Courses Practice Roadmap Pro

Full Stack Development

19 / 22

5 Convert Videos Hosted on Google Cloud Storage 24 min

12 - Add Sign In with Google

18:03

□ Mark Lesson Complete

Add Sign In With Google

1. Add Sign In Button

Create a `sign-in.tsx` file within the `navbar` directory.

```
import { signInWithGoogle, signOut } from
'../firebase/firebase';
```

Add
Firebase

```
import styles from './signIn.module.css';
import { User } from 'firebase/auth';


interface SignInProps {
  user: User | null;
}

export default function SignIn({ user }:
SignInProps) {

  return (
    <div>
      {user ? (
        // If user is signed in, show a
welcome message (or something else)
        <button className={styles.signin}
onClick={signOut}>
          Sign Out
        </button>
      ) : (
        // If user is not signed in, show
sign-in button
        <button className={styles.signin}
onClick={signInWithGoogle}>
          Sign in
        </button>
      )}
    </div>
  );
}
```

And create a `sign-in.module.css` file within the `navbar` directory.

```
.signin {
  display: inline-block;
  border: 1px solid gray;
  color: #065fd4;
  padding: 10px 20px;
  border-radius: 24px;
  font-family: "Roboto","Arial",sans-serif;
  font-size: 14px;
  font-weight: 500;
  cursor: pointer;
}

.signin:hover {
  background-color: #bee0fd;
  border: 1px solid transparent;
}
```

2. Add Sign In Button to Navbar

Update the `navbar.tsx` file to look like this:

```
'use client';

import SignIn from "./signIn";
import Link from "next/link";

import styles from "./navbar.module.css";
import Upload from "./upload";
import { useEffect, useState } from "react";
import { onAuthStateChangedHelper } from
"../firebase/firebase";
import { User } from "firebase/auth";


function NavBar() {
  // Initialize user state
  const [user, setUser] = useState<User |
null>(null);

  useEffect(() => {
    const unsubscribe =
onAuthStateChangedHelper((user) => {
      setUser(user);
    });

    // Cleanup subscription on unmount
    return () => unsubscribe();
  }, [] /* No dependencies, never rerun */);


  return (
    <nav className={styles.nav}>
      <Link href="/">
        <span className=
{styles.logoContainer}>
          <img className={styles.logo}
src="/youtube-logo.svg" alt="YouTube Logo" />
        </span>
      </Link>
      <SignIn user={user} />
    </nav>
  );
}

export default NavBar;
```

We convert it into a client side component by adding `'use client';` to the top of the file.

This is because we want to use hooks like `useState` and `useEffect` which are not allowed in server side components.

This allows us to have 'interactivity' in our component, which means we can update the component after it has been rendered, without having to reload the page.

Here's is a table provided by Next.js of when to use server vs client components:

https://nextjs.org/docs/getting-started/react-essentials#when-to-use-server-and-client-components

3. Test Sign In Button

Run the app with `npm run dev` and test the sign in button.

After signing in with your google account, navigate to the firebase console: **https://console.firebase.google.com/**

Click on the project you created for this course, and navigate to the `Authentication` tab. It may be under the `Build` category.

You should see your google account listed under `Users`.