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 -

1. External provider -
   1. https://auth0.com/
   2. https://clerk.com/
   3. Firebase auth
2. In house using cookies
3. NextAuth

Why not use JWT + localstorage?

Next.js is slightly different from React + Express apps

Express app

NextJS app

NextAuth

Nextjs lets you add `Authentication` to your Next.js app

1. It supports various `providers`
    1. Login with email
    2. Login with google
    3. Login with facebook
    4. ...

Catch all routes

If you want to add a single route handler for

1. `/api/auth/user`
2. `/api/auth/random`
3. `/api/auth/123`
4. `/api/auth/...`

You can create a `catch all` route

1. Create a simple next.js app

```
npx create-next-app@latest
```

1. Create `app/api/auth/[...nextauth]/route.ts`

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

export function GET(req: NextRequest) {
    return NextResponse.json({
        message: "Handler"
    })
}
```

1. Try going to a few endpoints

```
http://localhost:3000/api/auth/signin
http://localhost:3000/api/auth/123
http://localhost:3000/api/auth/random/random2
```

1. Try logging the sub-route you're at

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

export function GET(req: NextRequest, { params }: { params: { nextauth:
    console.log(params.nextauth[0])
    return NextResponse.json({
        message: "Handler"
    })
}
```

Give NextAuth access to a catch-all

Ref https://next-auth.js.org/configuration/initialization#route-handlers-app

1. Create `/api/auth/[…nextauth]/route.ts`
2. Install next-auth

```
npm install next-auth
```

1. Updated handler

```
import NextAuth from "next-auth"

const handler = NextAuth({
  ...
})

export { handler as GET, handler as POST }
```

1. Adding providers - There are three broad types of providers
   1. OAuth (Login with google)
   2. Email (Passwordless Email login via email OTP)
   3. Credentials (your own strategy)

Let's them one by one

Credentials provider

This lets you create your own authentication strategy

For example

1. Email + Password
2. Phone number
3. Login with Metamask

Steps to follow

1. Add a credentials provider

```
import NextAuth from "next-auth"
import CredentialsProvider from 'next-auth/providers/credentials';

const handler = NextAuth({
  providers: [
    CredentialsProvider({
        name: 'Credentials',
        credentials: {
          username: { label: 'email', type: 'text', placeholder: '' },
          password: { label: 'password', type: 'password', placeholder:
        },
        async authorize(credentials: any) {

            return {
                id: "user1"
          };
        },
      })
    ],
    secret: process.env.NEXTAUTH_SECRET
  })

  export { handler as GET, handler as POST }
```

1. Add NEXTAUTH_URL to `.env`

Copy

```
NEXTAUTH_URL=http://localhost:3000
NEXTAUTH_SECRET=password_nextauth
```

1. Update `App.tsx` to have a simple Appbar

Copy

```
"use client";
import { signIn, signOut } from "next-auth/react"

export const Appbar = () => {
    return <div>
    <button onClick={() => signIn()}>Signin</button>
    <button onClick={() => signOut()}>Sign out</button>
  </div>
}Click
```

1. Add `providers.tsx`

Copy

```
'use client';
import React from 'react';
import { SessionProvider } from 'next-auth/react';

export const Providers = ({ children }: { children: React.ReactNode }) =
  return (
    <SessionProvider>
      {children}
    </SessionProvider>
  );
};
```

1. Wrap `layout` with `Providers`

```
import { Providers } from "./provide  Copy
                                        ,

export default function RootLayout({
  children,
}: Readonly<{
  children: React.ReactNode;
}>) {
  return (
    <html lang="en">
      <body className={inter.className}>
        <Providers>
          {children}
        </Providers>
      </body>
    </html>
  );
}
```

1. Get the user details in the top level `page.tsx` (client component)

```
"use client"

import { useSession } from "next-auth/react";

export default function Home() {
  const session = useSession();
  return (
    <div>
      {JSON.stringify(session.data?.user)}
    </div>
  );
}
```

1. Get the user details on the server (server component)

```
import { getServerSession } from "next-au

async function getUser() {
  const session = await getServerSession();
  return session;
}

export default async function Home() {
  const session = await getUser();

  return (
    <div>
      {JSON.stringify(session?.user?.name)}
    </div>
  );
}
```

1. Get user in an api route (/api/user)

```
Copy
import { getServerSession } from "next-auth
import { NextResponse } from "next/server";

export async function GET() {
    const session = await getServerSession();

    return NextResponse.json({
        name: session?.user?.name
    })
}
```

1. Persist more data (user id) (Ref https://next-auth.js.org/getting-started/example#using-nextauthjs-callbacks) (Ref https://next-auth.js.org/configuration/callbacks)

```
Copy
  callbacks: {
      jwt: async ({ user, token }: any) => {
          if (user) {
              token.uid = user.id;
          }
          return token;
      },
    session: ({ session, token, user }: any) => {
        if (session.user) {
            session.user.id = token.uid
        }
        return session
    }
  },
```

1. Move auth config to `lib/auth.ts` https://github.com/nextauthjs/next-auth/issues/7658#issuecomment-1683225019

```
import CredentialsProvider from 'next-auth/providers/credentials';

export const NEXT_AUTH_CONFIG = {
    providers: [
      CredentialsProvider({
          name: 'Credentials',
          credentials: {
            username: { label: 'email', type: 'text', placeholder: '' },
            password: { label: 'password', type: 'password', placeholder
          },
          async authorize(credentials: any) {

              return {
                  id: "user1",
                  name: "asd",
                  userId: "asd",
                  email: "ramdomEmail"
              };
          },
        }),
    ],
    secret: process.env.NEXTAUTH_SECRET,
    callbacks: {
        jwt: async ({ user, token }: any) => {
        if (user) {
            token.uid = user.id;
        }
        return token;
        },
      session: ({ session, token, user }: any) => {
          if (session.user) {
              session.user.id = token.uid
          }
          return session
      }
    },
  }
```

Final code -

https://github.com/100xdevs-cohort-2/week-16-auth-2

Adding Google Provider

Ref https://next-auth.js.org/providers/google

Adding Github provider

Ref - https://next-auth.js.org/providers/github

Custom Signup page

What if you want to change how your signup page looks

1. Add `pages` to next auth

   Ref - https://github.com/code100x/cms/blob/main/src/lib/auth.ts#L207

1. Add `app/signin/page.tsx`
   Ref - https://github.com/code100x/cms/blob/main/src/app/signin/page.tsx
   Ref - https://github.com/code100x/cms/blob/main/src/components/Signin.tsx#L39

Copy

```
"use client"
import { signIn } from 'next-auth/react';
import { useRouter } from 'next/navigation';

export default function() {
    const router = useRouter();

    return <div>
        <button onClick={async () => {
            await signIn("google");
        }}>Login with google</button>

        <br />
        <button onClick={async () => {
            await signIn("github");
        }}>Login with Github</button>
        <br />
        <button onClick={async () => {
            const res = await signIn("credentials", {
                username: "",
                password: "",
                redirect: false,
            });
            console.log(res);
            router.push("/")
        }}>Login with email</button>
    </div>
}
```

Final code -

https://github.com/100xdevs-cohort-2/week-16-live-4