

Module4

Exercise 9(Authentication & Authorization)

Q1.) Create a simple google login feature using NextAuth.js library.

A1.)

```
Module4 > Exercise9 > next-auth > app > api > auth > [...nextauth] > route.ts > handler > providers > clientId
1  import NextAuth from "next-auth";
2  import GoogleProvider from 'next-auth/providers/google'
3  const handler=NextAuth({
4      providers:[
5          GoogleProvider({
6              clientId: process.env.GOOGLE_ID as string,
7              clientSecret:process.env.GOOGLE_SECRET as string
8          })
9      ],
10     pages:{
11         signIn:'/'
12     }
13 })
14
15
16 export {handler as GET,handler as POST}
```

Q2.) Create a Login and Dashboard page.

A2.) Dashboard

```
Module4 > Exercise9 > next-auth > app > dashboard > 📁 page.tsx > 🌐 Dashboard
1  "use client"
2  import { signOut } from "next-auth/react"
3  import { useEffect, useState } from "react"
4  import './page.css'
5  export default function Dashboard() {
6
7      const [users, setUsers] = useState([]);
8
9      useEffect(() => {
10          async function fetchUsers() {
11              const res = await fetch("https://jsonplaceholder.typicode.com/users");
12              const data = await res.json();
13              setUsers(data);
14          }
15
16          fetchUsers();
17      }, []);
18
19      return <>
20          <div className="dashboard-container">
21              <table className="dashboard-table">
22                  <thead>
23                      <tr>
24                          <th>
25                              Id
26                          </th>
27                          <th>
28                              Name
29                          </th>
30                          <th>
31                              UserName
32                          </th>
33                          <th>
34                              Email
35                          </th>
36                      </tr>

```



Login

```
Module4 > Exercise9 > next-auth > app > component > 📁 LoginPage.tsx > 🌐 LoginPage
1  "use client"
2  import { signIn } from "next-auth/react"
3  import './LoginPage.css'
4
5  export default function LoginPage() {
6
7
8      return (
9          <div className="login-container">
10              <p>Please log in</p>
11              <button className="login-btn" onClick={() => signIn("google", { callbackUrl: "/dashboard" })}>
12                  Sign In Google
13              </button>
14          </div>
15      )
16  }
```

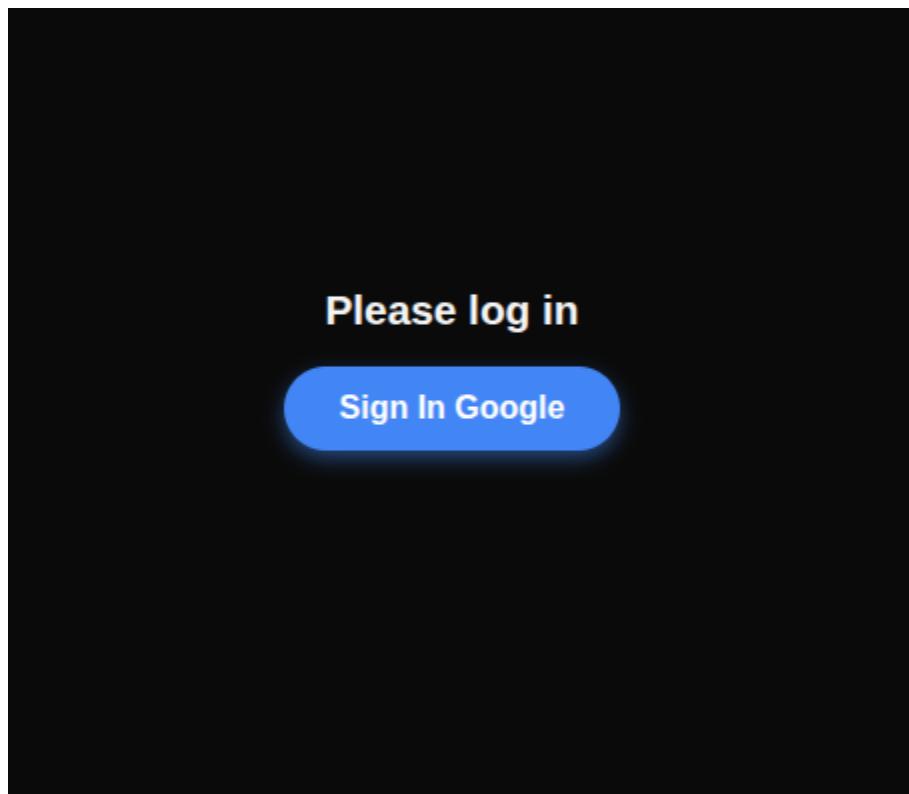


Q3.) After login user should navigate to Dashboard page.

A3.) Login

```
Module4 > Exercise9 > next-auth > app > component > LoginPage.tsx > LoginPage
1  "use client"
2  import {signIn} from "next-auth/react"
3  import './LoginPage.css'
4
5  export default function LoginPage() {
6
7
8    return (
9      <div className="login-container">
10        <p>Please log in</p>
11        <button className="login-btn" onClick={() => signIn("google", { callbackUrl: "/dashboard" })}>
12          | Sign In Google
13        </button>
14      </div>
15    )
16 }
```

Output



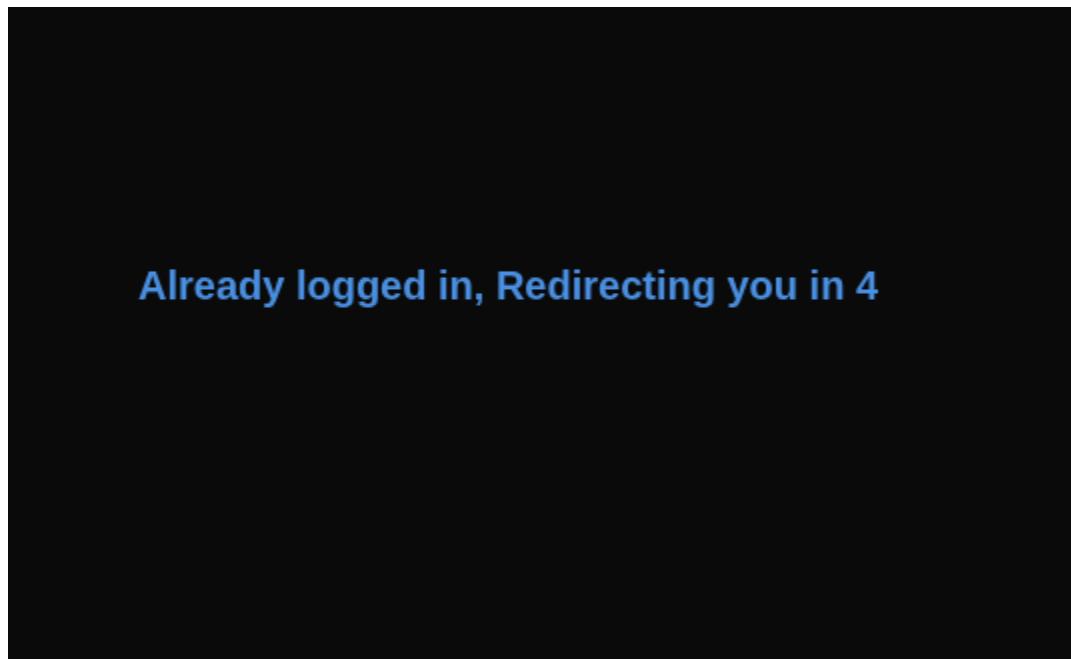
Id	Name	UserName	Email
1	Leanne Graham	Bret	Sincere@april.biz
2	Ervin Howell	Antonette	Shanna@melissa.tv
3	Clementine Bauch	Samantha	Nathan@yesenia.net
4	Patricia Lebsack	Karianne	Julianne.OConner@kory.org
5	Chelsey Dietrich	Kamren	Lucio_Hettinger@annie.ca
6	Mrs. Dennis Schulist	Leopoldo_Corkery	Karley_Dach@jasper.info
7	Kurtis Weissnat	Elwyn.Skiles	Telly.Hoeger@billy.biz
8	Nicholas Runolfsdottir V	Maxime_Nienow	Sherwood@rosamond.me
9	Glenna Reichert	Delphine	Chaim_McDermott@dana.io
10	Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz

Q4.) For logged in user, redirect to dashboard after 5 seconds with text "Already logged in, Redirecting you in {seconds}" (seconds should be live here: 5, 4, 3...)

A4.) Redirection code

```
Module4 > Exercise9 > next-auth > app > page.tsx > ...
1  "use client"
2  import { useRouter } from "next/navigation";
3  import LoginPage from "./component/LoginPage";
4  import { useSession } from "next-auth/react";
5  import { useEffect, useState } from "react";
6  import './page.css'
7
8  export default function Home(){
9    const { data: session } = useSession()
10   const router = useRouter()
11   const [seconds, setSeconds] = useState(5)
12
13   // Countdown effect
14   useEffect(() => {
15     if (!session?.user) return
16
17     if (seconds > 0) {
18       const timer = setTimeout(() => {
19         // Use functional update to get latest value
20         setSeconds(prev => prev - 1)
21       }, 1000)
22
23       return () => clearTimeout(timer)
24     } else {
25       router.push("/dashboard") // Redirect when countdown reaches 0
26     }
27   }, [seconds, session, router])
28
29   if (!session?.user) {
30     return (
31       <LoginPage/>
32     )
33   }
34   return(
35     <p className="redirect-message">Already logged in, Redirecting you in {seconds}</p>
36   )
37 }
```

Output



Already logged in, Redirecting you in 4

Q5.) Client Secret and Client Id should be in .env file.

A5.) .env

```
Module4 > Exercise9 > next-auth >  .env
1 NEXTAUTH_SECRET="IrnvDRSz0GArw2aS3Fdmgoo0t3ifXCUqmjohkL5Zw3k="
2 NEXTAUTH_URL="http://localhost:3000/"
3 GOOGLE_ID="99317314638-qc7idq8e4njsbpk3hm5ogc2niqnp9et6.apps.googleusercontent.com"
4 GOOGLE_SECRET="GO0CSPX-doG_GrSBSXoo2MLzvP4eg9uVJzr0"
5
6
```

Q6.) In dashboard page, fetch list of users using jsonplaceholder dummy API and show there with good UI.

(<https://jsonplaceholder.typicode.com/users>)

A6.) Dashboard

```
Module4 > Exercise9 > next-auth > app > dashboard > 🌐 page.tsx > 📄 Dashboard
1  "use client"
2  import { signOut } from "next-auth/react"
3  import { useEffect, useState } from "react"
4  import './page.css'
5  export default function Dashboard(){
6
7      const [users, setUsers] = useState([]);
8
9      useEffect(() => {
10          async function fetchUsers() {
11              const res = await fetch("https://jsonplaceholder.typicode.com/users");
12              const data = await res.json();
13              setUsers(data);
14          }
15
16          fetchUsers();
17      }, []);
18
19      return <>
20          <div className="dashboard-container">
21              <table className="dashboard-table">
22                  <thead>
23                      <tr>
24                          <th>
25                              | Id
26                          </th>
27                          <th>
28                              | Name
29                          </th>
30                          <th>
31                              | UserName
32                          </th>
33                          <th>
34                              | Email
35                          </th>
36                      </tr>
37                  </thead>
38                  <tbody>
39                      <tr>
40                          <td>1</td>
41                          <td>Leanne Graham</td>
42                          <td>Monica Bertoglio</td>
43                          <td>Leopoldo.Corkery@april.biz</td>
44                          <td>Sincere@april.biz</td>
45                      </tr>
46                  </tbody>
47              </table>
48          </div>
49      </>
50  
```

```
        </tr>
    </thead>
    <tbody>
        {users.map(({id, name, username, email})=>
            <tr key={id}>
                <td>{id}</td>
                <td>{name}</td>
                <td>{username}</td>
                <td>{email}</td>
            </tr>
        )}
    </tbody>
</table>
<button className="logout-btn" type="button" onClick={()=>signOut({callbackUrl:"/"})}>Log
</div>

</>
}
```

Output

Id	Name	UserName	Email
1	Leanne Graham	Bret	Sincere@april.biz
2	Ervin Howell	Antonette	Shanna@melissa.tv
3	Clementine Bauch	Samantha	Nathan@yesenia.net
4	Patricia Lebsack	Karianne	Julianne.OConner@kory.org
5	Chelsey Dietrich	Kamren	Lucio_Hettinger@annie.ca
6	Mrs. Dennis Schulist	Leopoldo_Corkery	Karley_Dach@jasper.info
7	Kurtis Weissnat	Elwyn.Skiles	Telly.Hoeger@billy.biz
8	Nicholas Runolfsdottir V	Maxime_Nienow	Sherwood@rosamond.me
9	Glenna Reichert	Delphine	Chaim_McDermott@dana.io
10	Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz

Q7.) Add a logout button in the dashboard page.

A7.)

```
        </tr>
      </thead>
      <tbody>
        {users.map(({id, name, username, email})=>
          <tr key={id}>
            <td>{id}</td>
            <td>{name}</td>
            <td>{username}</td>
            <td>{email}</td>
          </tr>
        )}
      </tbody>
    </table>
    <button className="logout-btn" type="button" onClick={()=>signOut({callbackUrl:"/"})}>Log
  </div>
</>
}
```

Output

6	Mrs. Dennis Schulist	Leopoldo_Corkery	Karley_Dach@jasper.info
7	Kurtis Weissnat	Elwyn.Skiles	Telly.Hoeger@billy.biz
8	Nicholas Runolfsdottir V	Maxime_Nienow	Sherwood@rosamond.me
9	Glenna Reichert	Delphine	Chaim_McDermott@dana.io
10	Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz

LogOut