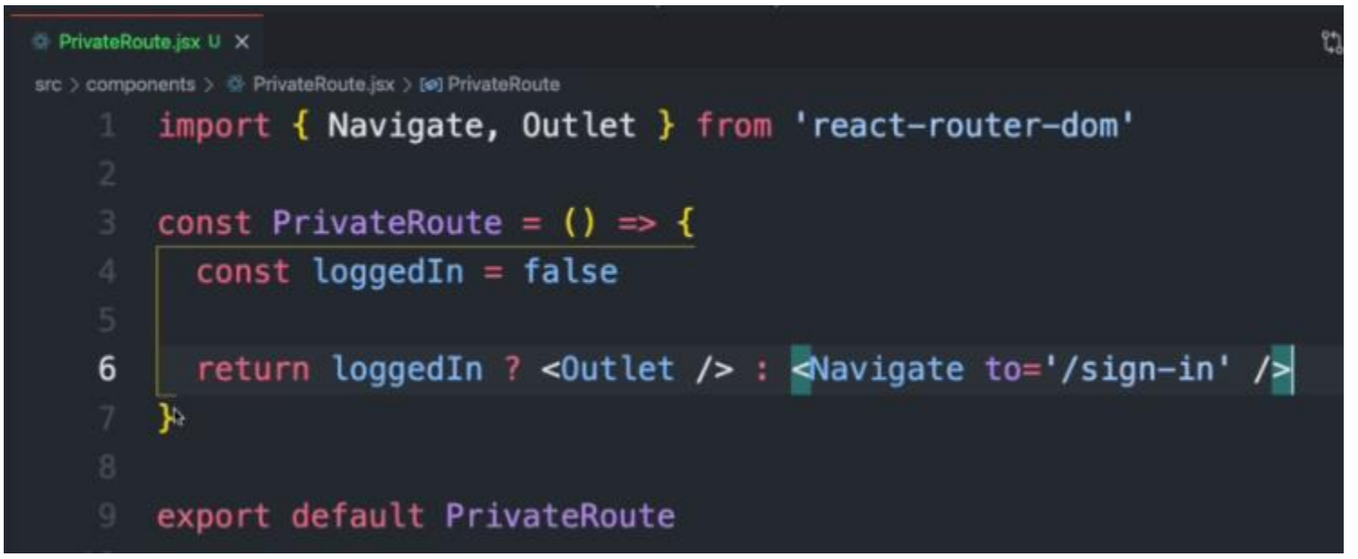
8. PrivateRoute Component & useAuthStatus Hook

Now we want to create private route compt so that any route we have that we don't want the user to access unless they're logged in , we can use that compt !

So in compt folder we create a PrivateRoute.jsx

We import **Navigate Compt** (way to redirect {old redirect compt}) from react router dom

We also need **Outlet** compt which allows us to render a child routes/elements from router ,



If logged in return Outlet (which allows us to return child elements) else return to sign in.

Then ; in app.js we implement that , we have to nest the route which we want to make private

So import Private Route in App.js and :

Inner Profile will be treated as outlet

Ab kisi bhi route ko private krna ha to isi tarah wrap kro except usko outer me path change kro !

Ab ham ne check krna ha k user logged in ha k nahi uske liye ham custom hook banayen gy

Src me folder banao **hooks** ka or wahan ek **useAuthStatus.js** file banao !

Firebase se hame 2 chez chaye !

1.getAuth

2.onAuthStateChanged :

Ye function tab tab fire off hota ha jab jab user log in ya logout krta ha

Ye ham se **auth** leta ha or ek cb function , us function k parameter me hamen **user milta !**

PrivateRoute.jsx U | useAuthStatus.js U | App.js M

src > hooks > useAuthStatus.js > useAuthStatus

```
1  import { useEffect, useState } from 'react'
2  import { getAuth, onAuthStateChanged } from 'firebase/auth'
3
4  export const useAuthStatus = () => {
5    const [loggedIn, setLoggedIn] = useState(false)
6    const [checkingStatus, setCheckingStatus] = useState(true)
7
8    return <div></div>
9  }
```

checkingStatus can be think as of we are loading !

We want to check to see if we are logged in , right after we get the response we will set the CheckingStatus to false & logged in to true !

```
8    useEffect(() => {
9      const auth = getAuth()
10     onAuthStateChanged(auth, (user) => {
11       if (user) {
12         setLoggedIn(true)
13       }
14       setCheckingStatus(false)
15     })
16   })
17
18   return { loggedIn, checkingStatus }
19 }
```

Is hook se ham return kr rhy hen loggedIn or checkingStatus

Ab ham is hook ko private Route me use kren gy is se hamen ye pta chaly ga k user logged in ha ya nahi agar hoga to routes show kren gy otherwise nahi !

PrivateRoute.jsx U | useAuthStatus.js U | App.js M

src > components > PrivateRoute.jsx > PrivateRoute

```jsx
import { Navigate, Outlet } from 'react-router-dom'
import { useAuthStatus } from '../hooks/useAuthStatus'

const PrivateRoute = () => {
  const { loggedIn, checkingStatus } = useAuthStatus()

  if (checkingStatus) {
    return <h3>Loading...</h3>
  }

  return loggedIn ? <Outlet /> : <Navigate to='/sign-in' />
}

export default PrivateRoute
```

Now we need to fix memeory leak !

So in custom hook we need to import useRef,

```jsx
const isMounted = useRef(true)
```

```jsx
useEffect(() => {
  if(isMounted) {
    const auth = getAuth()
    onAuthStateChanged(auth, (user) => {
      if (user) {
        setLoggedIn(true)
      }
      setCheckingStatus(false)
    })
  }
})
```

```
    return () => {
      isMounted.current = false
    }
  }, [isMounted])
```

Now Spinner Compt:

PrivateRoute.jsx U | useAuthStatus.js U | App.js M | Spinner.jsx U

src > components > Spinner.jsx > Spinner

```
function Spinner() {
  return (
    <div className='loadingSpinnerContainer'>
      <div className='loadingSpinner'></div>
    </div>
  )
}

export default Spinner
```

In pvt route :

```
if (checkingStatus) {
  return <Spinner />
}
```

****

Profile.jsx | SignIn.jsx | SignUp.jsx | PrivateRoute.jsx U | Spinner.jsx U | useAuthStatus.js U

src > hooks > useAuthStatus.js > useAuthStatus

```jsx
import { useEffect, useState, useRef } from 'react';
import { getAuth, onAuthStateChanged } from 'firebase/auth';

export const useAuthStatus = () => {
  const [loggedIn, setLoggedIn] = useState(false);
  const [checkingStatus, setCheckingStatus] = useState(true);
  const isMounted = useRef(true);

  useEffect(() => {
    if (isMounted) {
      const auth = getAuth();
      onAuthStateChanged(auth, (user) => {
        if (user) {
          setLoggedIn(true);
        }
        setCheckingStatus(false);
      });
    }
    return () => {
      isMounted.current = false;
    };
  }, [isMounted]);
  return { loggedIn, checkingStatus };
};
```

Profile.jsx | SignIn.jsx | SignUp.jsx | PrivateRoute.jsx U | Spinner.jsx U | useAuthS

src > components > PrivateRoute.jsx > PrivateRoute

```jsx
import { Navigate, Outlet } from 'react-router-dom';
import { useAuthStatus } from '../hooks/useAuthStatus';
import Spinner from './Spinner';

const PrivateRoute = () => {
  const  {loggedIn, checkingStatus}=useAuthStatus()

  if (checkingStatus) {
    return <Spinner />
  }
  return loggedIn ? <Outlet /> : <Navigate to='/sign-in' />;
};

export default PrivateRoute;

```

Profile.jsx | SignIn.jsx | SignUp.jsx | PrivateRoute.jsx U | Spinner.jsx U

src > components > Spinner.jsx > default

```jsx
const Spinner = () => {
  return (
    <div className="loadingSpinnerContainer">
        <div className="loadingSpinner"></div>
    </div>
  )
}

export default Spinner
```

```css
.loadingSpinnerContainer {
  position: fixed;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  background-color: rgba(0, 0, 0, 0.5);
  z-index: 5000;
  display: flex;
  justify-content: center;
  align-items: center;
}

.loadingSpinner {
  width: 64px;
  height: 64px;
  border: 8px solid;
  border-color: #00cc66 transparent #00cc66 transparent;
  border-radius: 50%;
  animation: spin 1.2s linear infinite;
}
@keyframes spin {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}
```