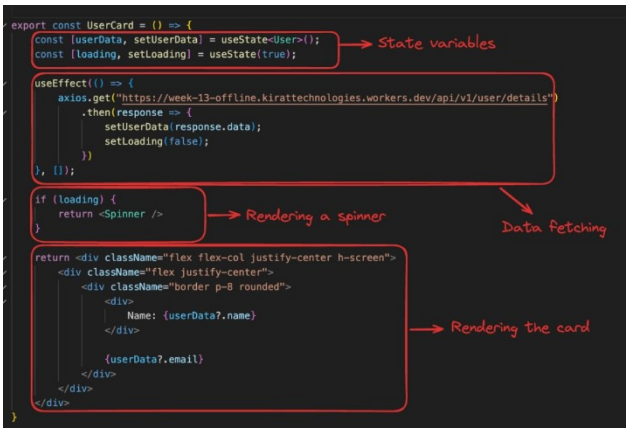
Ease of deployment, deploy a single codebase

Data fetching in React



Data fetching in next js

Earlier in next js there used to be page router but its now app router



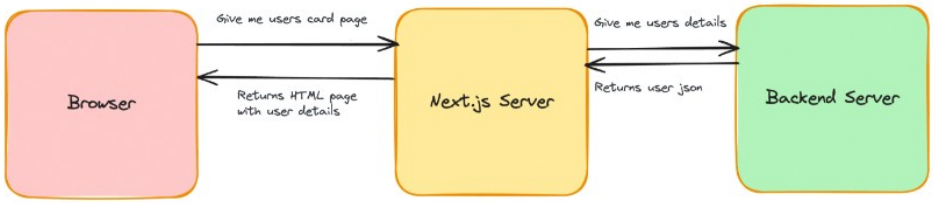
bad way of fetching data in next js

whenever you use usestate, useeffect in the next js at the top you have to add “use client” to make it a client component

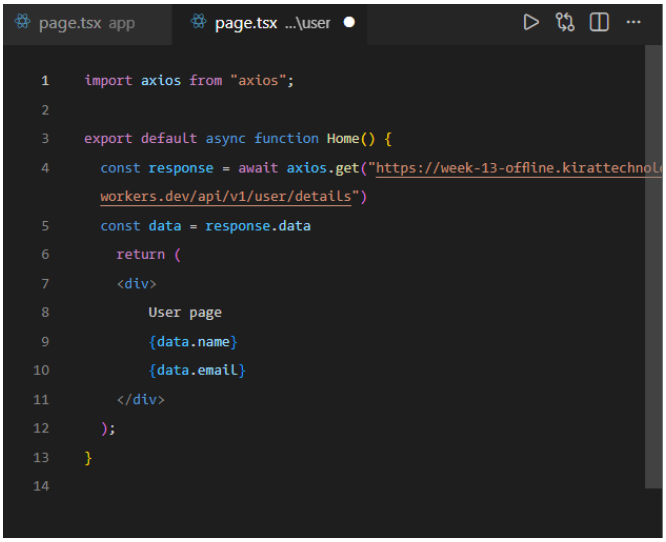
server component cannot use useeffect and usestate that’s why need to use the use client to render it on client side

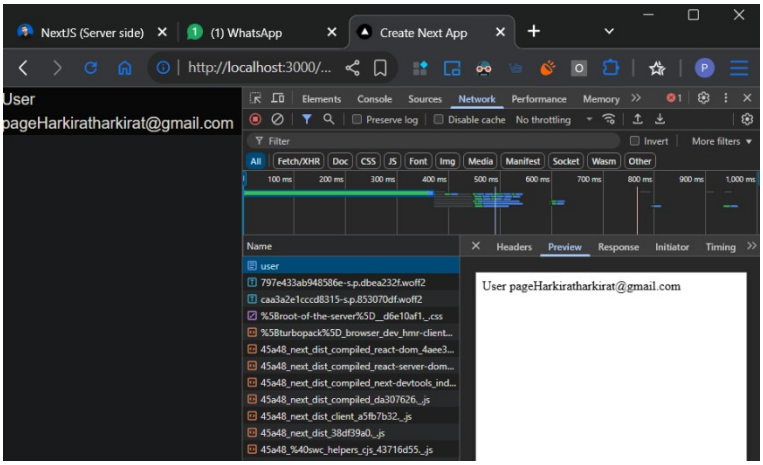
this is not the right data to fetch the data in client way

as the useeffect is running on the client side not on the server side which means the server is stuck on loading and frontend is talking to backend rather than the server side should talk to backend



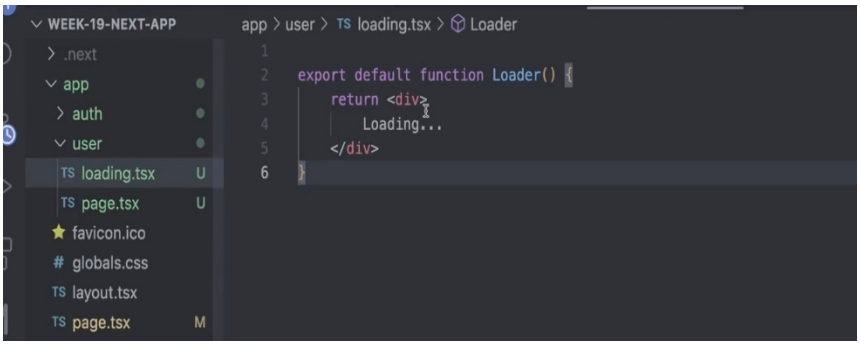
currently this is not happening the user details talk is done btw browser and backend directly which is bad





See in this page is already pre rendered by the next js server the client don't have to hit the backend this is the power

Already rendering the page and getting the data earlier helps in getting the SEO better



the crawler can crawl only the main page not the user page so if you want to add a loading page is your user is stuck in any async code then anything written in loading.tsx will run and when the code is loaded that will be shown and if you don't add loading and the page.tsx is stuck for imagine 5 sec the page will be hung for 5 sec and then the data will be shown

NextJS lets you write backend routes, just like express does.

This is why Next is considered to be a full stack framework.

The benefits of using NextJS for backend includes -

- Code in a single repo
- All standard things you get in a backend framework like express
- Server components can directly talk to the backend

Now if you want to add json data in the page.tsx file you cant so do that create a route.ts file in which you cannot write html and all

LEARNT A IMP THING

```
export function GET() {

    return NextResponse.json({
        user: "harkirat",
        email: "harkirat@gmail.com"
    })
}

export default function UseeHandler() {
    return <div>
    </div>
}
```

WHAT IS THE DIFFERENCE BETWEEN WHEN DEFAULT IS USED AND NOT USED? //

WHAT IS THE DIFFERNCE BETWEEN A FUNCTION WITH DEFAULT AND A FUNCTION WITH CONSTANT?

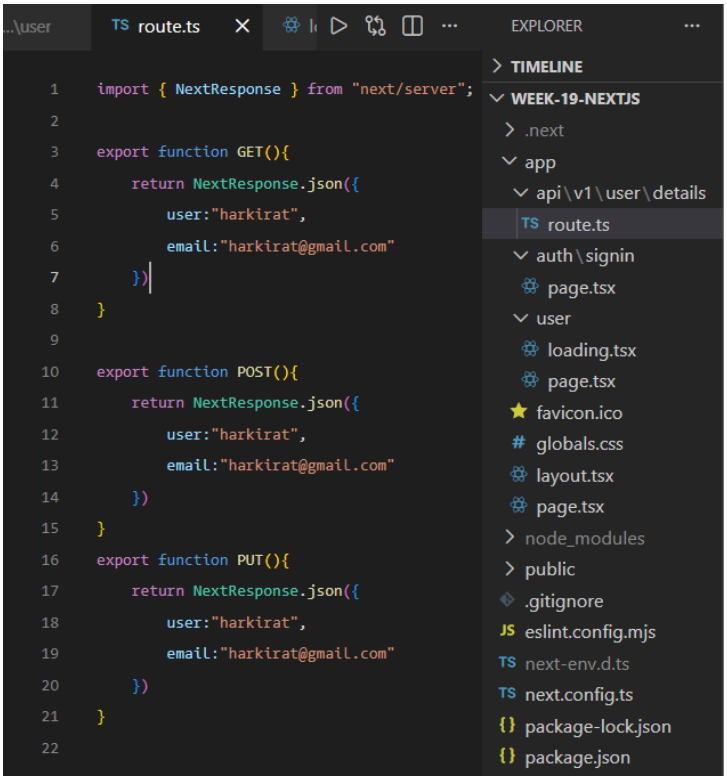
The default function is imported like this

import UserHandler from “./userhandler”;

and the constant export is done like this

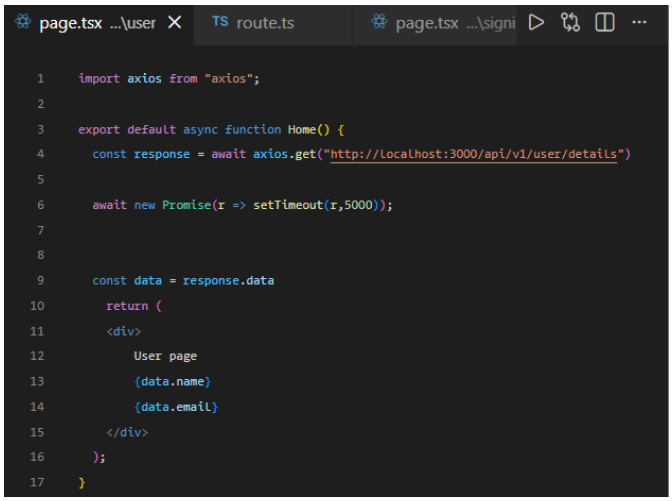
import {GET} from “./get”;

Now how does the next js handles backend?



see the format is little bit different like not a default function anymore and also there is nextReponse library used

THIS ADDS BACKEND TO THE PROJECT



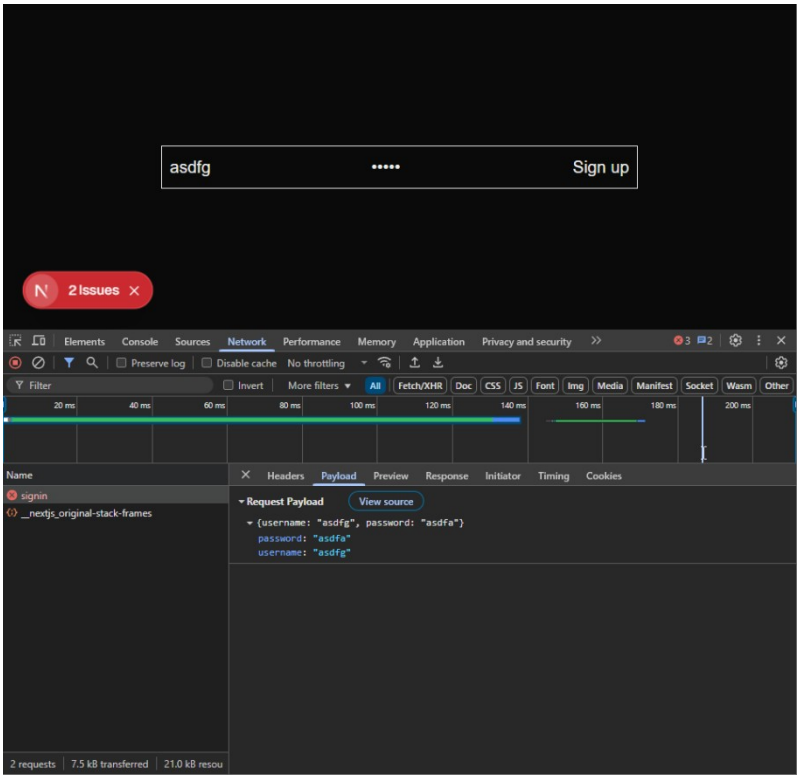
now u can use the backend route like this

await new Promise(r => setTimeout(r,5000)); //SET TIMEOUT FORMAT IN NEXT JS

like in the react we used to use a href for linking the page by clicking on a button but in next js we use link href

Creating a Sign up page

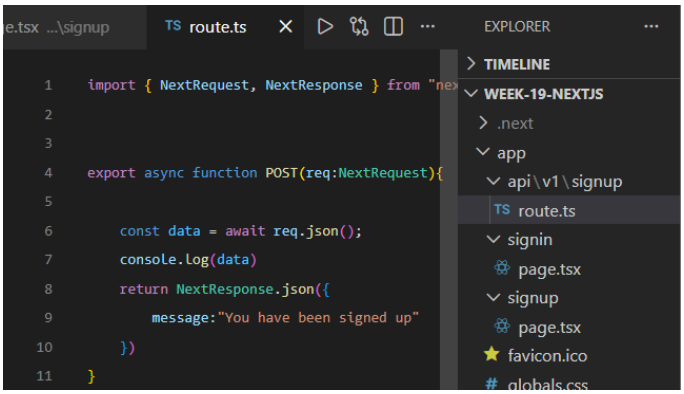
To get data from user we can use Refs, reachhookforms



in the prefetch part the data is shown when we click on the button and code is below

```
"use client"
import axios from "axios"
import { useState } from "react";

export default function Signup(){
    const [username,setUsername] = useState("");
    const [password,setPassword] = useState("");
    return (
        <div className='w-screen h-screen flex justify-center items-center'>
            <div className='border p-2'>
                <input type='text' placeholder='username' onChange={e =>{
                    setUsername(e.target.value);
                }}></input>
                <input type='password' placeholder='password' onChange={e =>{
                    setPassword(e.target.value);
                }}></input>
                <button onClick={() =>{
                    axios.post("http://LocalHost:3000/api/v1/signin",{
                        username,
                        password
                    })
                }}>Sign up</button>
            </div>

        </div>
    )
}
```

Now after getting the data from the user we have to send to backend so that it can send to the db so to do that the route.ts also contains the res thing to get the data from the entry the user sends



see we made the backend code and in this using req we can get the data and also log it and can also return to console if req

Now the next step is to send the data to db

to mongoose or postgress or prisma

for routing from signup page to signin do this

import { useRouter } from "next/navigation";

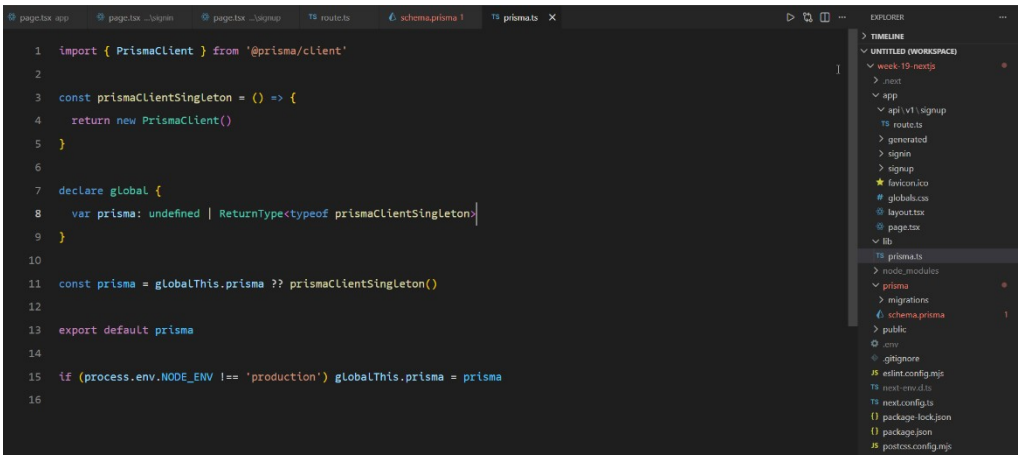
const router = useRouter();

add this after axios code = router.push("/signin");

Setting up prisma with Next js

man install the prisma and then create the user using this and add the data

Now there is a problem like in which when you have db in your system is fine but when in neon tech or any other website in that the prisma client() creates a lot of connection which is a issue to fix that we use **singleton pattern thingy**



this codes means like when the code is running and one connection is made that connection will be pushed to the global prisma which will contain the data of the prisma connection and also does not create multiple connections which fixes this issue

create a lib folder and inside that create prisma.ts

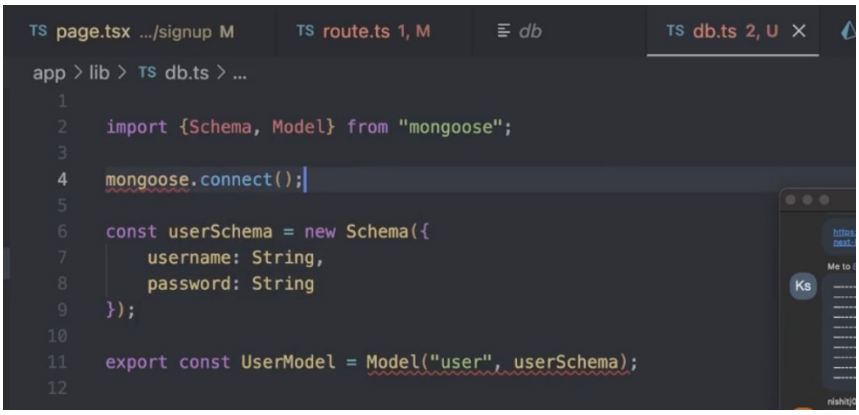
import { PrismaClient } from '@prisma/client'

```
const prismaClientSingleton = () => {
  return new PrismaClient()
}
declare global {
  var prisma: undefined | ReturnType<typeof prismaClientSingleton>
}
const prisma = globalThis.prisma ?? prismaClientSingleton()
export default prisma
if (process.env.NODE_ENV !== 'production') globalThis.prisma = prisma
```

it prevents the hot reload

FOR USING MONGOOSE IN THE DB

Install the mongoose and its types and then



In route.ts use the above url and in the get , post function just use UserModel.insert()

what is module reloading? When the page reloads when we make any changes to the code and apply the changes

WHAT IS HOT MODULE RELOADING IN NEXT JS?

When there is any change in the code the page does not reload

The normal parenthesis is like when you want to skip the auth folder in url then you put this eg: (auth)

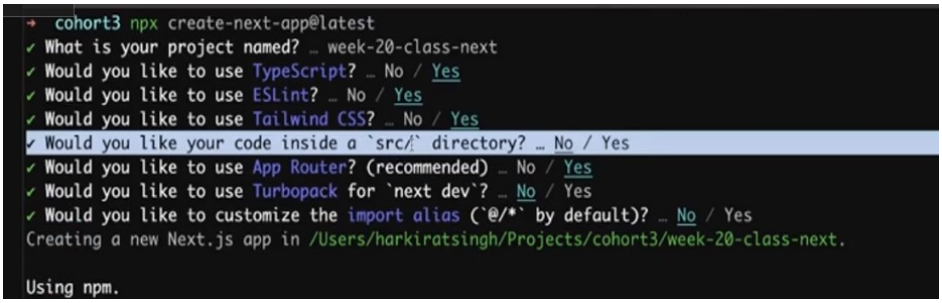
WHAT DOES CACHING MEANS?

Means storing the data so that when later we need to access we can use that

SERVERLESS AND CLOUD

In serverless its like you don’t know where is your server exist its just like you have given the code and they himselft maintaining the code and himself upscaling and downscaling the server

We can choose the src folder option while making the project



Here is the src option mentioned

Jaisa jaisa folder waise waise routing which is called as folder routing

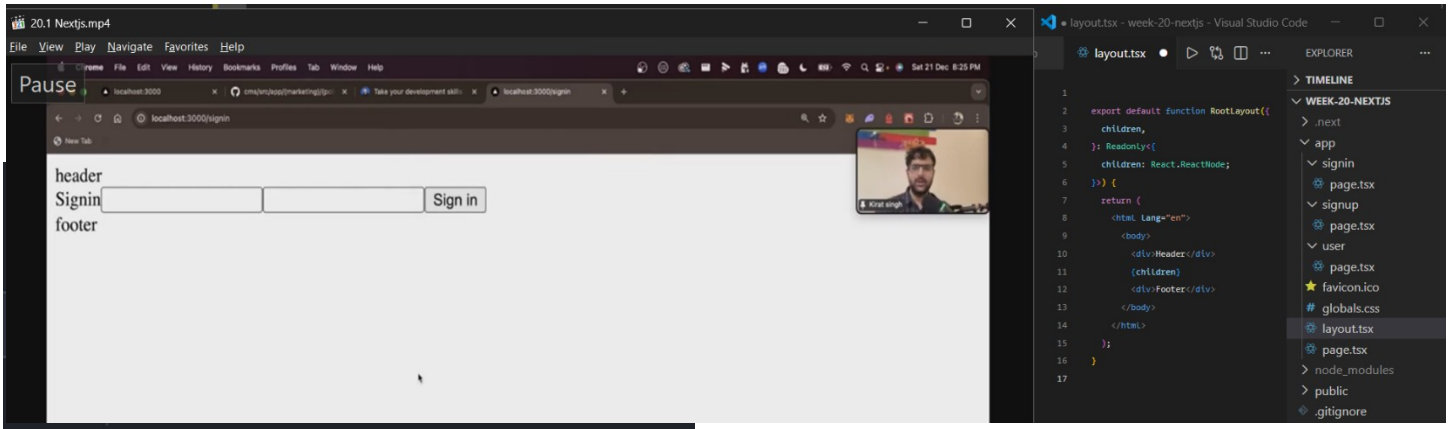
Types of brackets meaning in next js

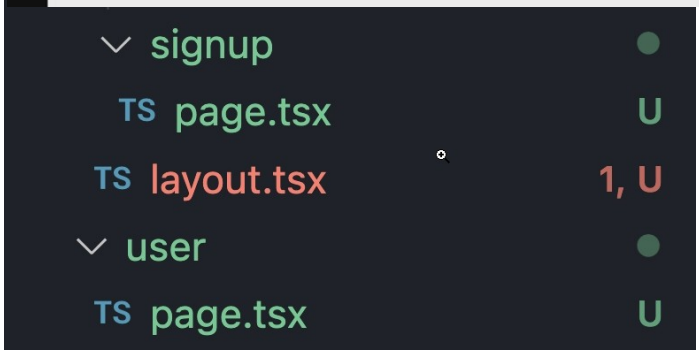
1.Route Groups()

- Parentheses allow you to create **grouped** routes that **do not** affect the URL path.

- For example, if you have a folder named `(marketing)` or `(auth)`, Next.js will not include `(marketing)` or `(auth)` in the URL—it’s just an organizational tool to group certain routes or apply layouts without changing the URL structure.

Main usage of this is the layout thing and can be used in header and footer also





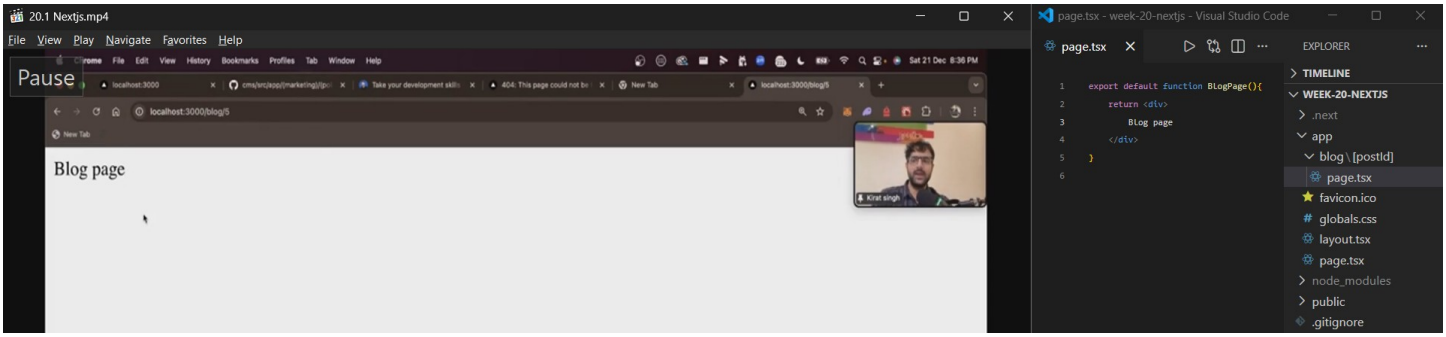
See the use of layout in this

Now the problem is we want to apply the layout to signin and signup only not the user then these brackets comes in picture

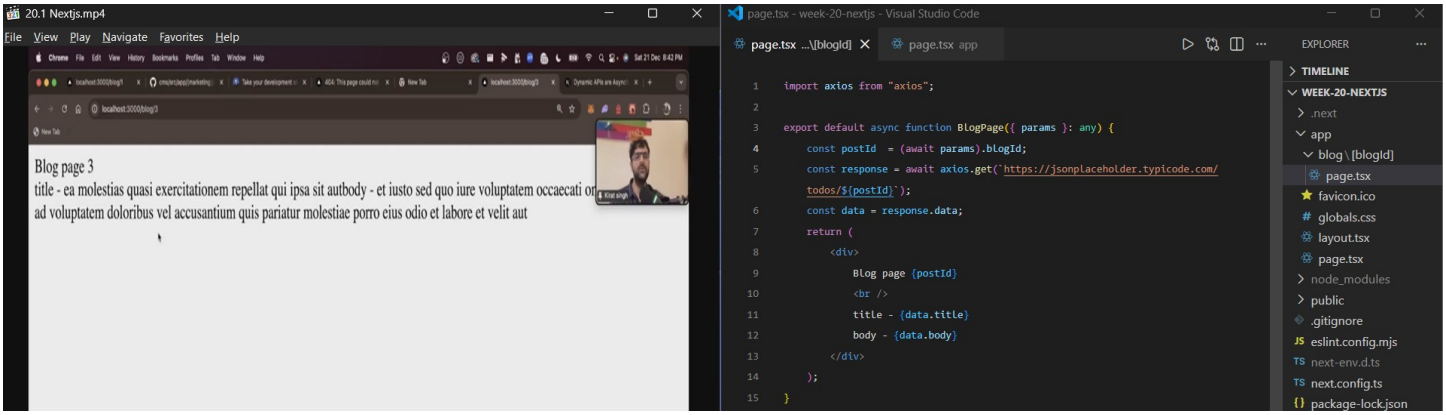
2. Dynamice Routes

What is a slug? == A unique name for a specific page like in utube for my channel the url name contains swinegamers that's a slug

Now in this like you have to create a routes for like many blogs like `localhost:3000/post/1` so cannot do like create a folder for every blog that's inefficient so there this route is used

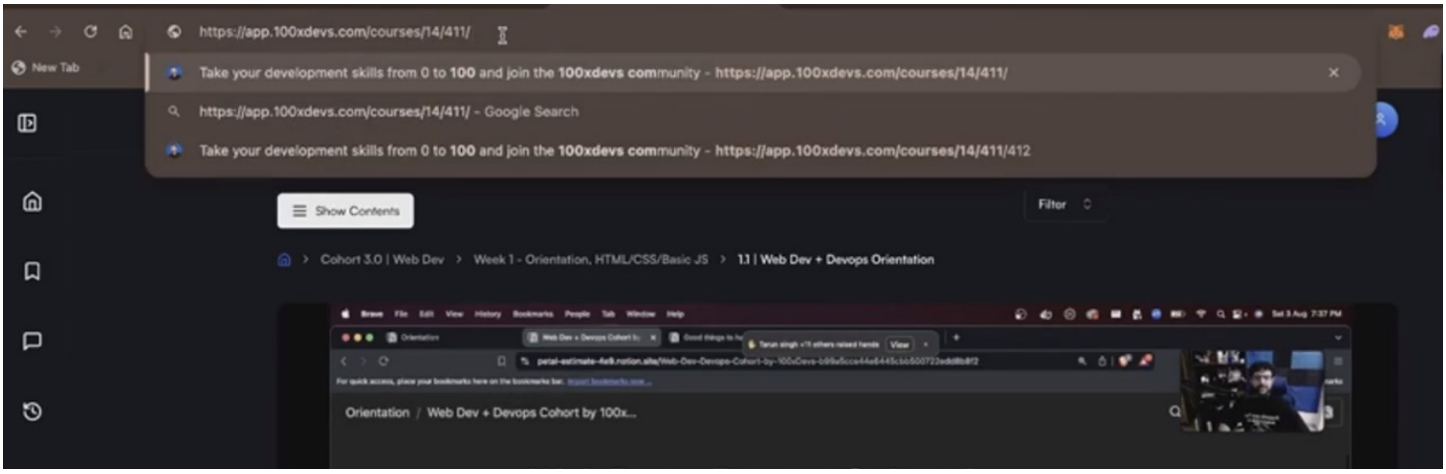


Now anything like blog/1, blog/harkirat etc will be handled but not the blog/1/2 not this



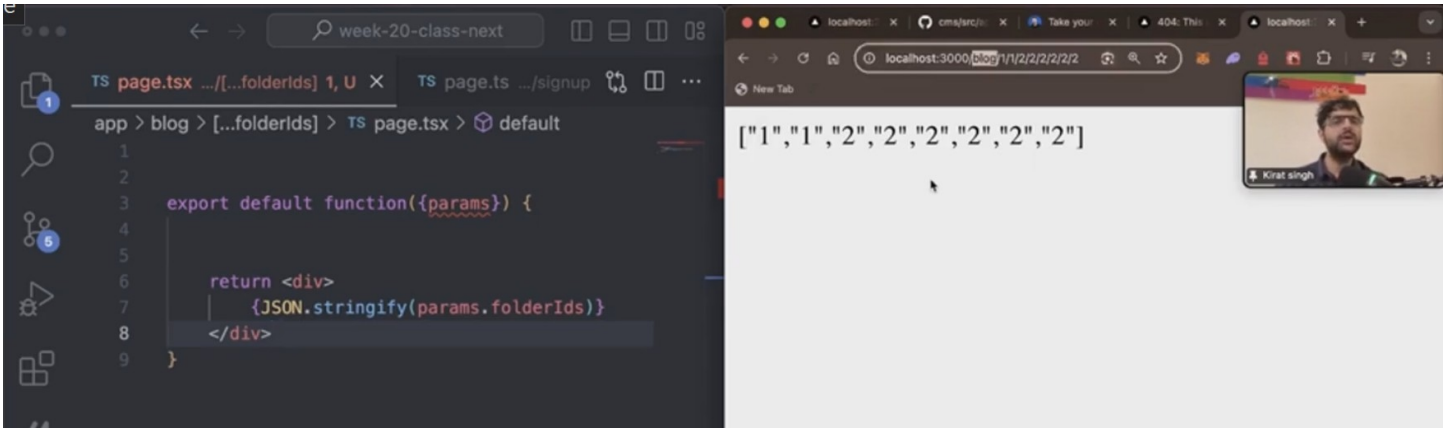
From the newer version of nodejs the params is a promise which needs to be awaited before using

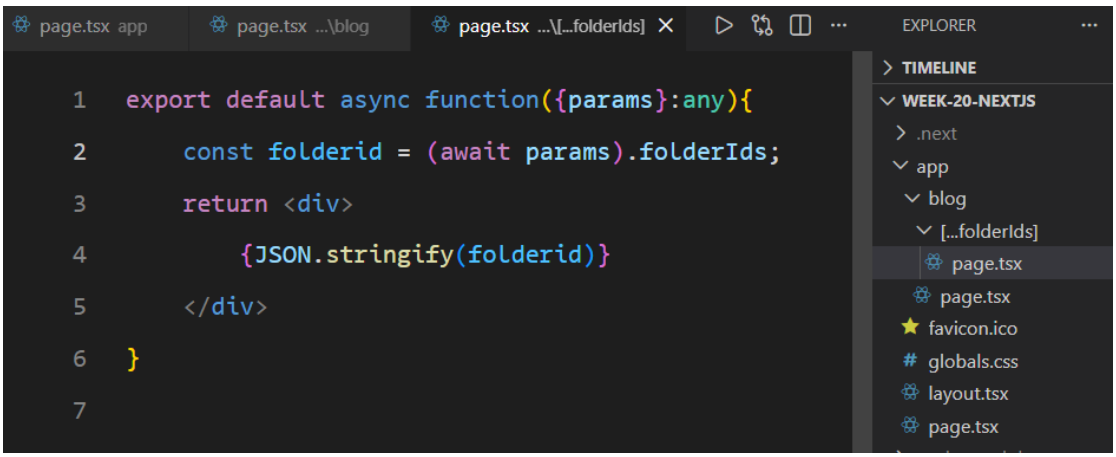
3. Catch all Segment […]



Like in this the issue is the course page will have multiple videos in which there will be many folders like in week 1 there will be 1.1, 1.2 and then in that also there will be video id also to manage all that routes we use this catch all segment

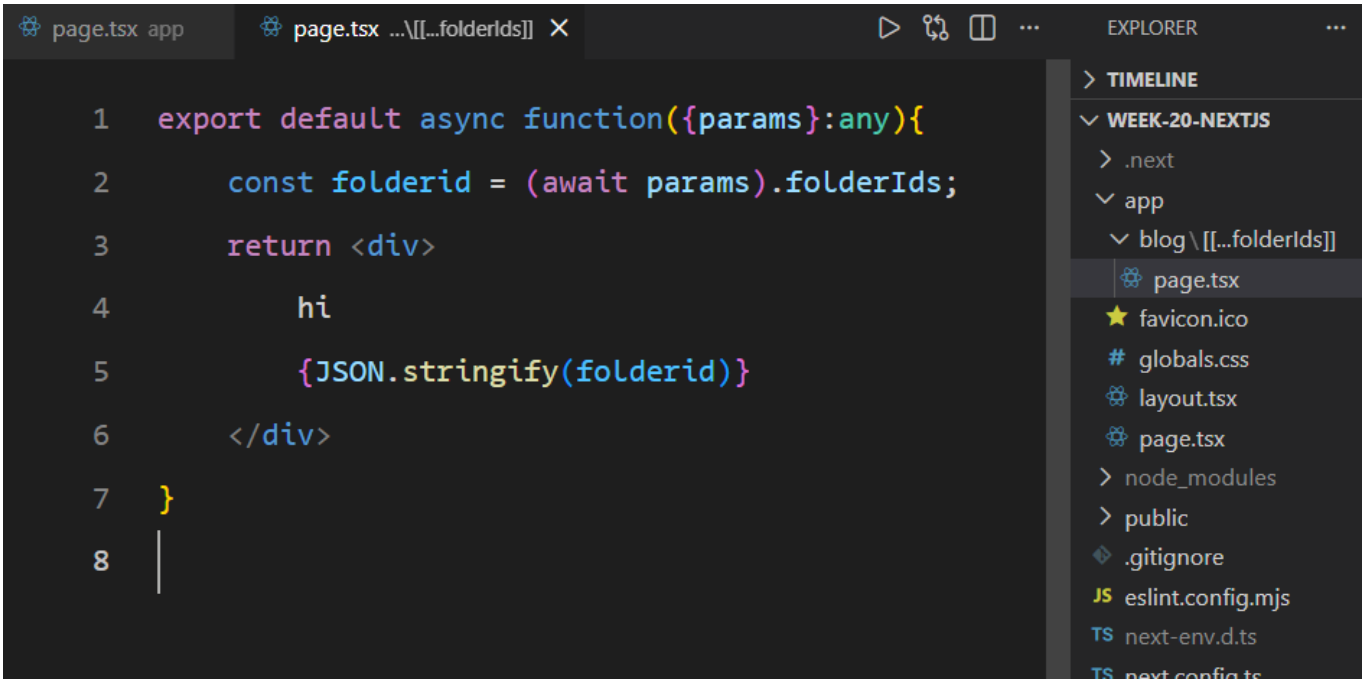
Matlab courses kae baad kuch bhi aaye caught ho jayega



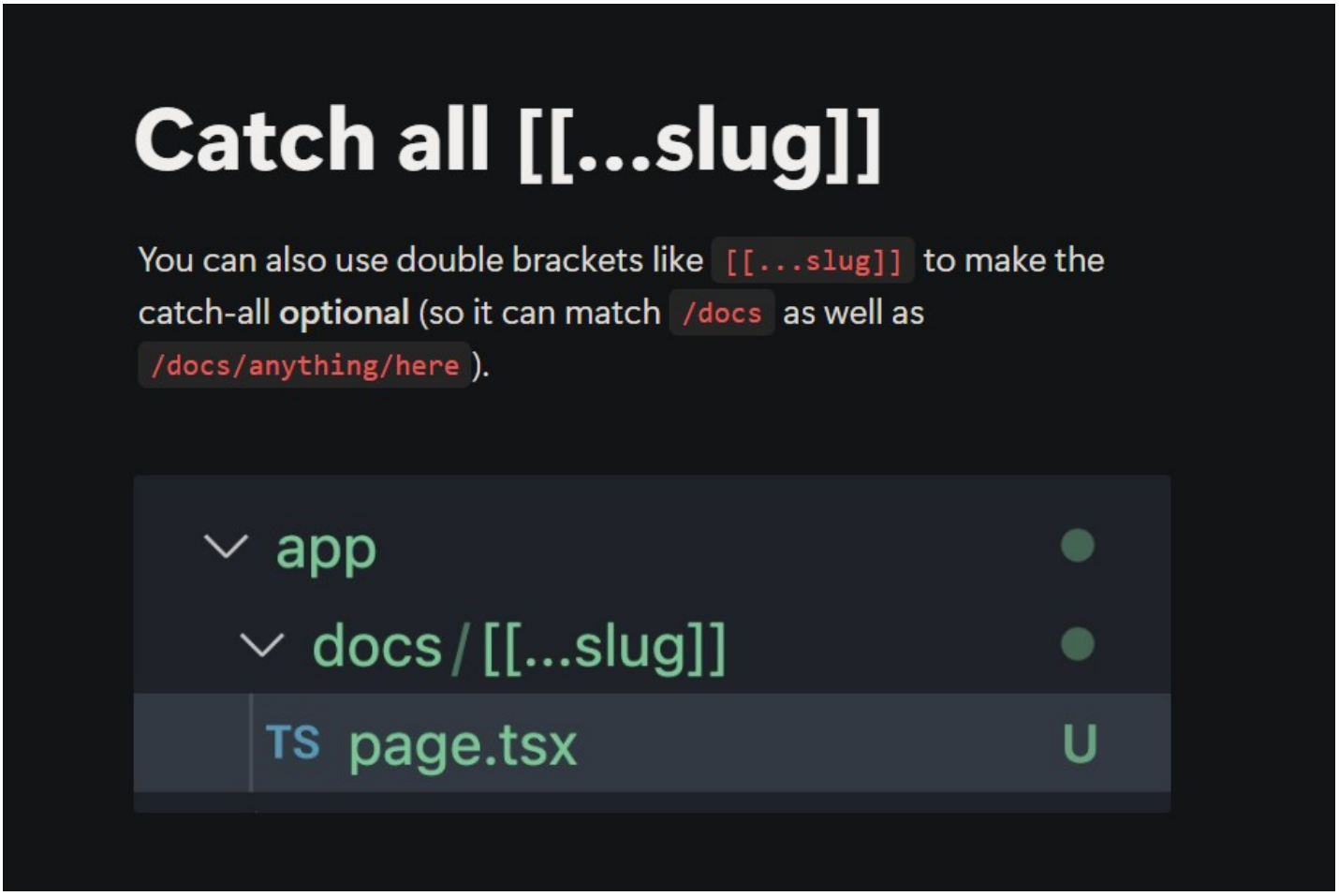


In this the blog route is handled by own and the blog/1/2/3 is handled by folderIds

[[….folderIds]] = this handles the blog route also



/blog will also be handled by this only



Moat = very hard to replicate eg. My typing speed is fast

Nuanced = special thing

<u>There are three things</u>

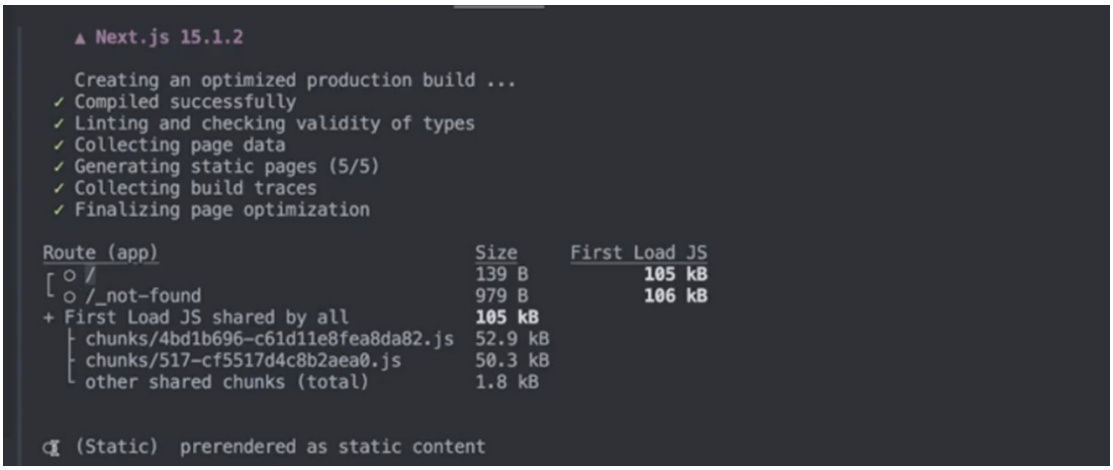
Cilent side rendering (React do this,next js also) but react has this for everything

Server side rendering (Next js can do this) the logic we used to run on the client now runs on the server side and then send to user so that the client side does not have to do anything

Static side Generation its like imagine there is component which is same for all user so just think if its same for everyone why to even use the server side computation power its less only but then also for that we can directly return that component with the index.html

Some code which can be dumped in a html file

Eg run npm run build and in that you can see next js is smart it automatically understands that which page can be used as static



after npm build if its starts with a empty circle means that page can be static generated

Also there are some libraries which helps us in rendering the specific page on client or server side prev we used to use getStaticProps

Before nextjs 13 there was no async, await lol

Whenever you build a `nextjs app`, some pages can be statically generated.

Static generation = HTML is built at build time and is served directly.

SERVER COMPONENTS AND CLIENT COMPONENTS

There are some pages joh server par toh render honge but client par kabhi nahi honge and also aise bhi kuch honge joh kabhi bhi server par nhi honge

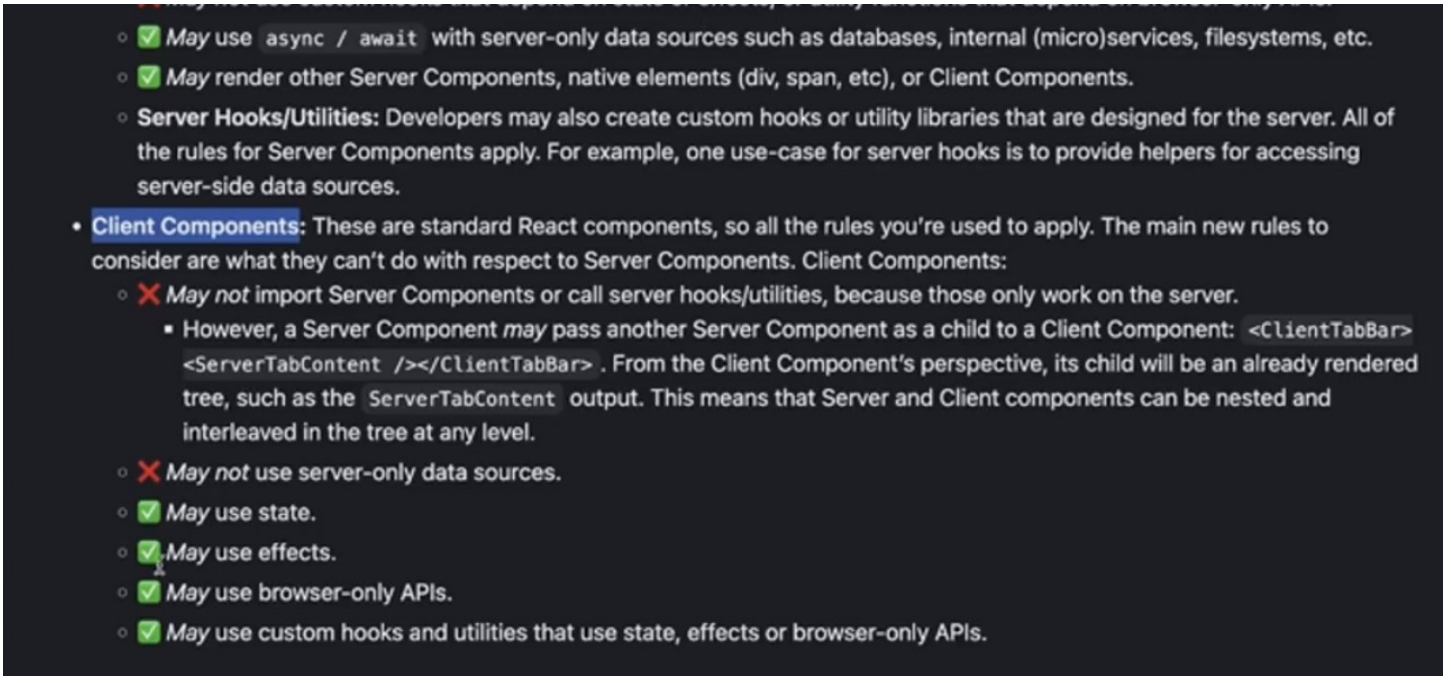
Eg client side a eg a button when we click it increases the number

Eg server side a normal hello page

And this is not true

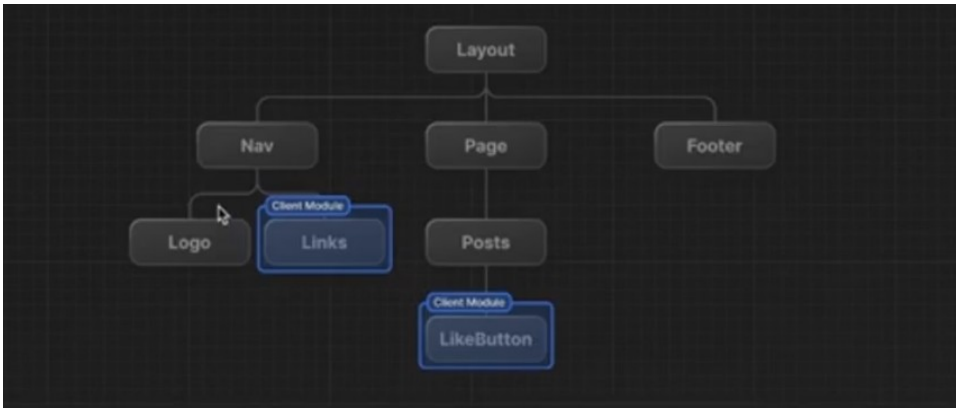
Client component will be client side rendered and server component will be server side rendered

Server components are the top one:



BY DEFAULT EVERY COMPONENT IS A SERVER COMPONENT

In a client component what we get is like html(divs)+ script tag(JS) also but in the server component we get just html and divs or unitlll and unless we are specifically using “use client” and when we explicitly use the server c as client c it will just have some script tag which is JS



Good eg which wil be server and which will be client blue are the client one

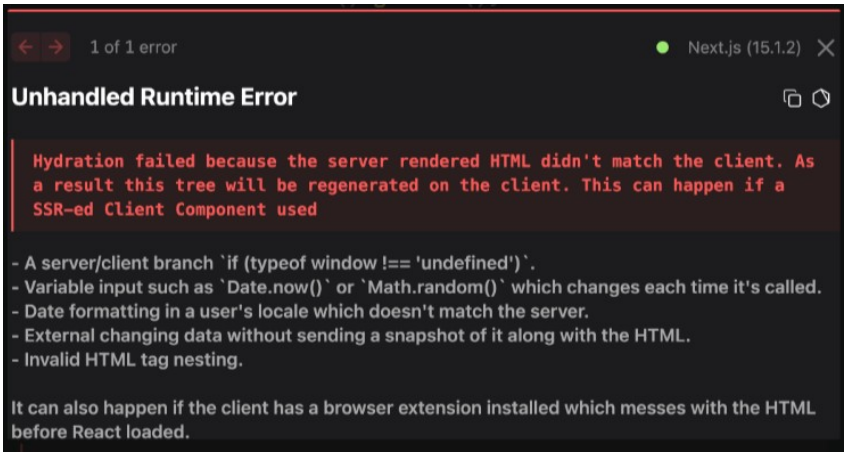
HYDRATION

SERVER COMPONENT MADE ON SERVER AND THEN SENT TO CLIENT

CLIENT COMPONENT MADE ON SERVER AND HYDRATED ON CLIENT

Client component mai hota kya hai static wali chije joh code mai hai who banker aa jati hai client mai fhir joh listeners wagera hai woh fhir attach hotae hai joh html code return hota hai initially usme interactive features nhi honge and fhir who client par jakar hydrate hoga

Hydration is the process by which a client-side JavaScript framework (such as React) takes over an already rendered HTML page and makes it interactive. In a Next.js application, pages are often server-rendered (SSR) or statically generated (SSG). The server sends a fully formed HTML document to the browser, allowing users to see meaningful content quickly (which is great for SEO and performance). Once the page arrives in the browser, React’s JavaScript bundle “hydrates” that static HTML by attaching event listeners and other interactive behaviors so that the page becomes a fully functional React application.



Authentication can be done using libraries such as Passport JS, check the chess video on youtube to check for authentication thingy, OAuth , Next Auth

What is next auth?

NextAuth is a library that lets you do authentication in Next.js

Can you do it w/o next-auth - Yes

Should you - Probably not!

Popoular choices while doing auth include -

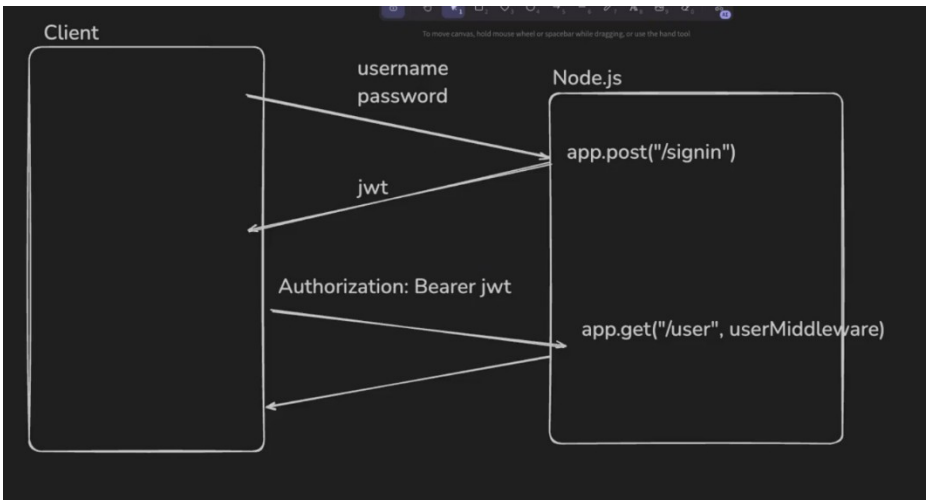
External provider -

1. https://auth0.com/ Used by chatgpt also, significant in the market , they charges based on the number of users

2. https://clerk.com/

3. Firebase auth

4. In house using cookies

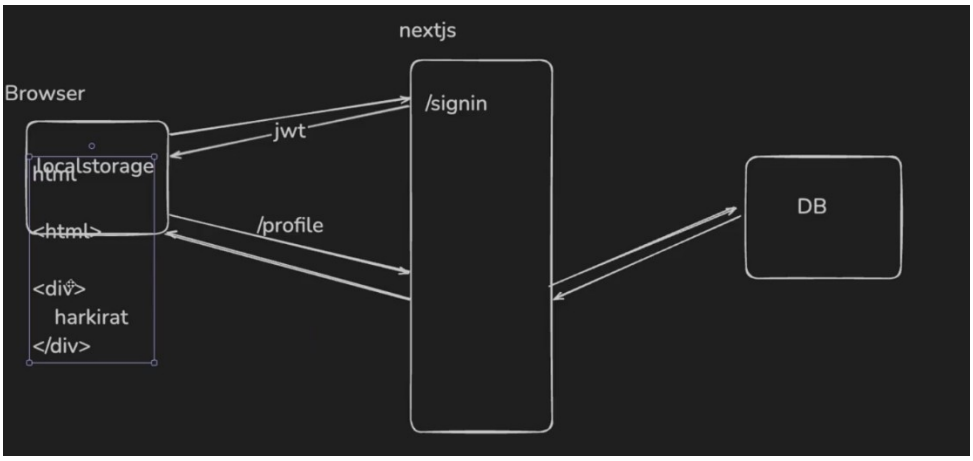


5. NextAuth

Why not use JWT + localstorage?

This is what we used to do in the jwt token when we used to send

But why you cannot do that in nextjs ? bcoz



when the users signin we store the jwt token in local storage but what happens in nextjs is when we hit /profile it does server side rendering during that it does not get the jwt token and then how it will identify that which user is making the req that’s the problem.

The first req does not have the token but the later req can have the token

In the first req we cannot send any token in the req headers

So in nextjs if you want to use server side rendering, you cant use localStorage + header approach , you have to use something else

How we used to use the jsonwebtoken in express?

```
app.post("/signin", (req, res) => {
    const body = req.body;
    const username = body.username;
    const password = body.password;
    const token = jwt.sign({
        userId
    }, "SECRET");

    res.json({
        token
    })
})
```

now in next js

```
import jwt from "jsonwebtoken";
import { NextRequest, NextResponse } from "next/server";

export async function POST(req: NextRequest) {
    // ideally we should check the username and password in the DB and only if it is right
    // we should return the jwt

    const body = await req.json();

    const username = body.username;
    const password = body.password;
    // check in the db

    const userId = 1;
    const token = jwt.sign({
        userId
    }, "SECRET");

    return NextResponse.json({
        token
    })

}
```

In the first req of /profile there is problem that we cannot send the token in that and due to that the server cannot identify who has sent the req and cannot authorize it that’s the main problem to understand but we send like empty html first then we can get the user token in that

Setting up NextAuth

1. install the lib npm install next-auth

2. then copy the default code to app/api/[…nextauth] **//refer to docs**

import NextAuth from "next-auth"

import CredentialsProvider from "next-auth/providers/credentials";

const handler = NextAuth({

    providers: []

})

export const GET = handler

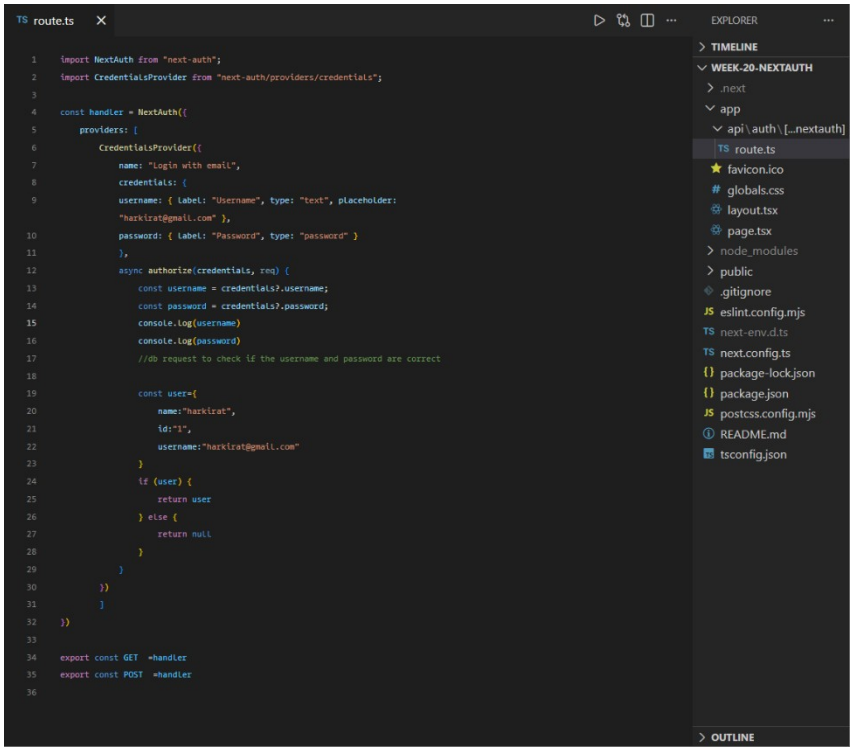
export const POST = handler

authoptions contains the google, facebook mtlb kisse authentication karni hai

providers = jiss company ki auth tumhe use karni hai who

good options : **credentials**(highly configurable)

type: password means it will show ****** not the actual password

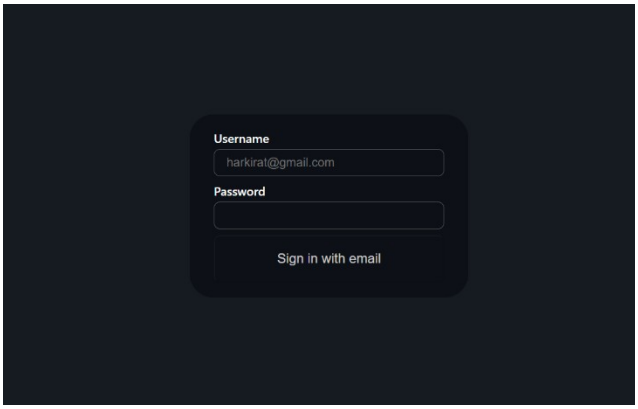


this is done using credentials and this is the format

the commands to send to db is send from here

we used the credentials code given on the website inside this

http://localhost:3000/api/auth/signin this url shows the signin page

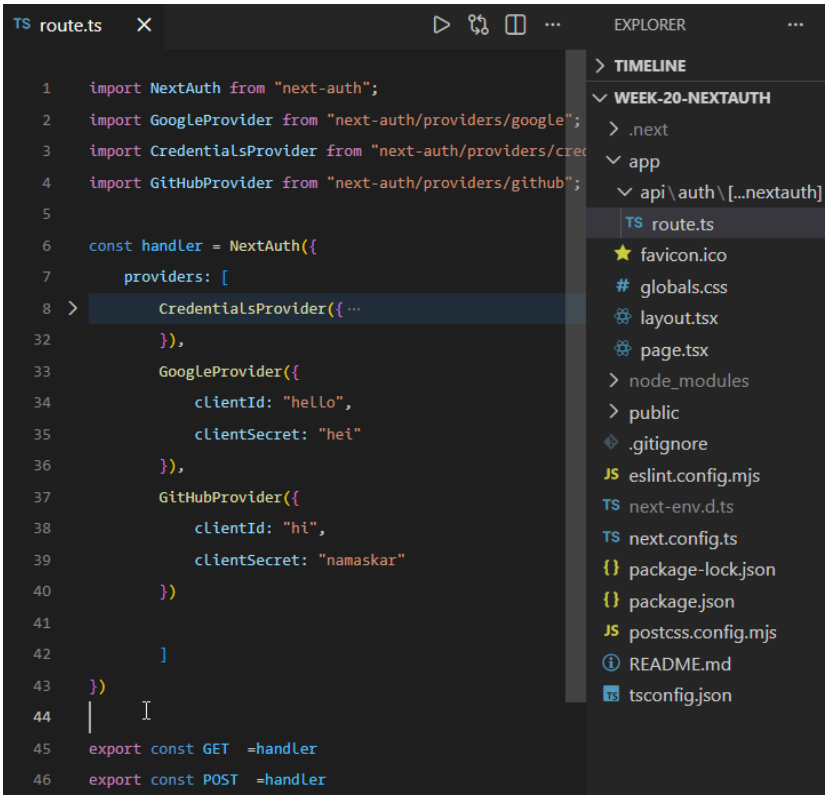


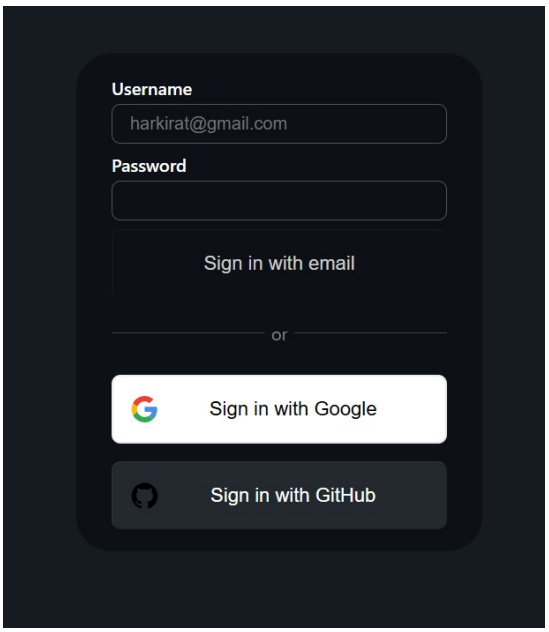
http://localhost:3000/api/auth/signin this url shows the signin page

the credentials thingy is only tough you can use google, facebook other also its easy as f just copy the import and then past the code in the providers section

Adding other providers

```ts
import NextAuth from "next-auth";
import GoogleProvider from "next-auth/providers/google";
import CredentialsProvider from "next-auth/providers/cred
import GitHubProvider from "next-auth/providers/github";

const handler = NextAuth({
    providers: [
        CredentialsProvider({ ···
        }),
        GoogleProvider({
            clientId: "hello",
            clientSecret: "hei"
        }),
        GitHubProvider({
            clientId: "hi",
            clientSecret: "namaskar"
        })

        ]
})

export const GET  =handler
export const POST  =handler
```



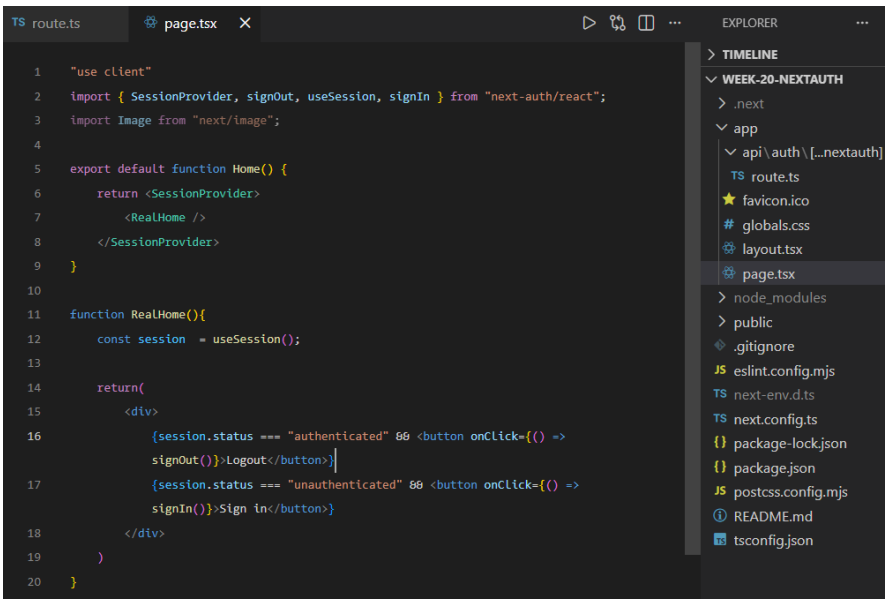


Just copy paste lol

UseSession() easy way to check whether someone is logged in or logged out

This must be wrapped inside the SessionProvider

```ts
"use client"
import { SessionProvider, signOut, useSession, signIn } from "next-auth/react";
import Image from "next/image";

export default function Home() {
    return <SessionProvider>
        <RealHome />
    </SessionProvider>
}

function RealHome(){
    const session  = useSession();

    return(
        <div>
            {session.status === "authenticated" && <button onClick={() =>
            signOut()}>Logout</button>}
            {session.status === "unauthenticated" && <button onClick={() =>
            signIn()}>Sign in</button>}
        </div>
    )
}
```



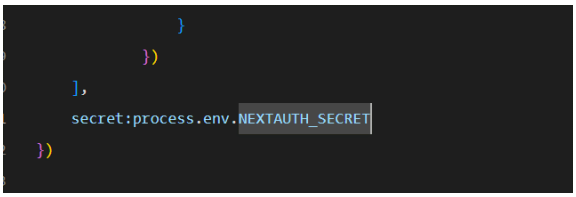
using this code u can like login , logout and signin also and it also redirect to the pages

NOW THE ISSUE IS THAT THE **USESESSION** IS A CLIENT SIDE FUNCTION BUT WE NEED IS SERVER SIDE RENDERING SO TO DO THAT WE USE **GETSERVERSESSION**

Now to do that you have to make a .env file outside

NEXTAUTH_SECRET =1231231sfasdfsaf

And then change the code in page.tsx



Add this in route.tsx for the secret to be accessible