ATELIER 2 : NextJS : authentification Client et gestion du panier

Modifier le contenu de



import React from 'react'

const page = () => {

  return (

    <div>

      <h1>APPLICATION</h1>

    </div>

  )

}

export default page

Créer sous /src/components/client le fichier Navabar.js en ajoutant le lien vers la page home



import Link from "next/link";

const NavBar = () => {

return (

<nav className="navbar navbar-expand-lg bg-light">

    <ul className="navbar-nav me-auto mb-2 mb-lg-0">

    <li className="nav-item">

        <Link className="nav-link" href={"/client"}>Home</Link>

    </li>

</ul>

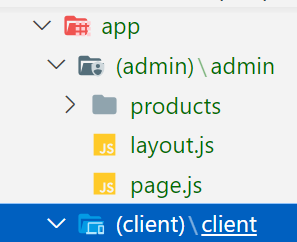
</nav>

);

};

export default NavBar;

Créer le dossier (client) dans le quel un dossier client sous /src/app.



Dans ce dossier créer page.js



import React from 'react'

const page = () => {

  return (

    <div>

      <h1>page client</h1>

    </div>

  )

}

export default page

Créer layout.js sous /src/app/(client)/client



import Navabar from '@/components/client/Navabar'

export default function ClientLayout({ children }) {

  return (

    <>

       <Navabar/>

        {children}

    </>

  )

}

<http://localhost:3000/client>

# Authentification User

## Package next-auth

Dans le projet next installer

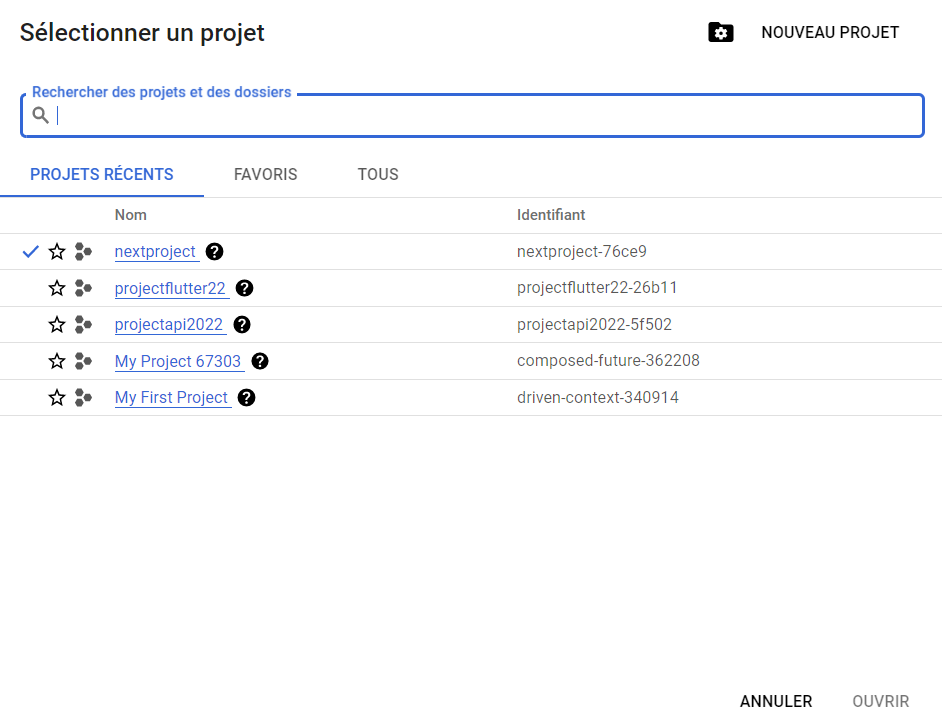
npm i next-auth

## Google Login

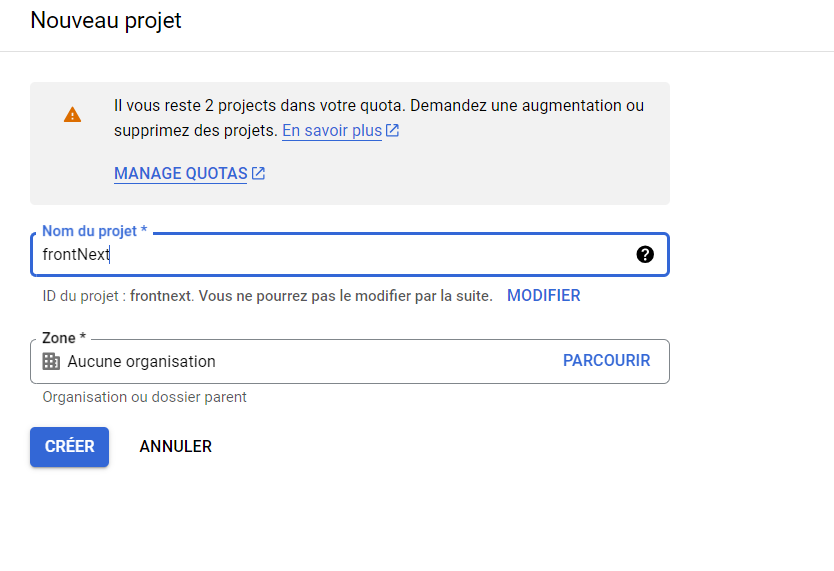
Aller au site :

<https://console.cloud.google.com/>

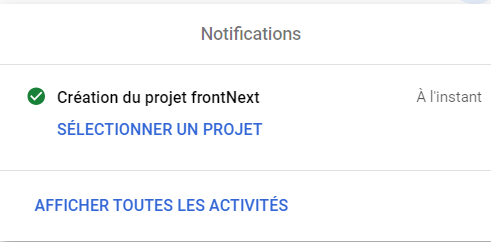




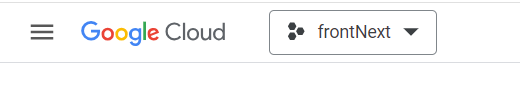




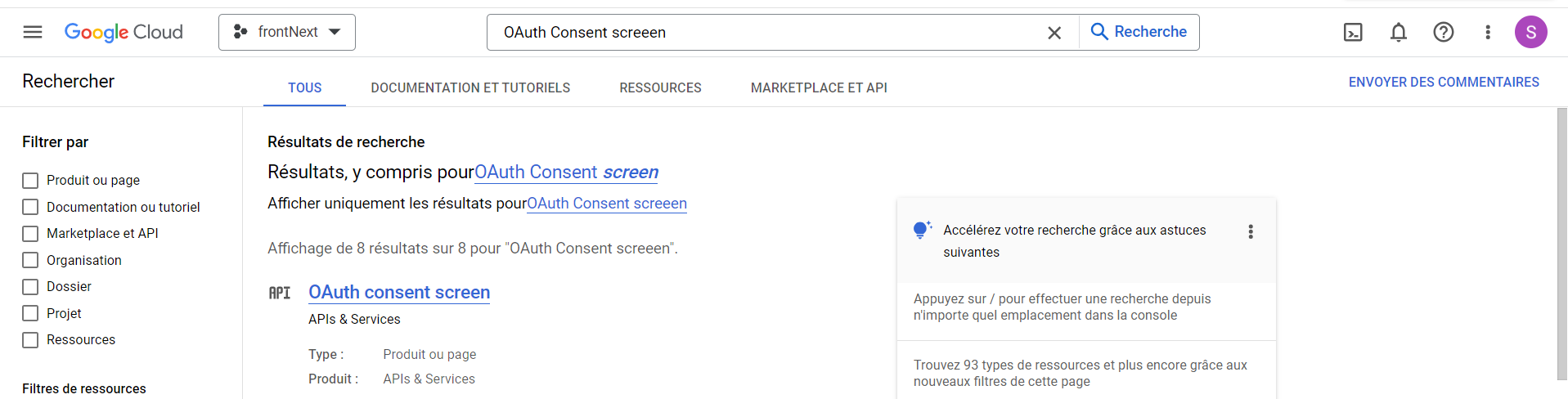


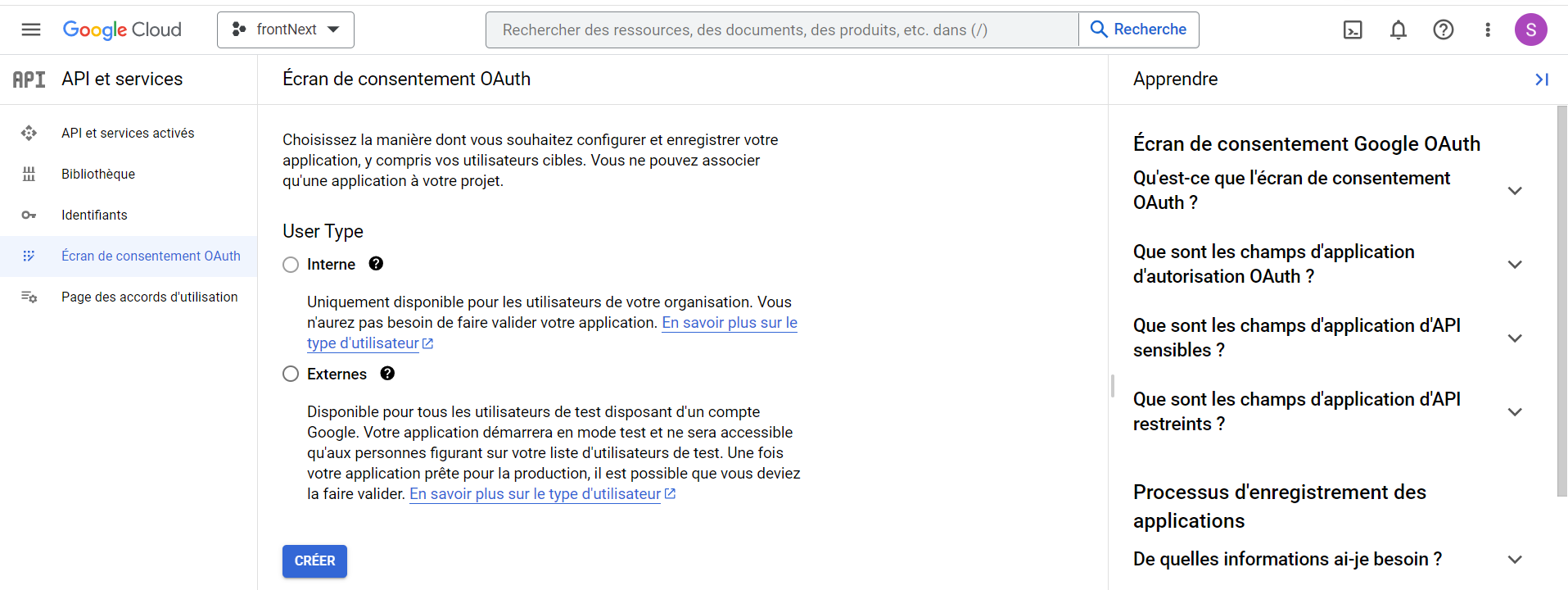


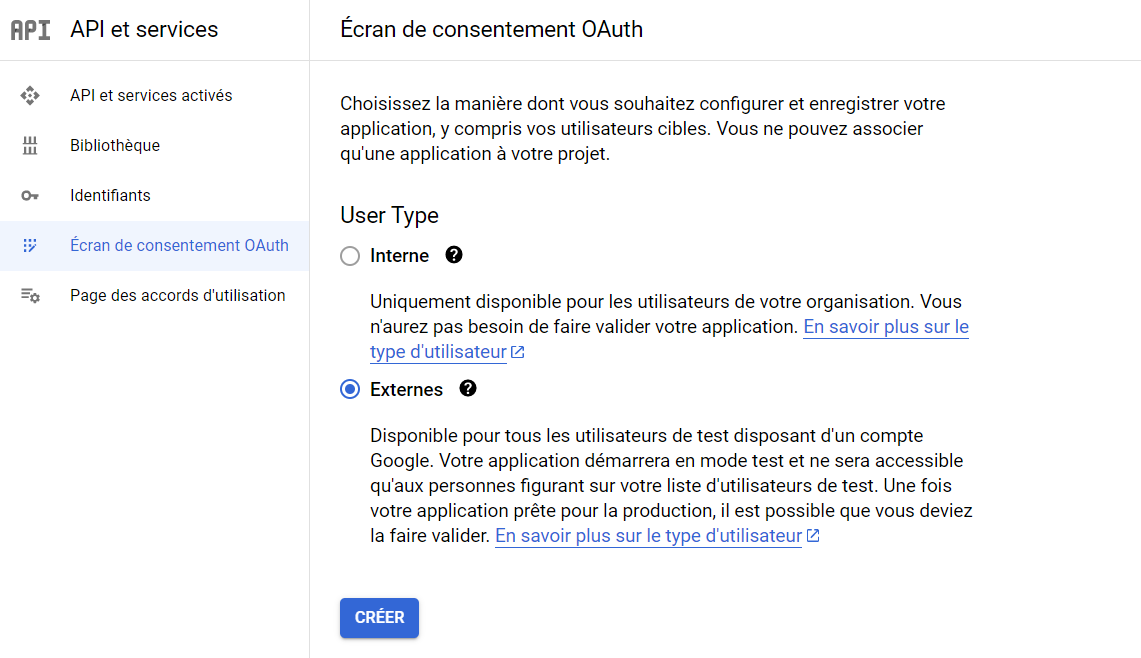


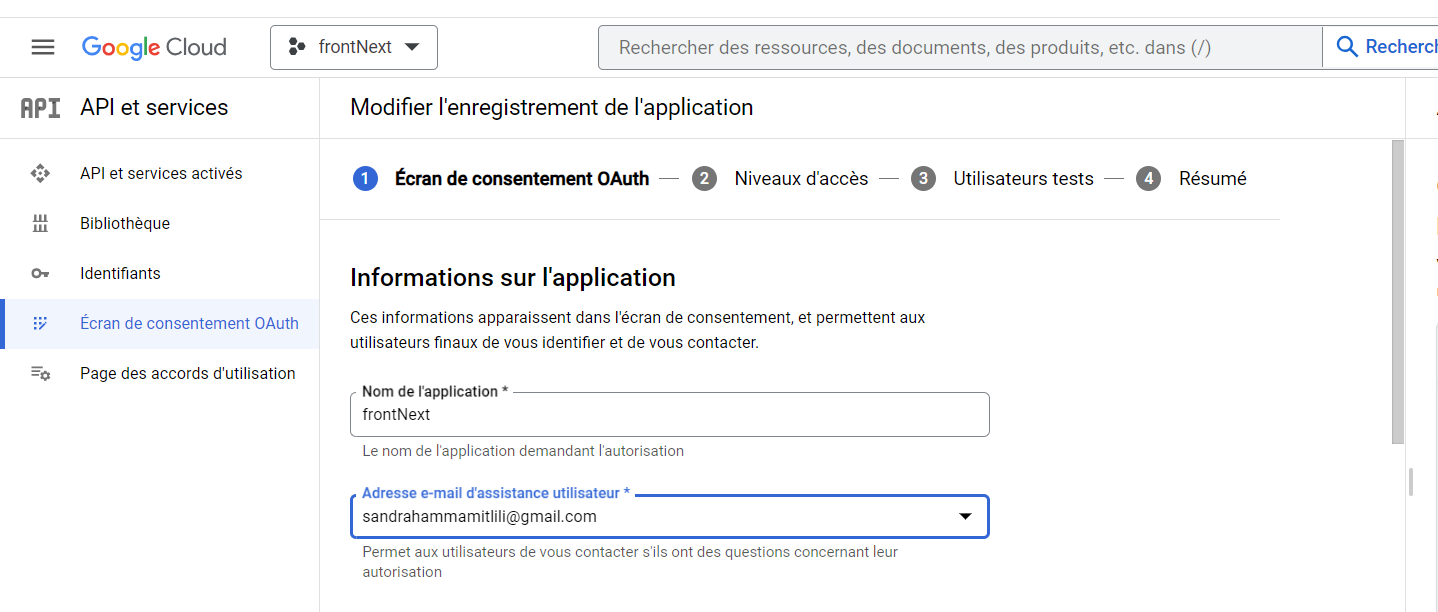


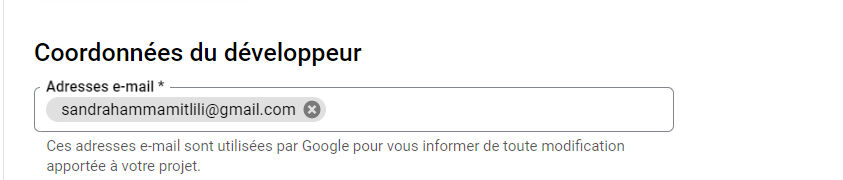




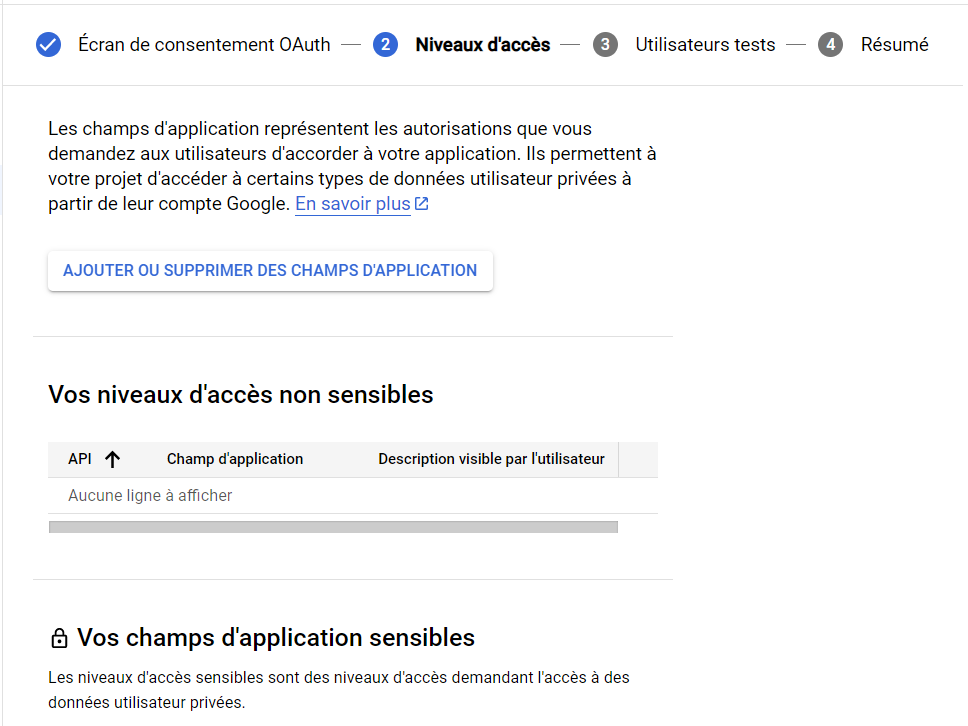


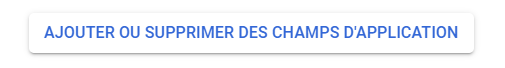


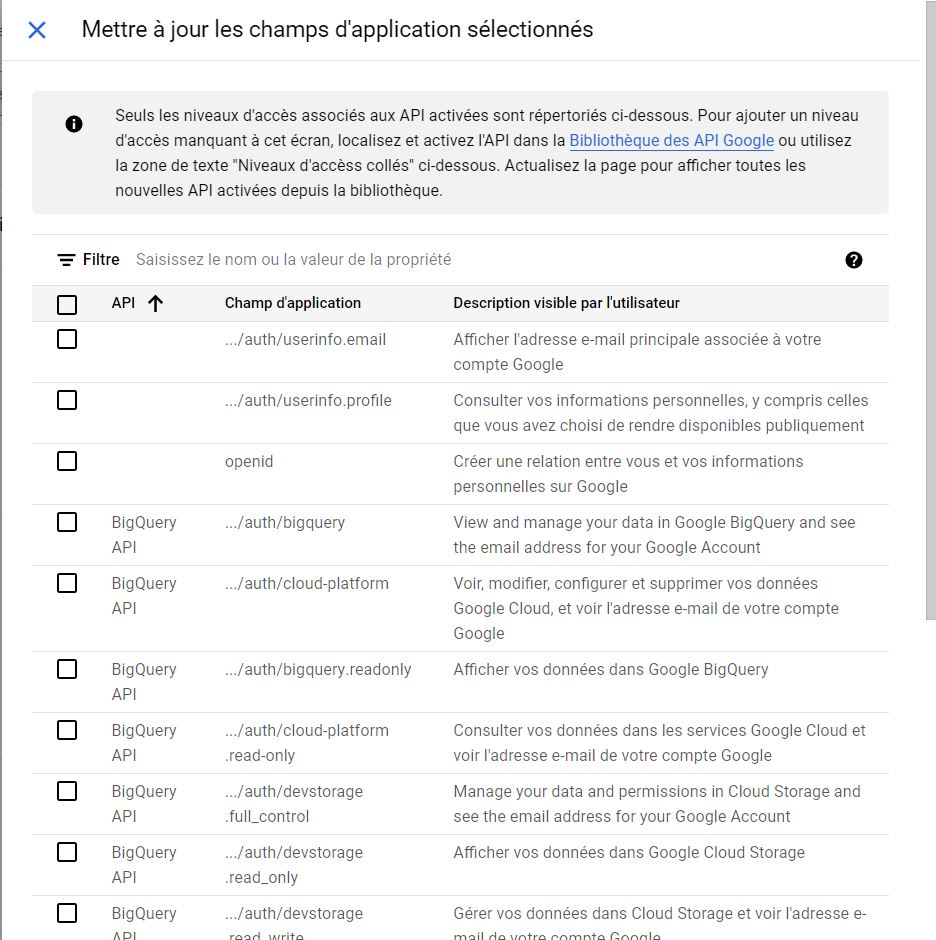


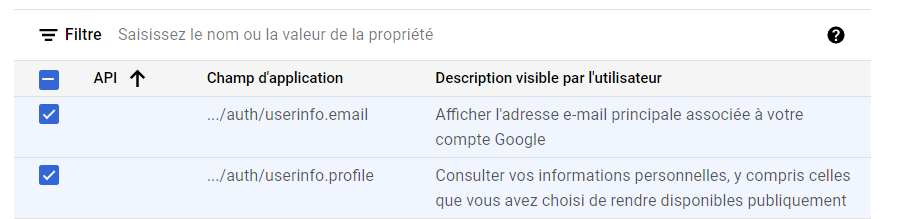


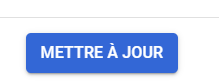


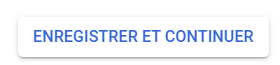


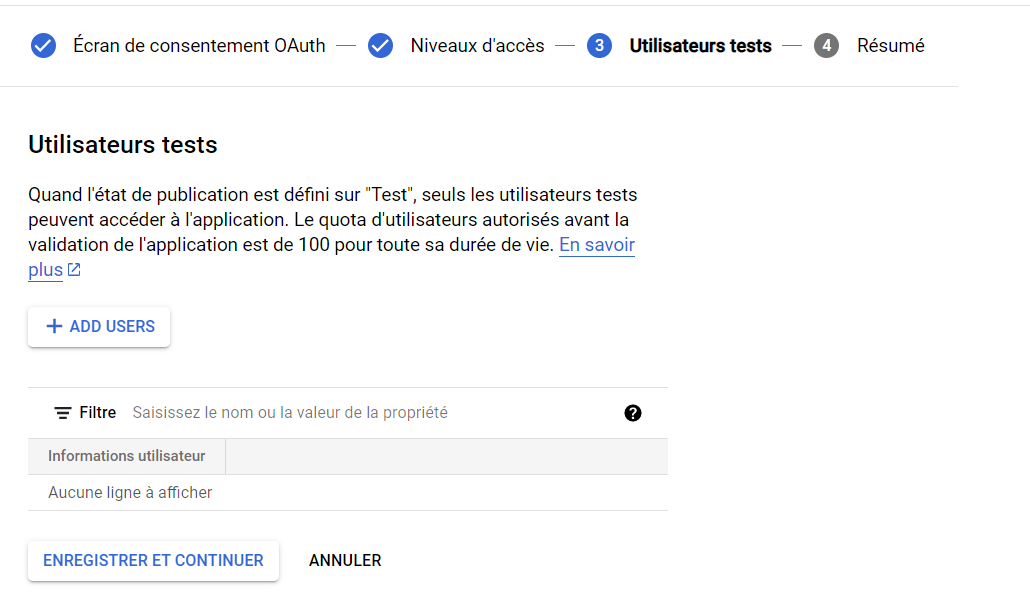


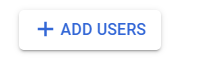


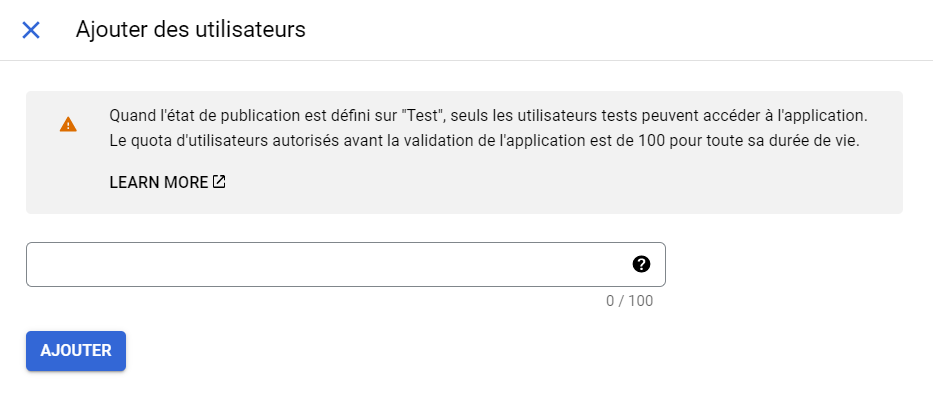








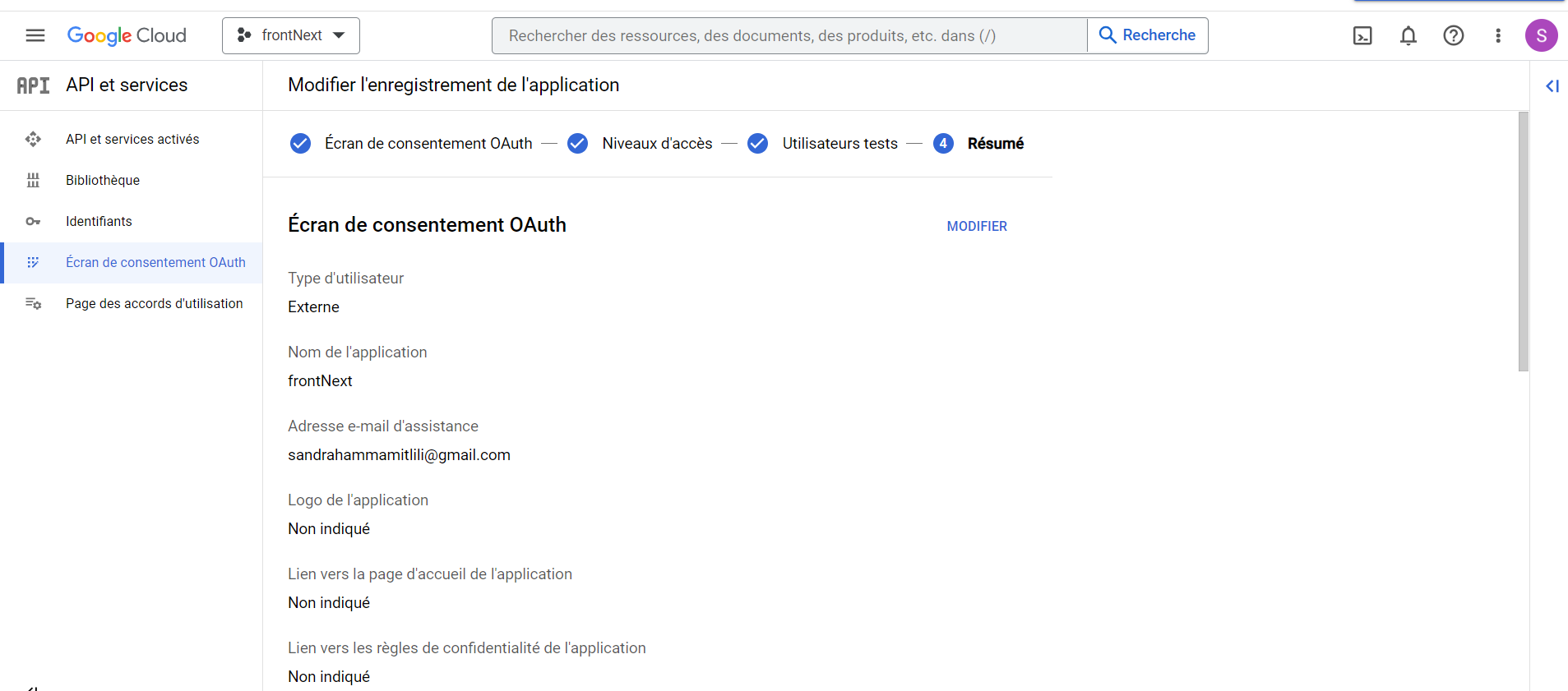


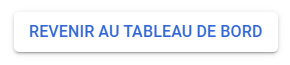


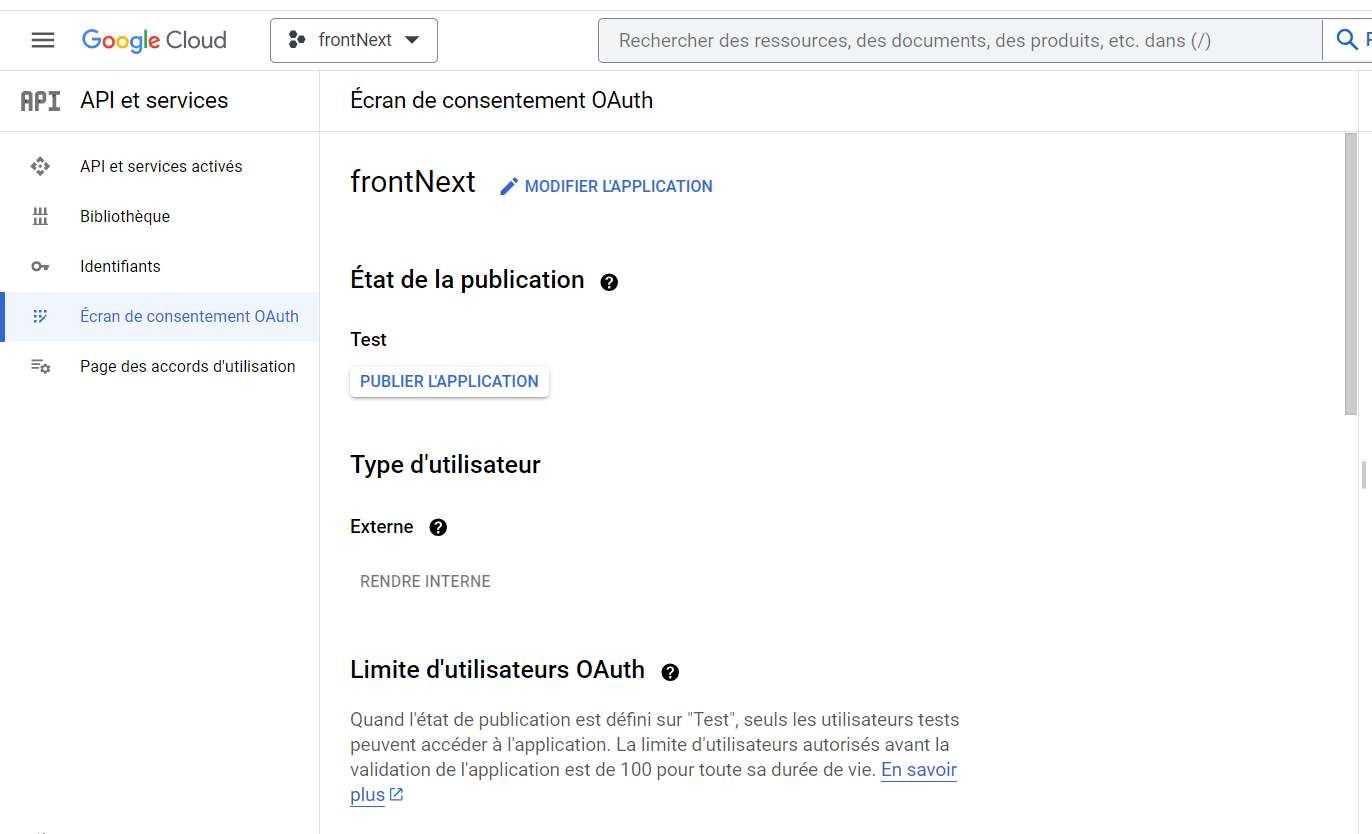




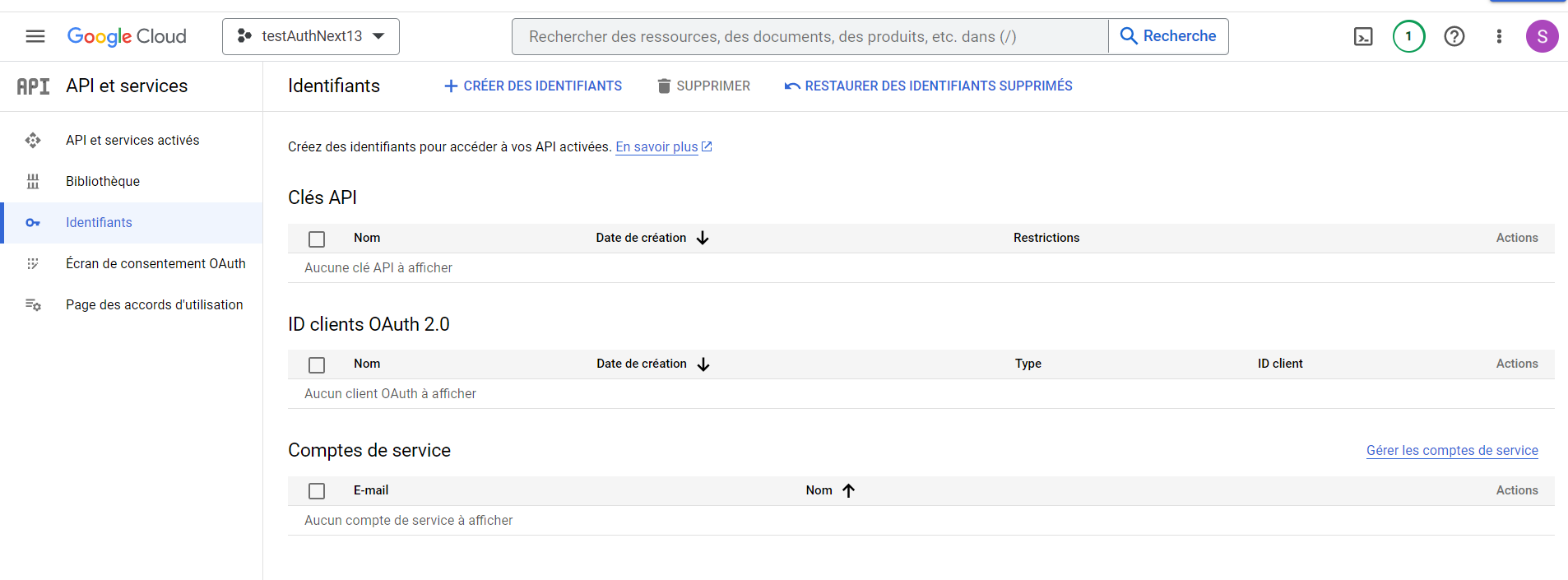




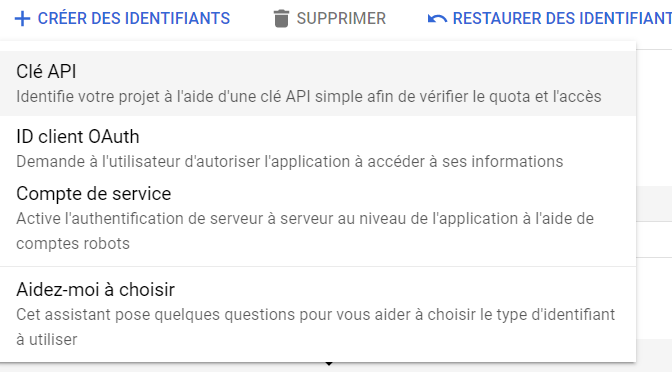




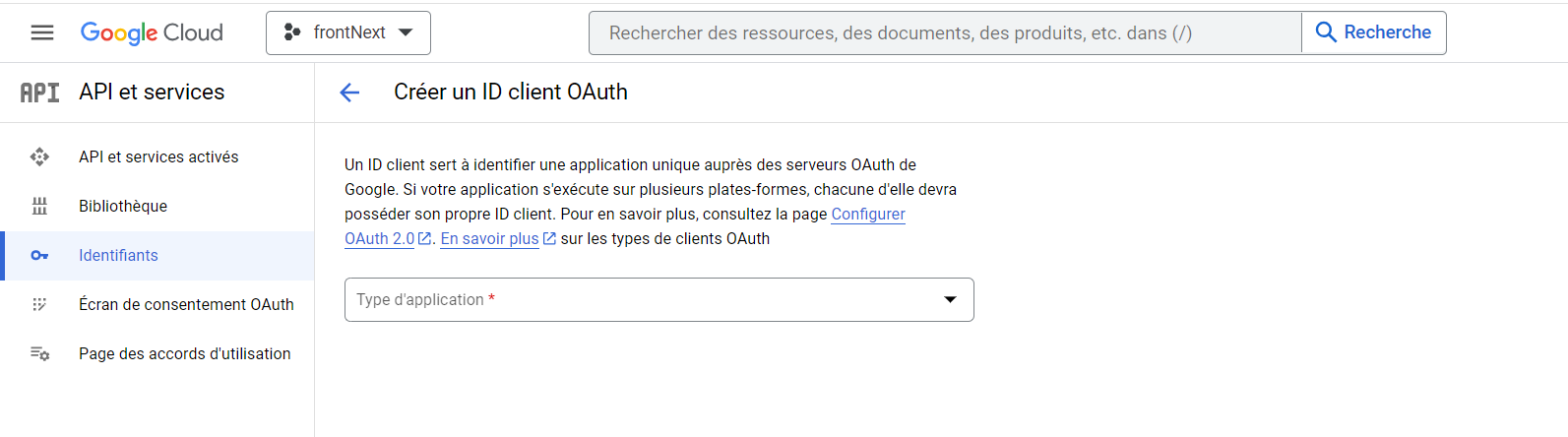




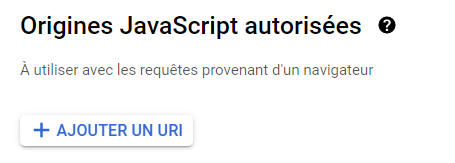


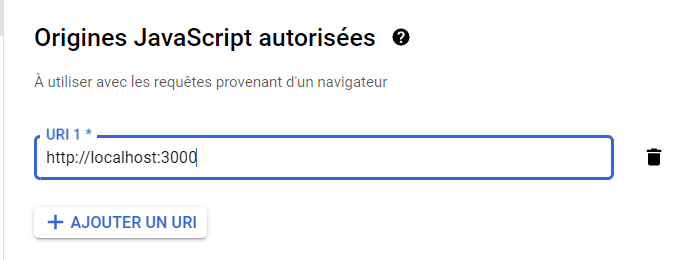






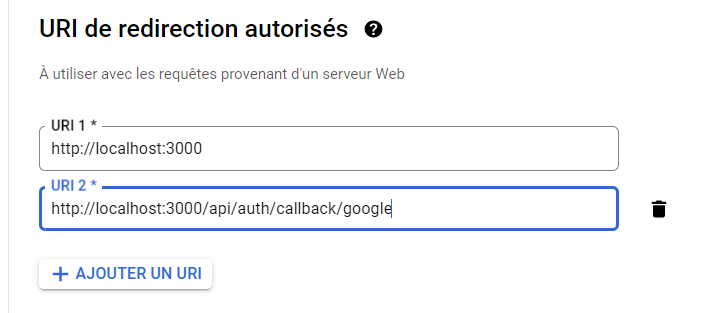




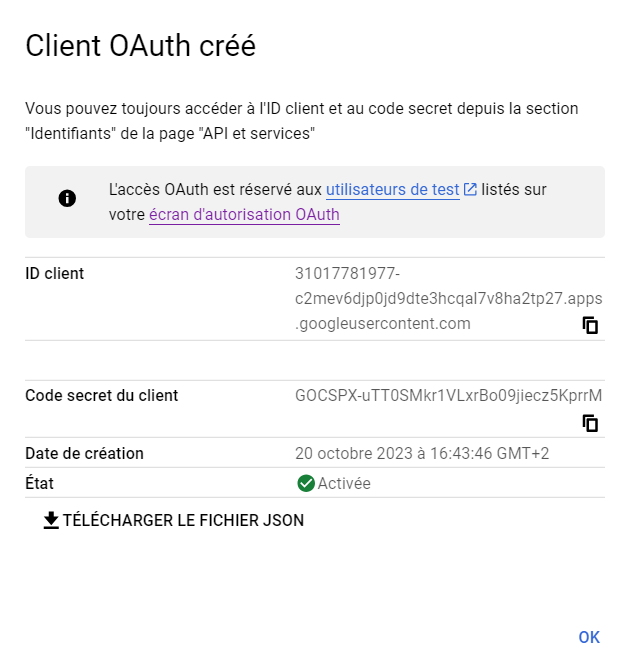










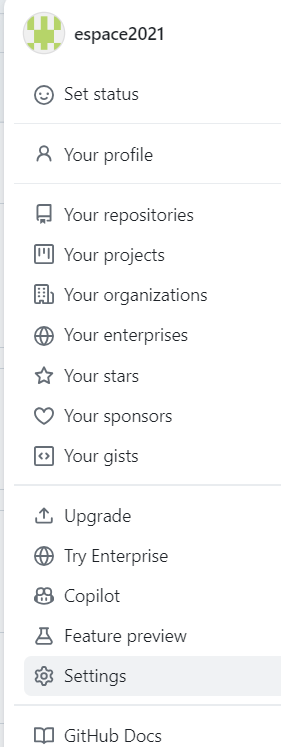


ID client : 31017781977-c2mev6djp0jd9dte3hcqal7v8ha2tp27.apps.googleusercontent.com

Code secret du client : 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

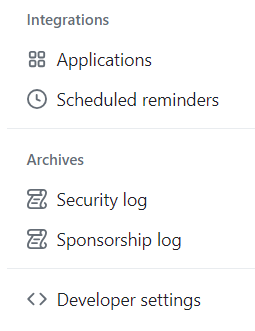
## GitHub Credentials

Connectez-vous à votre tableau de bord Github et cliquez sur l'image de profil dans le coin supérieur droit.

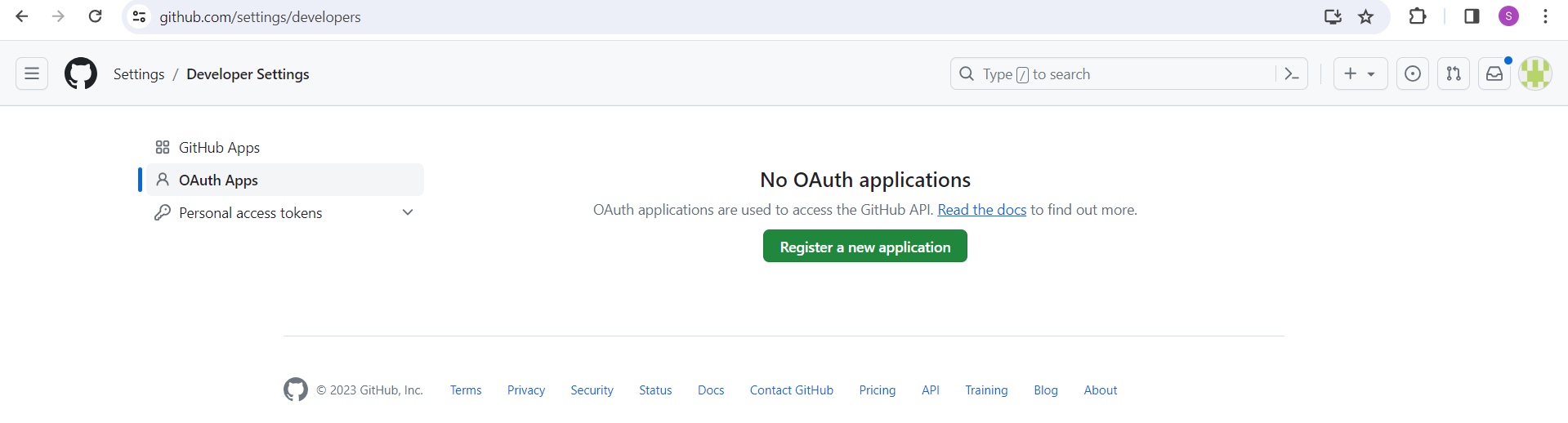


Sélectionnez Settings dans le menu déroulant :

Dans la page qui suit, faites défiler vers le bas et sélectionnez Developer settings :



Cliquez sur OAuth Apps sur la page suivante et choisissez Register a New Application :

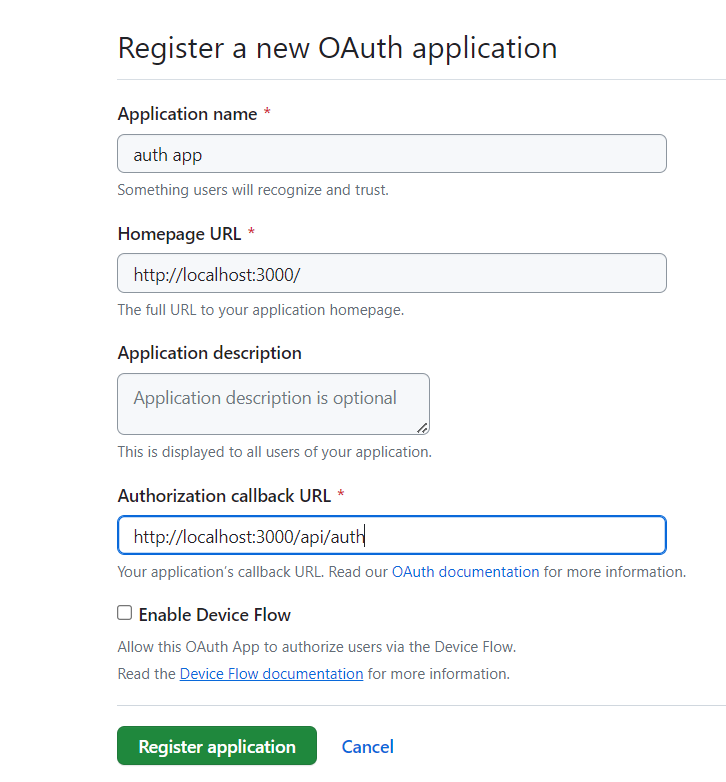


Remplissez le formulaire que vous voyez sur la page qui suit :

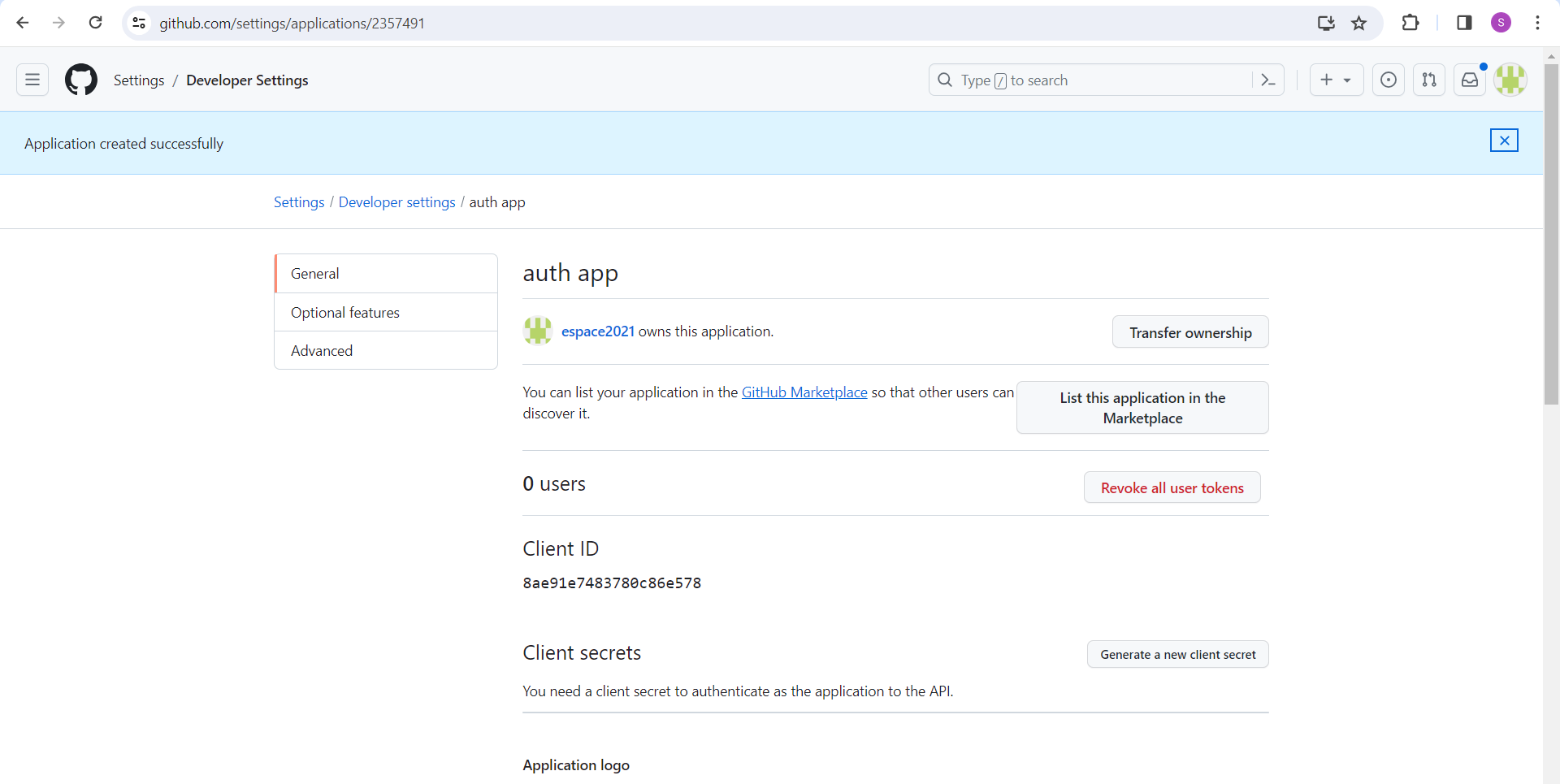
Application name - auth app

Homepage URL - http://localhost:3000/

Authorization callback URL - http://localhost:3000/api/auth



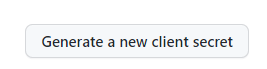
Cliquez sur le bouton Register Application. Une nouvelle page apparaît avec le GITHUB\_ID et le GITHUB\_SECRET :

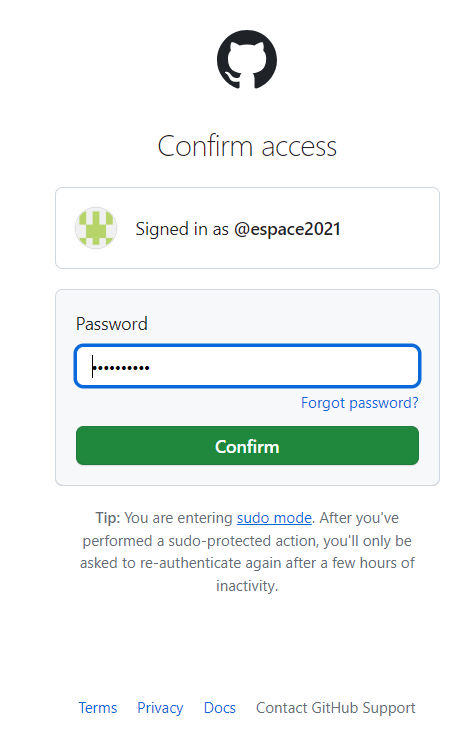


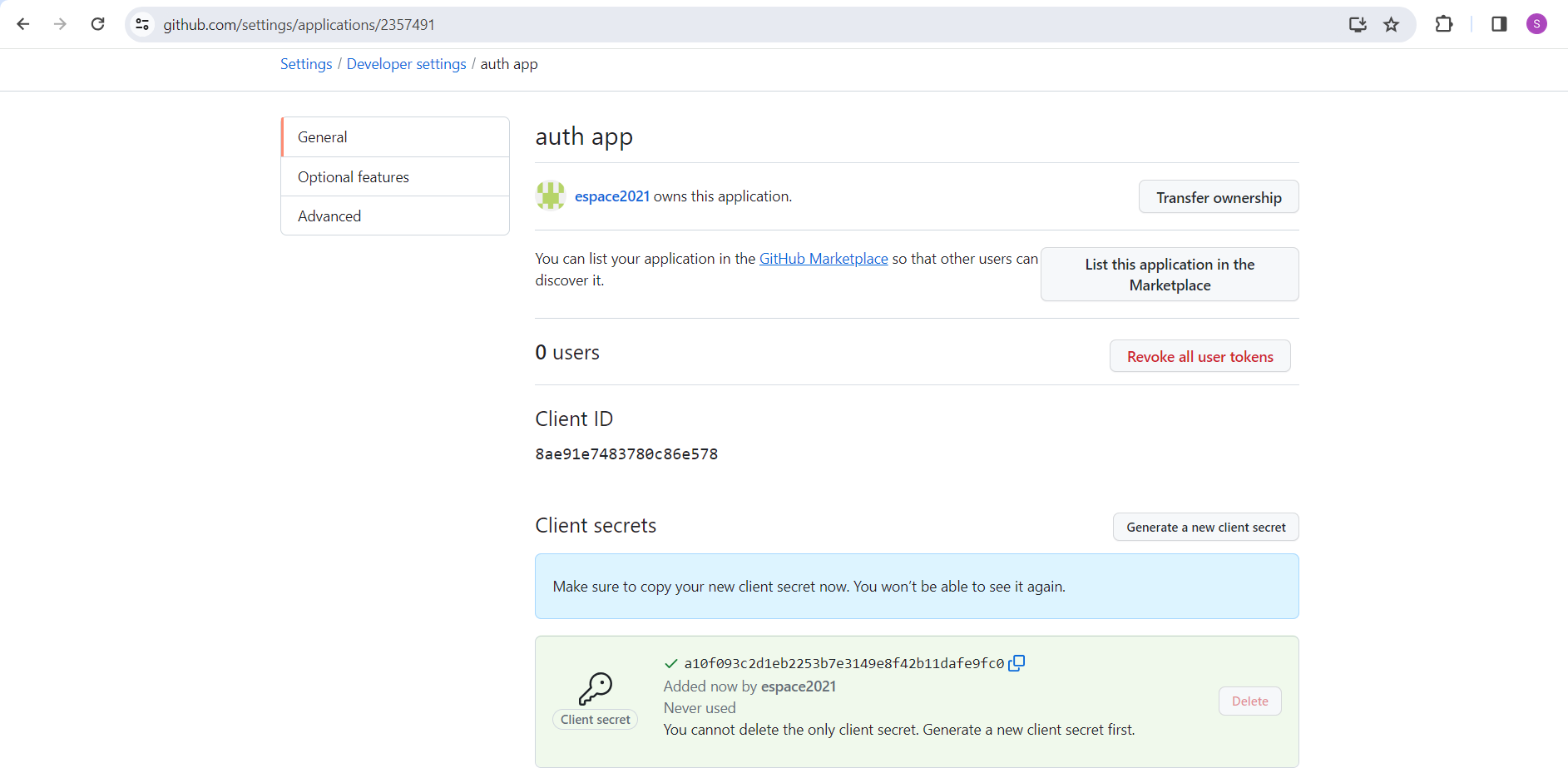
## Client ID

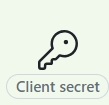
8ae91e7483780c86e578

Cliquez sur Generate a new client secret pour voir le GITHUB\_SECRET.









a10f093c2d1eb2253b7e3149e8f42b11dafe9fc0

## Provider

Dans layout ajouter le code de provider



'use client';

import Navabar from '@/components/client/Navabar'

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

export default function ClientLayout({ children,session }) {

  return (

    <>

    <SessionProvider session={session}>

       <Navabar/>

        {children}

    </SessionProvider>

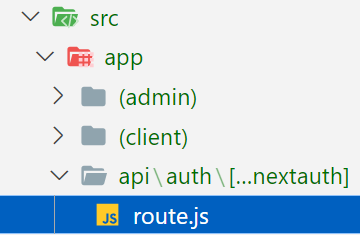
    </>

  )

}

## /app/api/auth/[…nextauth]/route.js

Sous /app créer /api/auth/[…nextauth]/route.js



import NextAuth from "next-auth";

import GoogleProvider from "next-auth/providers/google";

import GitHubProvider from "next-auth/providers/github";

export const authOptions = {

    providers: [

        GitHubProvider({

            clientId: "8ae91e7483780c86e578",

            clientSecret: "a10f093c2d1eb2253b7e3149e8f42b11dafe9fc0",

          }),

        GoogleProvider({

            clientId: "31017781977-c2mev6djp0jd9dte3hcqal7v8ha2tp27.apps.googleusercontent.com",

            clientSecret: "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"

        })

    ],

}

const handler = NextAuth(authOptions)

export { handler as GET, handler as POST }

## Logout

Dans Navbar changer le code pour ajouter l’action logout



'use client';

import Link from "next/link";

import {useSession, signOut} from 'next-auth/react';

const NavBar = () => {

    const {data} =useSession();

return (

<nav className="navbar navbar-expand-lg bg-light">

    <ul className="navbar-nav me-auto mb-2 mb-lg-0">

    <li className="nav-item">

        <Link className="nav-link" href={"/client"}>Home</Link>

    </li>

   <li className="nav-item">

   {data ?  ( <>

      <button className="btn btn-default"  onClick={() => {signOut();}}>Logout </button>

              </>

              ):  null

              }

   </li>

</ul>

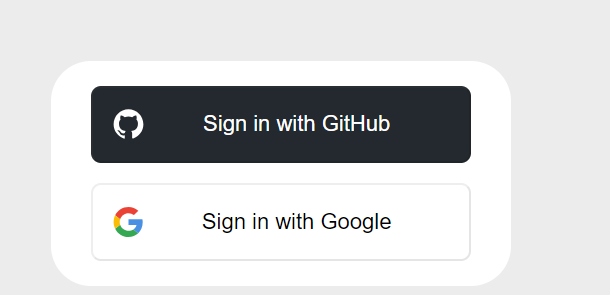
</nav>

);

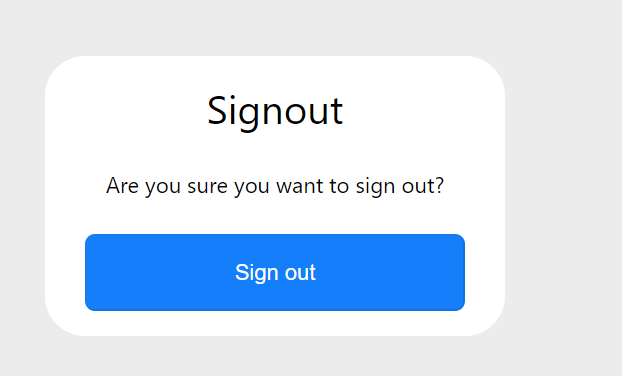
};

export default NavBar;

# <http://localhost:3000/api/auth/signin>



<http://localhost:3000/api/auth/signout>





import React from 'react'

import Link from "next/link";

const page = () => {

  return (

    <div>

      <h1>APPLICATION</h1>

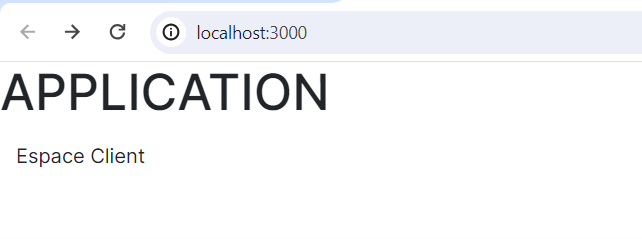
      <Link className="btn btn-default" href={'/client'}> Espace Client </Link>

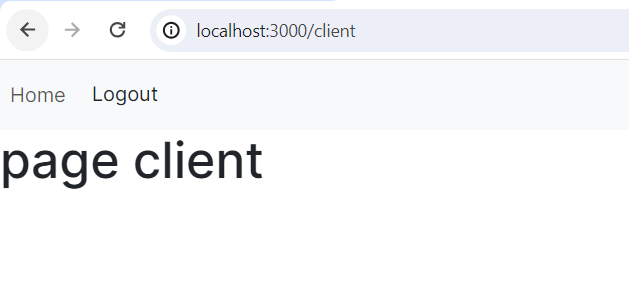
    </div>

  )

}

export default page





# Shopping Cart et protected page

Référence : <https://www.npmjs.com/package/use-shopping-cart>

Exécutez la commande suivante pour installer la bibliothèque :

npm install use-shopping-cart --force

## Ajouter le composant cartProvider

Nous devons envelopper le composant racine de notre application avec <CartProvider/> de use-shopping-cart. Cela nous permettra d'accéder à l'état et à la logique du panier à partir de n'importe quelle page ou composant de notre application. Pour notre projet, le composant racine est **layout** sous le dossier **/app**

'use client';

import Navabar from '@/components/client/Navabar'

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

import { CartProvider } from "use-shopping-cart";

export default function ClientLayout({ children,session }) {

  return (

    <>

    <SessionProvider session={session}>

    <CartProvider

    // Enables local storage

    shouldPersist={true}

  >

       <Navabar/>

        {children}

        </CartProvider>

    </SessionProvider>

    </>

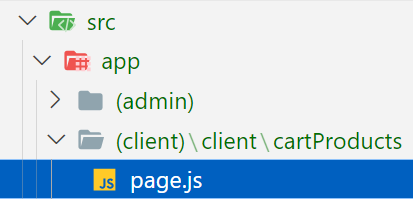
  )

}

shouldPersist est mise à true pour stocker les articles du panier dans le localStorage.

# Page cartProducts

Créer le dossier cartProducts sous le répertoire /client contenant le ficher page.js



import CartProductList from '@/components/client/CartProductList';

import {fetchArticles} from "@/services/ArticlesService"

const getProducts=async()=>{

const {data}=await fetchArticles();

return data

};

const CartProductsPage= async ()=> {

    const products = await getProducts();

    return (

      <>

      <CartProductList products={products} />

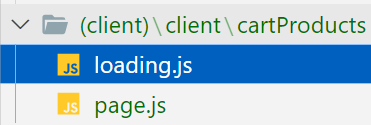
     </>

    )

}

export default CartProductsPage;

Puis créer loading.js



import React from 'react'

const loading = () => {

return (

<div className="d-flex justify-content-center mt-5">

<h1>Loading...</h1>

</div>

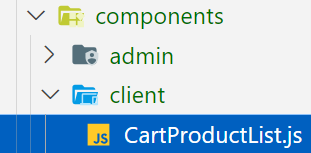
)

}

export default loading

# Component CartProductList

Sous le dossier /components créer le dossier client puis y créer le composant CartProductList.js



const CartProductList=  async ({products})=> {

       return (

        <div className="row">

            {products && products?.map((product) => (

              <article className="col-sm-3">

              <div className="card">

                <img

                  src={product?.imageart}

                  className="card-img-top p-5"

                  alt={product.designation}

                />

              </div>

             <div className="text-center">

             <div>{product.designation}</div>

             <div>Prix : {product.prix} TND </div>

             </div>

              <div className="text-center">

              <button className="btn btn-warning">

                Add to cart

              </button>

              </div>

            </article>

            ))}

          </div>

       );

}

export default CartProductList;

# Ajouter le lien dans le NavBar

Changer le code pour ajouter le lien vers la page shopping cart

'use client';

import Link from "next/link";

import ShoppingBasketIcon from '@mui/icons-material/ShoppingBasket';

import {useSession, signOut} from 'next-auth/react';

const NavBar = () => {

    const {data} =useSession();

return (

<nav className="navbar navbar-expand-lg bg-light">

    <ul className="navbar-nav me-auto mb-2 mb-lg-0">

    <li className="nav-item">

        <Link className="nav-link" href={"/client"}>Home</Link>

    </li>

   <li className="nav-item">

   <Link className="btn btn-default" href={'/client/cartProducts'}><ShoppingBasketIcon style={{ color: 'pink' }}/> Shopping Cart </Link>

   </li>

   <li className="nav-item">

   {data ?  ( <>

      <button className="btn btn-default"  onClick={() => {signOut();}}>Logout </button>

              </>

              ):  null

              }

   </li>

</ul>

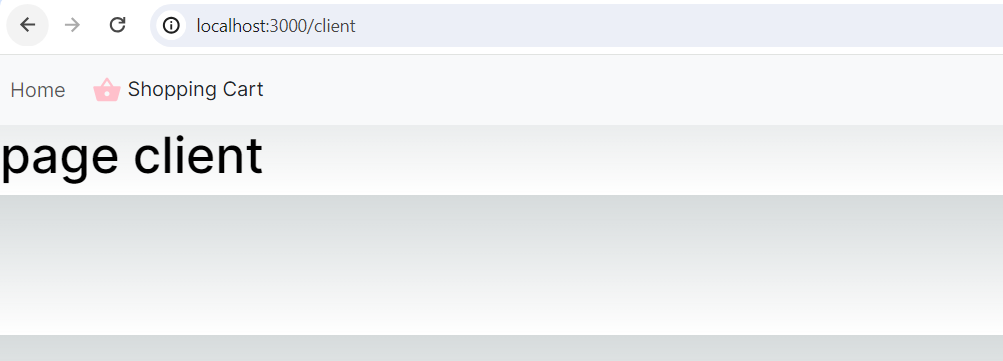
</nav>

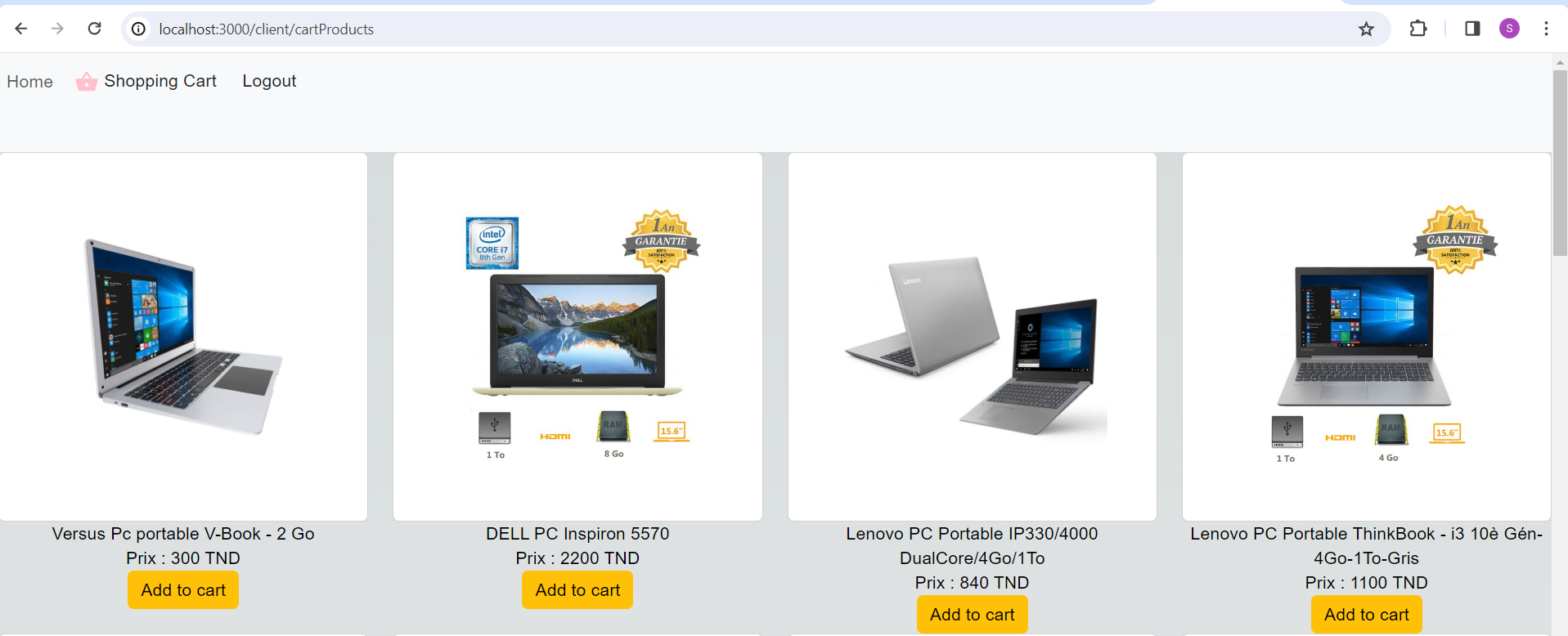
);

};

export default NavBar;

[http://localhost:3000/client/](http://localhost:3000/client/cartProducts)





# Protéger la page avec la session



Ajouter les imports et use client puisqu’on va intégrer au code un hook :

**'use client';**

import { useSession } from 'next-auth/react'

import { redirect } from 'next/navigation'

et le code :

  const { data: session } = useSession({

    required: true,

    onUnauthenticated() {

    redirect('/signin?callbackUrl=/client/cartProducts')

    }

  })

Le code complet devient :

'use client';

import { useSession } from 'next-auth/react'

import { redirect } from 'next/navigation'

const CartProductList=  async ({products})=> {

  const { data: session } = useSession({

    required: true,

    onUnauthenticated() {

    redirect('/signin?callbackUrl=/client/cartProducts')

    }

  })

       return (

        <div className="row">

            {products && products?.map((product) => (

              <article className="col-sm-3">

              <div className="card">

                <img

                  src={product?.imageart}

                  className="card-img-top p-5"

                  alt={product.designation}

                />

              </div>

             <div className="text-center">

             <div>{product.designation}</div>

             <div>Prix : {product.prix} TND </div>

             </div>

              <div className="text-center">

              <button className="btn btn-warning">

                Add to cart

              </button>

              </div>

            </article>

            ))}

          </div>

       );

}

export default CartProductList;

## signIn

Sous le répertoire /src/app/ créer le dossier signin dans le quel créer page.js



Nous avon besoin de cette page pour retrouver l’url de callback et ne pas la perdre (callbackUrl).

'use client';

import { signIn } from 'next-auth/react'

import { useSearchParams } from 'next/navigation'

const SignInPage = () => {

    const searchParams = useSearchParams()

    const callbackUrl = searchParams.get('callbackUrl')

    signIn('resend', { callbackUrl })

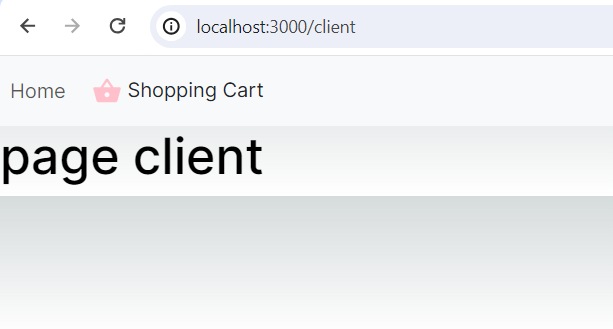
  return (

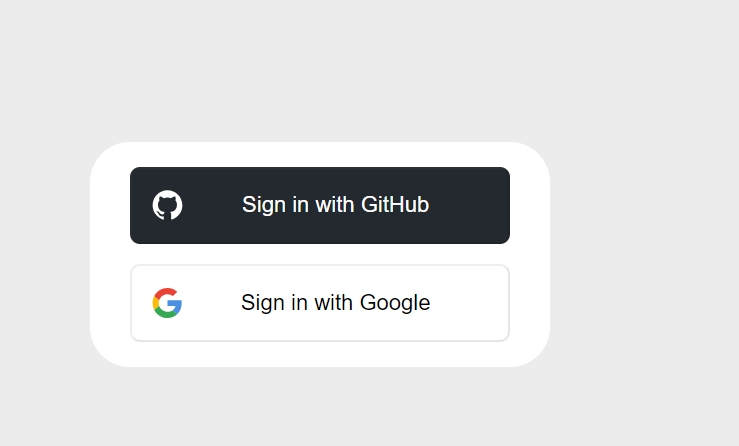
<></>

  )

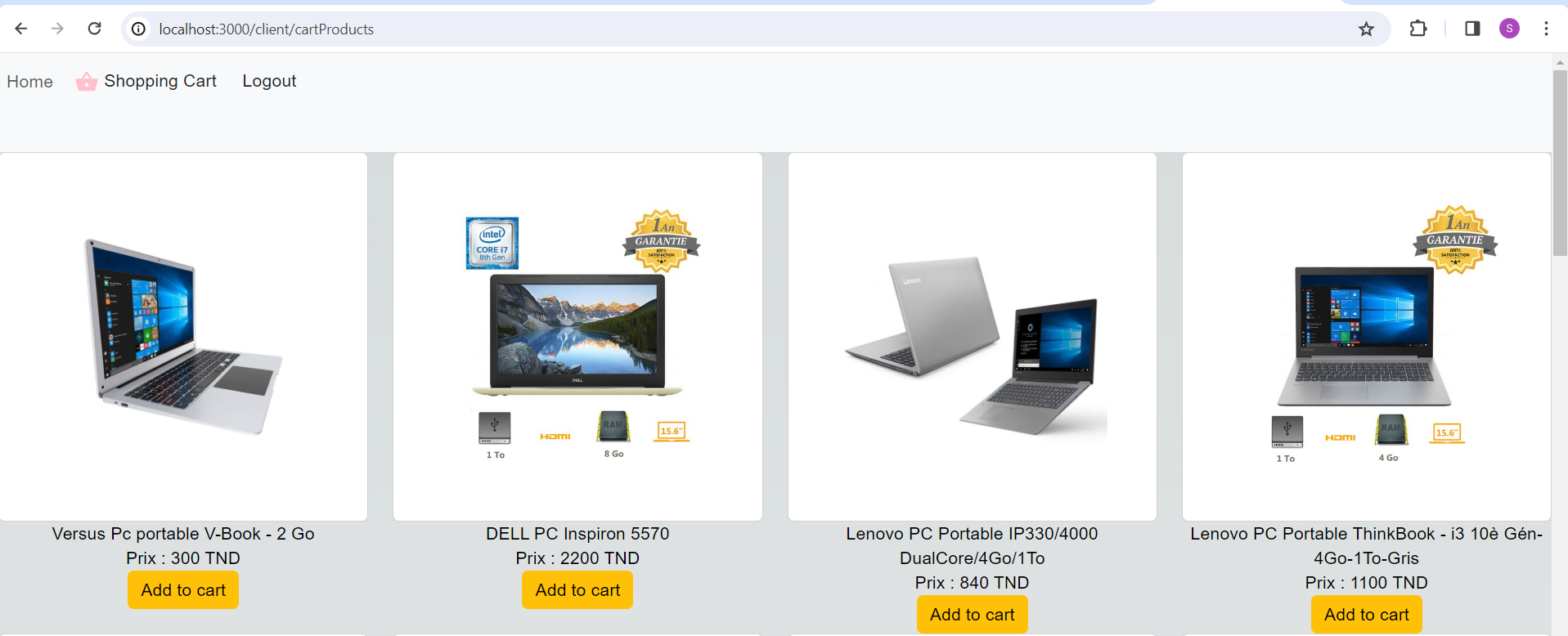
}

export default SignInPage

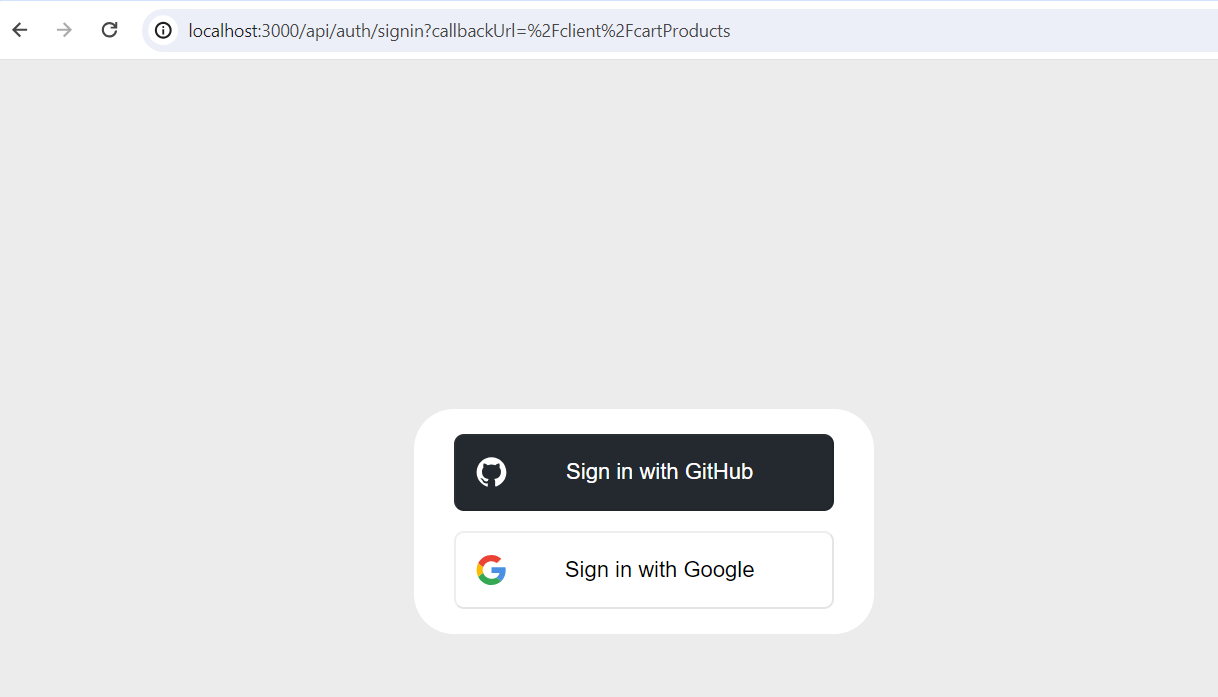




Après authentification par l’un de ces choix, on peut enfin visualiser la page



Si on clique sur logout on aura de nouveau la page signin





'use client';

import Link from "next/link";

import ShoppingBasketIcon from '@mui/icons-material/ShoppingBasket';

import {useSession, signOut} from 'next-auth/react';

import Image from "next/image";

const NavBar = () => {

    const {data} =useSession();

return (

<nav className="navbar navbar-expand-lg bg-light">

    <ul className="navbar-nav me-auto mb-2 mb-lg-0">

    <li className="nav-item">

        <Link className="nav-link" href={"/client"}>Home</Link>

    </li>

   <li className="nav-item">

   <Link className="btn btn-default" href={'/client/cartProducts'}><ShoppingBasketIcon style={{ color: 'pink' }}/> Shopping Cart </Link>

   </li>

   <li className="nav-item">

   {data ?  ( <>

      <button className="btn btn-default"  onClick={() => {signOut();}}>Logout </button>

      <button className="btn btn-default">

                <Image

                  src="/images/cart.png"

                  width={40}

                  height={40}

                  alt="shopping cart icon"

                />

                 <span className="badge bg-primary">0</span>

        </button>

              </>

              ):  null

              }

   </li>

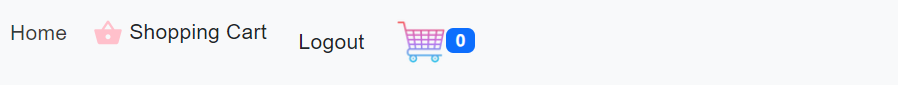
</ul>

</nav>

);

};

export default NavBar;



Maintenant on veut intégrer la méthode addToCart lorsqu’on clique sur le bouton. Pour cela on va faire appel à la méthode addItem de la bibliothèque de use-shopping-cart.



'use client';

import { useSession } from 'next-auth/react'

import { redirect } from 'next/navigation'

import { useShoppingCart } from "use-shopping-cart" ;

const CartProductList=  async ({products})=> {

  const { addItem } = useShoppingCart();

  const { data: session } = useSession({

    required: true,

    onUnauthenticated() {

    redirect('/signin?callbackUrl=/client/cartProducts')

    }

  })

const addToCart = (product) => {

      const target = {

      id:product.\_id,

      title:  product.designation,

      image: product.imageart,

      price : product.prix,

      qtestock : product.qtestock,

      quantity : 1

      };

            addItem(target);

      console.log('Item added to cart:', target);

    };

       return (

        <div className="row">

            {products && products?.map((product) => (

              <article className="col-sm-3">

              <div className="card">

                <img

                  src={product?.imageart}

                  className="card-img-top p-5"

                  alt={product.designation}

                />

              </div>

             <div className="text-center">

             <div>{product.designation}</div>

             <div>Prix : {product.prix} TND </div>

             </div>

              <div className="text-center">

              <button className="btn btn-warning" onClick={() => addToCart(product)}>

                Add to cart

              </button>

              </div>

            </article>

            ))}

          </div>

       );

}

export default CartProductList;

Il faut ajouter le provider dans le layout principal.



import { Inter } from 'next/font/google'

import './globals.css'

import 'bootstrap/dist/css/bootstrap.css'

import '@fortawesome/fontawesome-free/css/all.min.css';

import { CartProvider } from "use-shopping-cart";

const inter = Inter({ subsets: ['latin'] })

 export const metadata = {

  title: 'Create Next App',

  description: 'Generated by create next app',

}

export default function RootLayout({ children }) {

  return (

    <html lang="en">

      <body className={inter.className}>

      <CartProvider

    // Enables local storage

    shouldPersist={true}

  >

       {children}

       </CartProvider>

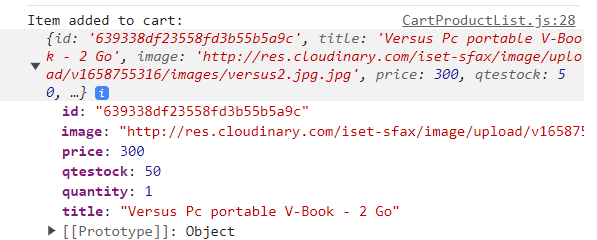
        </body>

    </html>

  )

}





On va préparer la page Shopping cart qui contiendra tous les éléments sélectionnés dans le cart pour cela créer :



'use client';

import React from 'react';

import { useShoppingCart } from 'use-shopping-cart';

import { useRouter} from 'next/navigation';

import Grid from '@mui/material/Grid';

import Button from '@mui/material/Button';

import Plus from '@mui/icons-material/AddAlarm';

import Minus from '@mui/icons-material/RemoveCircle';

import Delete from '@mui/icons-material/Delete';

const ShoppingCart = () => {

  const { cartDetails, removeItem , clearCart, totalPrice, cartCount,incrementItem,decrementItem } = useShoppingCart();

  const router = useRouter();

  const commander = () => {

  };

  const more = () => {

    router.push('/client/cartProducts');

  };

  const clear = () => {

    clearCart();

  };

  if (cartCount === 0) return <h1>Cart Empty</h1>;

  return (

    <div>

      <Grid container spacing={2} columns={15} marginTop={10} marginLeft={10}>

        <Grid item xs={8}>

          {cartDetails && Object.values(cartDetails).map((item) => {

            return (

              <Grid item xs={8} key={item.id}>

                <img

                  alt={item.title}

                  style={{ margin: "0 auto", maxHeight: "100px" }}

                  src={item.image}

                  className="img-fluid d-block"

                />

                <h5>{item.title}</h5>

                <p>Prix: {item.price} TND</p>

                <p>Qté: {item.quantity}</p>

                <button

                  onClick={() => {

                    if (item.quantity < item.qtestock) {

                      incrementItem(item.id);

                    } else {

                      alert("Quantité stock indisponible");

                    }

                  }}

                >

                  <Plus color="success" />

                </button>

                {item.quantity > 1 && (

                  <button

                  onClick={() => decrementItem(item.id)}

                  >

                    <Minus color="warning" />

                  </button>

                )}

                {item.quantity === 1 && (

                  <button onClick={() => removeItem(item.id)}>

                    <Delete color="error" />

                  </button>

           ) }

                <hr />

              </Grid>

            );

          })}

        </Grid>

        <Grid item xs={5}>

          <Button color="error" variant="outlined" onClick={more}>

            Ajouter des articles

          </Button>

          <p>Total Articles</p>

          <h4>{cartCount}</h4>

          <p>Total Payement</p>

          <h3>{totalPrice} TND</h3>

          <hr />

          <div>

            <Button color="warning" variant="outlined" onClick={commander}>

              Commander

            </Button>

            <Button color="info" variant="outlined" onClick={clear}>

              Annuler

            </Button>

          </div>

        </Grid>

      </Grid>

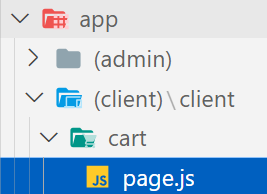
    </div>

  );

};

export default ShoppingCart;

sous /app/(client)/client créer le dossier cart ayant page.js



import React from 'react'

import ShoppingCartPage from '@/components/client/ShoppingCart'

const CartPage = () => {

  return (

    <div>

      <ShoppingCartPage />

    </div>

  )

}

export default CartPage



'use client';

import Link from "next/link";

import ShoppingBasketIcon from '@mui/icons-material/ShoppingBasket';

import {useSession, signOut} from 'next-auth/react';

import Image from "next/image";

import { useShoppingCart } from "use-shopping-cart" ;

const NavBar = () => {

    const {cartCount} = useShoppingCart();

    const {data} =useSession();

return (

<nav className="navbar navbar-expand-lg bg-light">

    <ul className="navbar-nav me-auto mb-2 mb-lg-0">

    <li className="nav-item">

        <Link className="nav-link" href={"/client"}>Home</Link>

    </li>

   <li className="nav-item">

   <Link className="btn btn-default" href={'/client/cartProducts'}><ShoppingBasketIcon style={{ color: 'pink' }}/> Shopping Cart </Link>

   </li>

   <li className="nav-item">

   {data ?  ( <>

      <button className="btn btn-default"  onClick={() => {signOut();}}>Logout </button>

      <button className="btn btn-default" >

**<Link href={"/client/cart"}>**

                <Image

                  src="/images/cart.png"

                  width={40}

                  height={40}

                  alt="shopping cart icon"

                />

                 <span className="badge bg-primary">**{cartCount}**</span>

**</Link>**

        </button>

              </>

              ):  null

              }

   </li>

</ul>

</nav>

);

};

export default NavBar;

