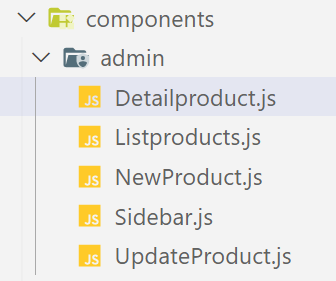
Admin side

Créer l’arborescence suivante :

ATELIER 2 : NextJS : Admin et CRUDS ARTICLE





npm install axios bootstrap react-bootstrap

ajouter la ligne suivante dans le fichier layout.js

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

créer dans le dossier components/admin un composant dont le nom est NavScroll.jsx

pour cela accéder au site https://react-bootstrap.github.io/

appuyer sur Get Started et Components Navbar

composant NavScroll.jsx

"use client"

import Container from 'react-bootstrap/Container';

import Nav from 'react-bootstrap/Nav';

import Navbar from 'react-bootstrap/Navbar';

import NavDropdown from 'react-bootstrap/NavDropdown';

import Link from 'next/link';

function NavScroll

() {

  return (

    <Navbar expand="lg" className="bg-body-tertiary">

      <Container>

        <Navbar.Brand href="#home">React-Bootstrap</Navbar.Brand>

        <Navbar.Toggle aria-controls="basic-navbar-nav" />

        <Navbar.Collapse id="basic-navbar-nav">

          <Nav className="me-auto">

          <Nav.Link as={Link} href="/">

            Home

            </Nav.Link>

          <Nav.Link as={Link} href="/admin/products">

            Articles

            </Nav.Link>

            <Nav.Link href="#link">Link</Nav.Link>

            <NavDropdown title="Dropdown" id="basic-nav-dropdown">

              <NavDropdown.Item href="#action/3.1">Action</NavDropdown.Item>

              <NavDropdown.Item href="#action/3.2">

                Another action

              </NavDropdown.Item>

              <NavDropdown.Item href="#action/3.3">Something</NavDropdown.Item>

              <NavDropdown.Divider />

              <NavDropdown.Item href="#action/3.4">

                Separated link

              </NavDropdown.Item>

            </NavDropdown>

          </Nav>

        </Navbar.Collapse>

      </Container>

    </Navbar>

  );

}

export default NavScroll;

# Fichier app/(admin)/admin/layout.js



import NavScroll from '@/components/admin/NavScroll'

export default function RootLayout({ children }) {

  return (

    <>

       <NavScroll/>

        {children}

    </>

  )

}

# Axios

Installer

npm i axios

Créer



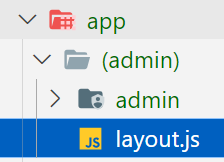
import axios from 'axios'

export default axios.create({

baseURL:"http://localhost:3001/api"

})

# Layout



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

import './globals.css'

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

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

 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}>

        {children}

        </body>

    </html>

  )

}

# Services

Créer



import Api from "../Axios/Api";

const ARTICLE\_API="/articles"

export const fetchArticles=async()=> {

return await Api.get(ARTICLE\_API);

}

export const fetchArticleById=async(articleId)=> {

return await Api.get(ARTICLE\_API + '/' + articleId);

}

export const deleteArticle=async(articleId) =>{

return await Api.delete(ARTICLE\_API + '/' + articleId);

}

export const addArticle=async(article)=> {

return await Api.post(ARTICLE\_API, article);

}

export const editArticle=(article) =>{

return Api.put(ARTICLE\_API + '/' + article.\_id, article);

}



import Api from "../Axios/Api";

const SCATEGORIE\_API="/scategories"

export const fetchSCategories=async()=> {

return await Api.get(SCATEGORIE\_API);

}

export const fetchSCategorieById=async(scategorieId)=> {

return await Api.get(SCATEGORIE\_API + '/' + scategorieId);

}

export const deleteSCategorie=async(scategorieId) =>{

return await Api.delete(SCATEGORIE\_API + '/' + scategorieId);

}

export const addSCategorie=async(scategorie)=> {

return await Api.post(SCATEGORIE\_API,scategorie);

}

export const editSCategorie=(scategorie) =>{

return Api.put(SCATEGORIE\_API + '/' + scategorie.\_id, scategorie);

}

***1./Afficher la liste des produits***

App/admin/products/page.js

import Listproducts from '@/components/admin/Listproducts';

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

const getProducts=async()=>{

const {data}=await fetchArticles()

return data;

}

const ProductPage = async() =>{

    const produits=await getProducts()

  return (

    <div>

      <Listproducts produits={produits}/>

    </div>

  )

}

export default ProductPage

Créer maintenant le composant Listproducts.jsx pour afficher la liste des articles

Accéder au site[**https://www.material-react-table.com/**](https://www.material-react-table.com/)

**Installer le composant material-react-table**

**npm install material-react-table @mui/material @mui/icons-material @emotion/react @emotion/styled**

Quick examples->Basic Example

"use client"

import  { useMemo } from 'react';

import { MaterialReactTable } from 'material-react-table';

import { Box } from '@mui/material';

import Image from "next/image"

const Listproducts = ({produits}) => {

  const columns = useMemo(

    () => [

    {

    accessorKey: 'imageart', //access nested data with dot notation

    header: 'Image',

    Cell: ({ cell}) => (

    <Box

    sx={{

    display: 'flex',

    alignItems: 'center',

    gap: '1rem',

    }}

    >

    <Image

              src={cell.getValue()}

              alt="product anme"

              height="50"

              width="50"

            />

    </Box>),

    },

    {

    accessorKey: 'reference', //access nested data with dot notation

    header: 'Référence',

    size: 100,

    },

    {

    accessorKey: 'designation',

    header: 'Désignation',

    size: 100,

    },

    {

    accessorKey: 'marque', //normal accessorKey

    header: 'Marque',

    size: 100,

    },

    {

    accessorKey: 'prix',

    header: 'Prix',

    size: 100,

    },

    {

    accessorKey: 'qtestock',

    header: 'Stock',

    size: 100,

    },

    ],

    [produits],

    );

return (

<div>

<MaterialReactTable columns={columns} data={produits} />;

</div>

  )

}

export default Listproducts

installer fontawesome

npm i @fortawesome/fontawesome-free

dans le fichier layout.js de app

Dans fichier layout.js ajouter l’appel a fontawesome

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

import './globals.css'

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

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

import NavScroll from '@/components/admin/NavScroll'

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

 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}>

       <NavScroll/>

        {children}

        </body>

    </html>

  )

}

# Suppression d’un produit

Composant Listproducts.js

"use client"

import  { useMemo } from 'react';

import { MaterialReactTable } from 'material-react-table';

import { Box } from '@mui/material';

import Image from "next/image"

import Button from 'react-bootstrap/Button';

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

import { useRouter } from "next/navigation";

const Listproducts = ({produits}) => {

  const router = useRouter();

  const deletearticle=(id)=>{

    if(window.confirm("supprimer le produit O/N")) {

      deleteArticle(id)

      .then((res)=>{ console.log(res)

       router.refresh()

      })

      .catch(error=>{

          console.log(error)

         })

}

      }

  const columns = useMemo(

    () => [

    {

    accessorKey: 'imageart', //access nested data with dot notation

    header: 'Image',

    Cell: ({ cell}) => (

    <Box

    sx={{

    display: 'flex',

    alignItems: 'center',

    gap: '1rem',

    }}

    >

    <Image

              src={cell.getValue()}

              alt="product anme"

              height="50"

              width="50"

            />

    </Box>),

    },

    {

    accessorKey: 'reference', //access nested data with dot notation

    header: 'Référence',

    size: 100,

    },

    {

    accessorKey: 'designation',

    header: 'Désignation',

    size: 100,

    },

    {

    accessorKey: 'marque', //normal accessorKey

    header: 'Marque',

    size: 100,

    },

    {

    accessorKey: 'prix',

    header: 'Prix',

    size: 100,

    },

    {

    accessorKey: 'qtestock',

    header: 'Stock',

    size: 100,

    },

    {

      accessorKey: '\_id',

      header: 'actions',

      size: 100,

      Cell: ({ cell, row }) => (

      <div >

      <Button

      onClick={() => {

      console.log("modification ...")

      }}

      variant="warning"

      size="md"

      className="text-warning btn-link edit"

      >

      <i class="fa-solid fa-pen-to-square"></i>

      </Button>

      <Button

      onClick={(e) => {

      deletearticle(cell.row.original.\_id,e);

      }}

      variant="danger"

      size="md"

      className="text-danger btn-link delete"

      >

      <i className="fa fa-trash" />

      </Button>

      </div>

      ),

      },

    ],

  [produits],

  )

return (

<div>

{ produits && (

<MaterialReactTable

columns={columns}

data={produits}

/>)};

</div>

  )

}

export default Listproducts

# 

# Ajouter un produit

# Ajouter le bouton Nouveau au composant ListProducts.js

"use client"

import  { useMemo } from 'react';

import { MaterialReactTable } from 'material-react-table';

import { Box } from '@mui/material';

import Image from "next/image"

import Button from 'react-bootstrap/Button';

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

import { useRouter } from "next/navigation";

import Link from 'next/link';

const Listproducts = ({produits}) => {

  const router = useRouter();

  const deletearticle=(id)=>{

    if(window.confirm("supprimer le produit O/N")) {

      deleteArticle(id)

      .then((res)=>{ console.log(res)

       router.refresh()

      })

      .catch(error=>{

          console.log(error)

         })

}

      }

  const columns = useMemo(

    () => [

    {

    accessorKey: 'imageart', //access nested data with dot notation

    header: 'Image',

    Cell: ({ cell}) => (

    <Box

    sx={{

    display: 'flex',

    alignItems: 'center',

    gap: '1rem',

    }}

    >

    <Image

              src={cell.getValue()}

              alt="product anme"

              height="50"

              width="50"

            />

    </Box>),

    },

    {

    accessorKey: 'reference', //access nested data with dot notation

    header: 'Référence',

    size: 100,

    },

    {

    accessorKey: 'designation',

    header: 'Désignation',

    size: 100,

    },

    {

    accessorKey: 'marque', //normal accessorKey

    header: 'Marque',

    size: 100,

    },

    {

    accessorKey: 'prix',

    header: 'Prix',

    size: 100,

    },

    {

    accessorKey: 'qtestock',

    header: 'Stock',

    size: 100,

    },

    {

      accessorKey: '\_id',

      header: 'actions',

      size: 100,

      Cell: ({ cell, row }) => (

      <div >

      <Button

      onClick={() => {

      console.log("modification ...")

      }}

      variant="warning"

      size="md"

      className="text-warning btn-link edit"

      >

      <i class="fa-solid fa-pen-to-square"></i>

      </Button>

      <Button

      onClick={(e) => {

      deletearticle(cell.row.original.\_id,e);

      }}

      variant="danger"

      size="md"

      className="text-danger btn-link delete"

      >

      <i className="fa fa-trash" />

      </Button>

      </div>

      ),

      },

    ],

  [produits],

  )

return (

<div>

<Button

variant='success'

size="sm"

style={{float: 'left','margin':10,'left':10,fontFamily:'Arial'}}>

<Link

href="/admin/products/new"

style={{

  textDecoration: 'none',

  color: 'white',

  fontSize: 14,

}}

>

<i className="fa-solid fa-circle-plus">&nbsp;Nouveau</i>

</Link>

</Button>

{ produits && (

<MaterialReactTable

columns={columns}

data={produits}

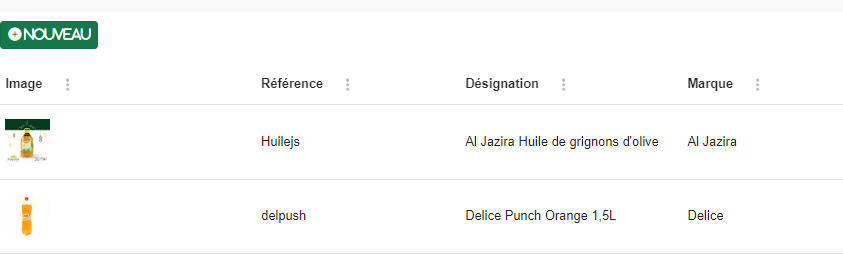
/>)};

</div>

  )

}

export default Listproducts



# Fichier page.js du dossier products/new/page.js

import NewProduct from "@/components/admin/NewProduct";

import {fetchSCategories} from "@/services/ScategorieService"

const getscategories=async()=>{

    const {data}=await fetchSCategories()

    return data;

}

const NewProductPage = async() => {

    const scategories=await getscategories()

  return (

    <div>

     <NewProduct scategories={scategories}/>

    </div>

  )

}

export default NewProductPage

**Ouvrir le composant components/admin/NewProduct.js**

"use client"

import React, { useState } from 'react';

import Button from 'react-bootstrap/Button';

import Form from 'react-bootstrap/Form';

import Col from 'react-bootstrap/Col';

import InputGroup from 'react-bootstrap/InputGroup';

import Row from 'react-bootstrap/Row';

import { useRouter } from "next/navigation";

import { addArticle } from '@/services/ArticlesService';

const NewProduct = ({scategories}) => {

  const router = useRouter();

  const [reference, setReference] = useState("");

const [designation, setDesignation] = useState("");

const [prix, setPrix] = useState("");

const [marque, setMarque] = useState("");

const [qtestock, setQtestock] = useState("");

const [imageart, setImageart] = useState("");

const [scategorieID, setScategorieID] = useState("");

const [validated, setValidated] = useState(false);

const handleSubmit = (e) => {

e.preventDefault();

const form = e.currentTarget;

if (form.checkValidity() === true) {

const newProduct = {

reference,

designation,

prix,

marque,

qtestock,

imageart,

scategorieID

};

//faire le add dans la BD

addArticle(newProduct)

.then(res => {

  router.push('/admin/products')

    router.refresh()

})

.catch(error=>{

alert("Erreur ! Insertion non effectuée")

})

}

setValidated(true);

}

return (

<div>

 <Form noValidate validated={validated} onSubmit={handleSubmit}>

 <h2>Create Product</h2>

<div className="container w-100 d-flex justify-content-center">

<div>

<div className='form mt-3'>

<Row className="mb-2">

<Form.Group as={Col} md="6" >

<Form.Label >Référence \*</Form.Label>

<Form.Control

required

type="text"

placeholder="Référence"

value={reference}

onChange={(e)=>setReference(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Référence Article

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Désignation \*</Form.Label>

<Form.Control

required

type="text"

placeholder="Désignation"

value={designation}

onChange={(e)=>setDesignation(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Désignation

</Form.Control.Feedback>

</Form.Group>

</Row>

<Row className="mb-2">

<Form.Group className="col-md-6">

<Form.Label>Marque \*</Form.Label>

<InputGroup hasValidation>

<Form.Control

type="text"

required

placeholder="Marque"

value={marque}

onChange={(e)=>setMarque(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Marque Incorrecte

</Form.Control.Feedback>

</InputGroup>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Prix</Form.Label>

<Form.Control

type="number"

placeholder="Prix"

value={prix}

onChange={(e)=>setPrix(e.target.value)}

/>

</Form.Group>

</Row>

<Row className="mb-3">

<Form.Group className="col-md-6 ">

<Form.Label>

Qté stock<span className="req-tag">\*</span>

</Form.Label>

<Form.Control

required

type="number"

value={qtestock}

onChange={(e)=>setQtestock(e.target.value)}

placeholder="Qté stock"

/>

<Form.Control.Feedback type="invalid">

Qté stock Incorrect

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Image</Form.Label>

<Form.Control

type="text"

placeholder="Image"

value={imageart}

onChange={(e)=>setImageart(e.target.value)}

/>

</Form.Group>

<Form.Group as={Col} md="12">

<Form.Label>S/Catégorie</Form.Label>

<Form.Control

as="select"

type="select"

value={scategorieID}

onChange={(e)=>setScategorieID(e.target.value)}

>

<option></option>

{scategories.map((scat)=><option key={scat.\_id}

value={scat.\_id}>{scat.nomscategorie}</option>

)}

</Form.Control>

</Form.Group>

</Row>

</div>

</div>

</div>

<Button type="submit">Enregistrer</Button>

<Button type="button" className="btn btn-warning"

onClick={()=>handleReset()}>Annuler</Button>

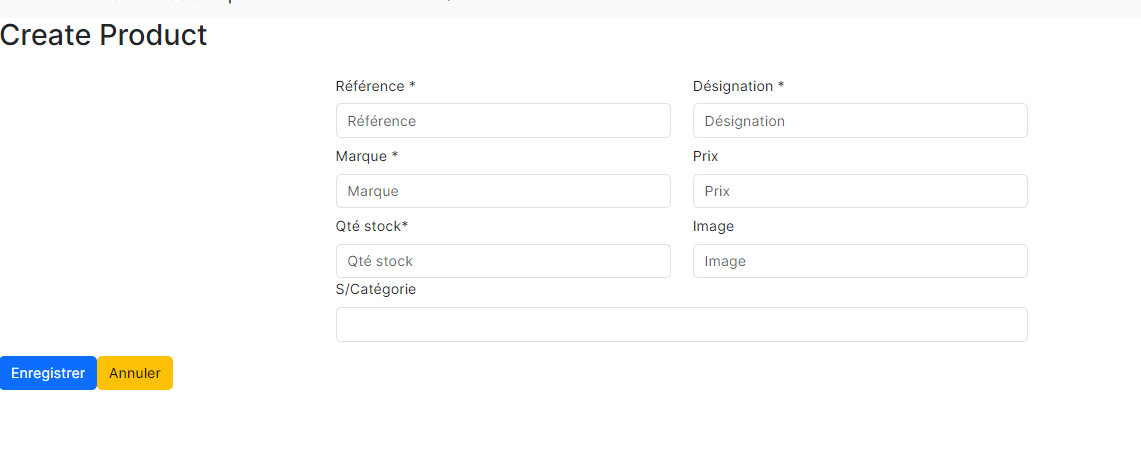
</Form>

</div>

);

};

export default NewProduct



# Ajouter un produit avec firebase filepond

# Configurer Firebase

Accéder au site :

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

# Créer un bucket de stockage cloud

Firebase stocke les fichiers dans un bucket de stockage cloud. Suivez les étapes suivantes pour le créer :

Sur la page de présentation du projet, cliquez sur l'onglet Storage dans le panneau de navigation de gauche.

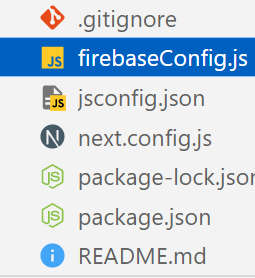


# Ajouter Firebase dans Next JS

Dans votre terminal, accédez au dossier de votre projet. Exécutez la commande suivante pour installer le SDK Firebase :

npm install firebase --legacy-peer-deps

Créez un nouveau fichier appelé firebaseConfig.js dans la racine du projet.



Il faut y coller le code précédemment copié dans les étapes de création du projet dans Firebase. Puis lui ajouter l’apppel de getStorage car on va en avoir besoin pour le stockage des images.

Ce qui donne ce code :

// Import the functions you need from the SDKs you need

import { initializeApp } from "firebase/app";

import { getStorage } from "firebase/storage";

// Initialize Firebase

const app = initializeApp ({

  apiKey: "AIzaSyATRVSxM4NOndpyJct3th74FY4HQqb-B2I",

  authDomain: "nextproject-76ce9.firebaseapp.com",

  projectId: "nextproject-76ce9",

  storageBucket: "nextproject-76ce9.appspot.com",

  messagingSenderId: "474273333859",

  appId: "1:474273333859:web:ae5786efe351cb8ca66e56"

});

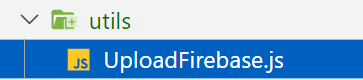
// Firebase storage reference

const storage = getStorage(app);

export default storage;

La dernière ligne exporte la référence de stockage Firebase afin que vous puissiez accéder à cette instance à partir du reste de votre application.

Ensuite, sous le dossier racine créer le dossier utils. Puis il faut y créer le fichier UploadFirebase.js



La méthode Promise.resolve() « résout » une valeur donnée en une promesse. Si la valeur est une promesse, cette promesse est renvoyée.

L'opérateur await permet d'attendre la résolution d'une promesse (Promise). Il ne peut être utilisé qu'au sein d'une fonction asynchrone (définie avec l'instruction async function).

import  storage from "../firebaseConfig";

import { ref, uploadBytesResumable, getDownloadURL } from "firebase/storage";

export const UploadFirebase=async(file)=>{

    let imageurl=""

    const storageRef = ref(storage, `/files/${file.name}`);

    const uploadTask = uploadBytesResumable(storageRef, file);

    await new Promise(resolve => {

    uploadTask.on(

        "state\_changed",

        (snapshot) => {

            const percent = Math.round(

                (snapshot.bytesTransferred / snapshot.totalBytes) \* 100

            );

           console.log(percent);

        },

        (err) => console.log(err),

        async () => {

                await getDownloadURL(uploadTask.snapshot.ref).then((url) => {

                imageurl=url;

                console.log(imageurl);

                resolve();

            });

        }

    );

}

)

    return imageurl;

}

# Formulaire d’ajout

D’abord installer

npm i react-bootstrap bootstrap

On va utiliser Filepond qui est une bibliothèque JavaScript qui peut télécharger tout ce que vous lui lancez, optimise les images pour des téléchargements plus rapides et offre une expérience utilisateur exceptionnelle, accessible et fluide.

<https://www.npmjs.com/package/filepond>

Faire les installations suivantes :

npm i filepond --legacy-peer-deps

npm i react-filepond --legacy-peer-deps

npm i filepond-plugin-image-exif-orientation --legacy-peer-deps

npm i filepond-plugin-image-preview --legacy-peer-deps

Les images seront sélectionnées puis seront uploadées dans Firebase. Ce dernier nous retournera l’url correspondantes. Pour cela il faut ajouter le site de Firebase dans le tableau domains dans le config pour que l’affichage de ces images soit autorisé.



Ajouter firebasestorage.googleapis.com

/\*\* @type {import('next').NextConfig} \*/

const nextConfig = {

    images:{

        domains:["firebasestorage.googleapis.com","res.cloudinary.com"]

        },

        env: {

        API\_URL: "http://localhost:3001/api"

        }

}

module.exports = nextConfig

composant NewProduct.js

"use client"

import React, { useState } from 'react';

import Button from 'react-bootstrap/Button';

import Form from 'react-bootstrap/Form';

import Col from 'react-bootstrap/Col';

import InputGroup from 'react-bootstrap/InputGroup';

import Row from 'react-bootstrap/Row';

import { useRouter } from "next/navigation";

import { addArticle } from '@/services/ArticlesService';

import {UploadFirebase} from '../../utils/UploadFirebase';

import { FilePond,registerPlugin } from 'react-filepond'

import 'filepond/dist/filepond.min.css';

import FilePondPluginImageExifOrientation from 'filepond-plugin-image-exif-orientation'

import FilePondPluginImagePreview from 'filepond-plugin-image-preview'

import 'filepond-plugin-image-preview/dist/filepond-plugin-image-preview.css'

registerPlugin(FilePondPluginImageExifOrientation, FilePondPluginImagePreview)

const NewProduct = ({scategories}) => {

  const router = useRouter();

  const [reference, setReference] = useState("");

const [designation, setDesignation] = useState("");

const [prix, setPrix] = useState("");

const [marque, setMarque] = useState("");

const [qtestock, setQtestock] = useState("");

const [imageart, setImageart] = useState("");

const [scategorieID, setScategorieID] = useState("");

const [validated, setValidated] = useState(false);

const [file, setFile] = useState("");

const handleReset=()=>{

  router.push('/admin/products')

}

const handleSubmit = (url) => {

const newProduct = {

reference,

designation,

prix,

marque,

qtestock,

imageart,

scategorieID

};

//faire le add dans la BD

addArticle(newProduct)

.then(res => {

  router.push('/admin/products')

    router.refresh()

})

.catch(error=>{

alert("Erreur ! Insertion non effectuée")

})

}

const handleUpload = (event) => {

  event.preventDefault();

  const form = event.currentTarget;

 if (form.checkValidity() === true) {

        if (!imageart) {

            alert("Please upload an image first!");

        }

        else {

          handleSubmit()

      }

 setValidated(true);

};

}

const resultHandleUpload = async(file,load) => {

  try {

  const url =  await UploadFirebase(file);

 if (url){

  load(url);

  setImageart(url)

 }

 } catch (error) {

    console.log(error);

 }

}

const process = (

  fieldName,

    file,

    metadata,

    load,

    error,

    progress,

    abort,

    transfer,

    options

  ) => {

    resultHandleUpload(file,load);

    }

return (

<div>

<Form noValidate validated={validated} onSubmit={handleUpload}>

 <h2>Create Product</h2>

<div className="container w-100 d-flex justify-content-center">

<div>

<div className='form mt-3'>

<Row className="mb-2">

<Form.Group as={Col} md="6" >

<Form.Label >Référence \*</Form.Label>

<Form.Control

required

type="text"

placeholder="Référence"

value={reference}

onChange={(e)=>setReference(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Référence Article

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Désignation \*</Form.Label>

<Form.Control

required

type="text"

placeholder="Désignation"

value={designation}

onChange={(e)=>setDesignation(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Désignation

</Form.Control.Feedback>

</Form.Group>

</Row>

<Row className="mb-2">

<Form.Group className="col-md-6">

<Form.Label>Marque \*</Form.Label>

<InputGroup hasValidation>

<Form.Control

type="text"

required

placeholder="Marque"

value={marque}

onChange={(e)=>setMarque(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Marque Incorrecte

</Form.Control.Feedback>

</InputGroup>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Prix</Form.Label>

<Form.Control

type="number"

placeholder="Prix"

value={prix}

onChange={(e)=>setPrix(e.target.value)}

/>

</Form.Group>

</Row>

<Row className="mb-3">

<Form.Group className="col-md-6 ">

<Form.Label>

Qté stock<span className="req-tag">\*</span>

</Form.Label>

<Form.Control

required

type="number"

value={qtestock}

onChange={(e)=>setQtestock(e.target.value)}

placeholder="Qté stock"

/>

<Form.Control.Feedback type="invalid">

Qté stock Incorrect

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Image</Form.Label>

<FilePond

              files={file}

              allowMultiple={false}

              onupdatefiles={setFile}

              labelIdle='<span class="filepond--label-action">Browse One</span>'

              server={{ process }}

              name="file"

            />

</Form.Group>

<Form.Group as={Col} md="12">

<Form.Label>S/Catégorie</Form.Label>

<Form.Control

as="select"

type="select"

value={scategorieID}

onChange={(e)=>setScategorieID(e.target.value)}

>

<option></option>

{scategories.map((scat)=><option key={scat.\_id}

value={scat.\_id}>{scat.nomscategorie}</option>

)}

</Form.Control>

</Form.Group>

</Row>

</div>

</div>

</div>

<Button type="submit">Enregistrer</Button>

<Button type="button" className="btn btn-warning"

onClick={()=>handleReset()}>Annuler</Button>

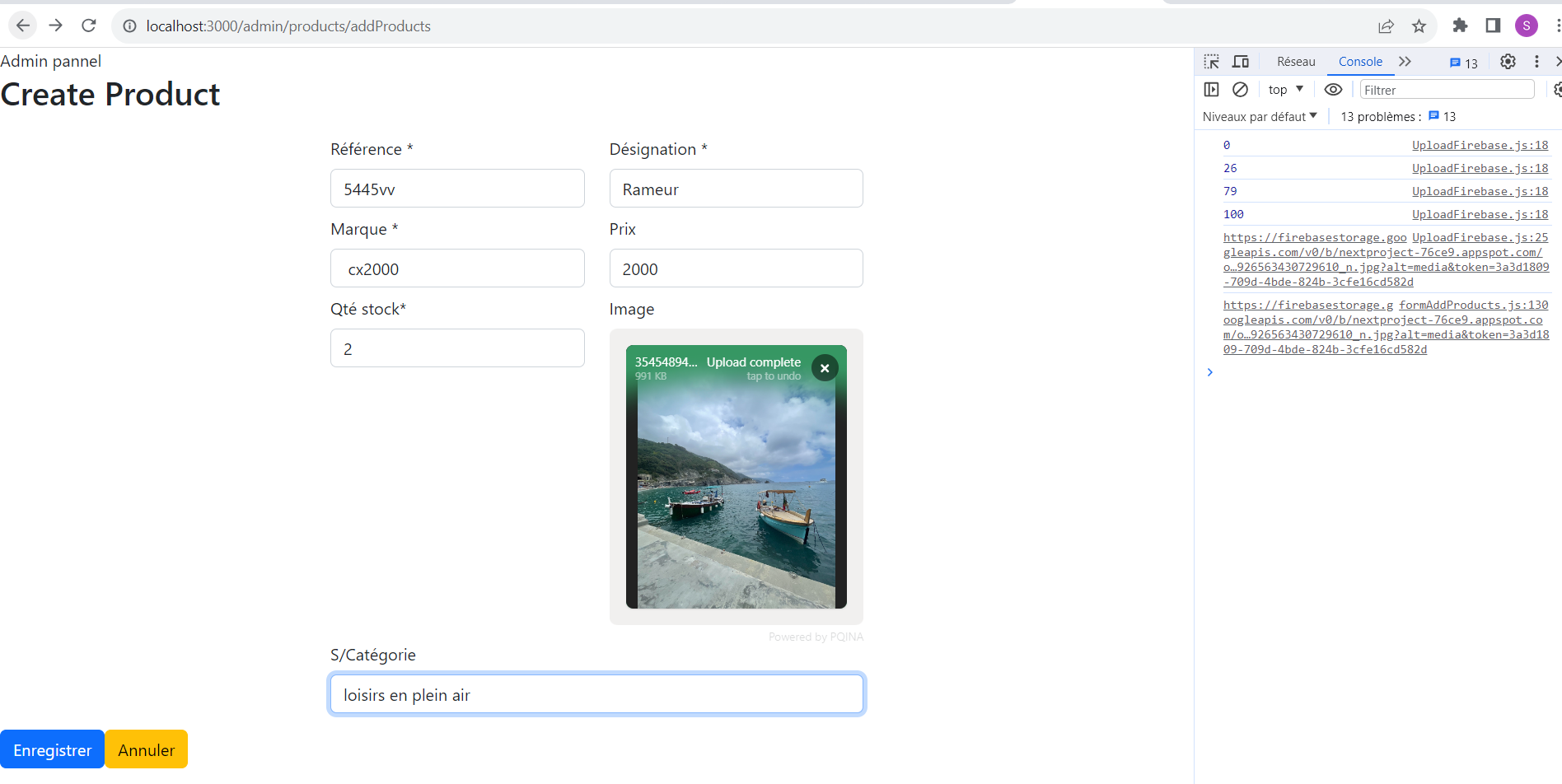
</Form>

</div>

);

};

export default NewProduct



Modifier un article



import {fetchSCategories} from "@/services/ScategorieService"

import UpdateProduct from '@/components/admin/UpdateProduct';

const getscategories=async()=>{

    const {data}=await fetchSCategories()

    return data;

}

const ProductUpdatePage = async({params}) => {

  const scategories = await getscategories();

  return (

    <div>

       <UpdateProduct params={params} scategories={scategories} />

    </div>

  )

}

export default ProductUpdatePage



Fichier components/admin/ListProducts.js

"use client"

import  { useMemo } from 'react';

import { MaterialReactTable } from 'material-react-table';

import { Box } from '@mui/material';

import Image from "next/image"

import Button from 'react-bootstrap/Button';

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

import { useRouter } from "next/navigation";

import Link from 'next/link';

const Listproducts = ({produits}) => {

  const router = useRouter();

  const deletearticle=(id)=>{

    if(window.confirm("supprimer le produit O/N")) {

      deleteArticle(id)

      .then((res)=>{ console.log(res)

       router.refresh()

      })

      .catch(error=>{

          console.log(error)

         })

}

      }

  const columns = useMemo(

    () => [

    {

    accessorKey: 'imageart', //access nested data with dot notation

    header: 'Image',

    Cell: ({ cell}) => (

    <Box

    sx={{

    display: 'flex',

    alignItems: 'center',

    gap: '1rem',

    }}

    >

    <Image

              src={cell.getValue()}

              alt="product anme"

              height="50"

              width="50"

            />

    </Box>),

    },

    {

    accessorKey: 'reference', //access nested data with dot notation

    header: 'Référence',

    size: 100,

    },

    {

    accessorKey: 'designation',

    header: 'Désignation',

    size: 100,

    },

    {

    accessorKey: 'marque', //normal accessorKey

    header: 'Marque',

    size: 100,

    },

    {

    accessorKey: 'prix',

    header: 'Prix',

    size: 100,

    },

    {

    accessorKey: 'qtestock',

    header: 'Stock',

    size: 100,

    },

    {

      accessorKey: '\_id',

      header: 'actions',

      size: 100,

      Cell: ({ cell, row }) => (

      <div >

      <Button

      variant="warning"

      size="md"

      className="text-warning btn-link edit"

      >

        <Link href={`/admin/products/update/${cell.row.original.\_id}`}>

        <i class="fa-solid fa-pen-to-square"></i>

    </Link>

      </Button>

      <Button

      onClick={(e) => {

      deletearticle(cell.row.original.\_id,e);

      }}

      variant="danger"

      size="md"

      className="text-danger btn-link delete"

      >

      <i className="fa fa-trash" />

      </Button>

      </div>

      ),

      },

    ],

  [produits],

  )

return (

<div>

<Button

variant='success'

size="sm"

style={{float: 'left','margin':10,'left':10,fontFamily:'Arial'}}>

<Link

href="/admin/products/new"

style={{

  textDecoration: 'none',

  color: 'white',

  fontSize: 14,

}}

>

<i className="fa-solid fa-circle-plus">&nbsp;Nouveau</i>

</Link>

</Button>

{ produits && (

<MaterialReactTable

columns={columns}

data={produits}

/>)};

</div>

  )

}

export default Listproducts

Fichier SSR app/admin/products/update/[id]/page.js

import {fetchSCategories} from "@/services/ScategorieService"

import UpdateProduct from '@/components/admin/UpdateProduct';

const getscategories=async()=>{

    const {data}=await fetchSCategories()

    return data;

}

const ProductUpdatePage = async({params}) => {

  const scategories = await getscategories();

  return (

    <div>

       <UpdateProduct params={params} scategories={scategories} />

    </div>

  )

}

export default ProductUpdatePage

Composant components/admin/UpdateProduct

'use client';

import React, { useState,useEffect } from 'react';

import { useRouter } from 'next/navigation';

import Button from 'react-bootstrap/Button';

import Form from 'react-bootstrap/Form';

import Col from 'react-bootstrap/Col';

import InputGroup from 'react-bootstrap/InputGroup';

import Row from 'react-bootstrap/Row';

import { FilePond,registerPlugin } from 'react-filepond'

import 'filepond/dist/filepond.min.css';

import FilePondPluginImageExifOrientation from 'filepond-plugin-image-exif-orientation'

import FilePondPluginImagePreview from 'filepond-plugin-image-preview'

import 'filepond-plugin-image-preview/dist/filepond-plugin-image-preview.css'

registerPlugin(FilePondPluginImageExifOrientation, FilePondPluginImagePreview)

import {UploadFirebase} from '../../utils/UploadFirebase';

import {editArticle,fetchArticleById} from "@/services/ArticlesService"

const getProductDetails=async(id)=>{

    const {data}=await fetchArticleById(id);

    return data;

}

const UpdateProduct = ({params,scategories}) => {

  const router = useRouter()

  const [reference, setReference] = useState("");

  const [designation, setDesignation] = useState("");

  const [prix, setPrix] = useState("");

  const [marque, setMarque] = useState("");

  const [qtestock, setQtestock] = useState("");

  const [imageart, setImageart] = useState("");

  const [scategorieID, setScategorieID] = useState("");

  const [file, setFile] = useState("");

  const [validated, setValidated] = useState(false);

  useEffect(() => {

    getProductDetails(params.id)

    .then((article) => {

      setReference(article.reference)

      setDesignation(article.designation)

      setPrix(article.prix)

      setMarque(article.marque)

      setQtestock(article.qtestock)

      setImageart(article.imageart)

      setScategorieID(article.scategorieID)

      setFile([  {

        source: article.imageart,

        options: {

          type: 'local'

        },

      },

    ])

    }).catch((err)=>console.log(err));

   }, []);

  const handleSubmit = async(event) => {

    event.preventDefault();

    const form = event.currentTarget;

    if (form.checkValidity() === true) {

      const article={

        \_id:params.id,

      reference: reference,

      designation: designation,

      prix: prix,

      marque: marque,

      qtestock: qtestock,

      imageart: imageart,

      scategorieID: scategorieID

      }

      await editArticle(article)

      .then(res=>{

      router.push("/admin/products")

      router.refresh()

      })

      .catch(error=>{

      console.log(error)

      alert("Erreur ! Modification non effectuée")

      })

      }

      setValidated(true);

      };

      const handleUpload = (event) => {

        event.preventDefault();

        const form = event.currentTarget;

       if (form.checkValidity() === true) {

              if (!imageart) {

                  alert("Please upload an image first!");

              }

              else {

                handleSubmit(event)

            }

       setValidated(true);

      };

      }

      const resultHandleUpload = async(file,load) => {

        try {

        const url =  await UploadFirebase(file);

       if (url){

        load(url);

        setImageart(url)

       }

       } catch (error) {

          console.log(error);

       }

      }

      const process = (

        fieldName,

          file,

          metadata,

          load,

          error,

          progress,

          abort,

          transfer,

          options

        ) => {

          resultHandleUpload(file,load);

          }

    const handleReset=()=>{

      setReference('');

        setDesignation('');

        setPrix('');

        setMarque('');

        setQtestock('');

        setImageart('');

        setScategorieID('');

        setValidated(false);

        setFile('')

    }

  return (

    <div>

    <Form noValidate validated={validated} onSubmit={handleUpload}>

<div className="container w-100 d-flex justify-content-center">

<div>

<div className='form mt-3'>

<Row className="mb-2">

<Form.Group as={Col} md="6" >

<Form.Label >Référence \*</Form.Label>

<Form.Control

required

type="text"

placeholder="Référence"

value={reference}

onChange={(e)=>setReference(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Référence Article

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Désignation \*</Form.Label>

<Form.Control

required

type="text"

placeholder="Désignation"

value={designation}

onChange={(e)=>setDesignation(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Désignation

</Form.Control.Feedback>

</Form.Group>

</Row>

<Row className="mb-2">

<Form.Group className="col-md-6">

<Form.Label>Marque \*</Form.Label>

<InputGroup hasValidation>

<Form.Control

type="text"

required

placeholder="Marque"

value={marque}

onChange={(e)=>setMarque(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Marque Incorrecte

</Form.Control.Feedback>

</InputGroup>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Prix</Form.Label>

<Form.Control

type="number"

placeholder="Prix"

value={prix}

onChange={(e)=>setPrix(e.target.value)}

/>

</Form.Group>

</Row>

<Row className="mb-3">

<Form.Group className="col-md-6 ">

<Form.Label>

Qté stock<span className="req-tag">\*</span>

</Form.Label>

<Form.Control

required

type="number"

value={qtestock}

onChange={(e)=>setQtestock(e.target.value)}

placeholder="Qté stock"

/>

<Form.Control.Feedback type="invalid">

Qté stock Incorrect

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Image</Form.Label>

<div style={{width:200, height:250}}>

<FilePond

              files={file}

              allowMultiple={false}

              onupdatefiles={setFile}

              labelIdle='<span class="filepond--label-action">Browse One</span>'

              name="file"

              server={{

                  load: (source, load, error, progress, abort, headers) => {

                  var myRequest = new Request(source);

                  fetch(myRequest).then(function(response) {

                    response.blob().then(function(myBlob) {

                      load(myBlob);

                    });

                  });

                },

                process

              }}

            />

    </div>

</Form.Group>

<Form.Group as={Col} md="12">

<Form.Label>S/Catégorie</Form.Label>

<Form.Control

as="select"

type="select"

value={scategorieID}

onChange={(e)=>setScategorieID(e.target.value)}

>

<option></option>

{scategories?.map((scat)=><option key={scat.\_id}

value={scat.\_id}>{scat.nomscategorie}</option>)}

</Form.Control>

</Form.Group>

</Row>

</div>

</div>

</div>

<Button type="submit">Enregistrer</Button>

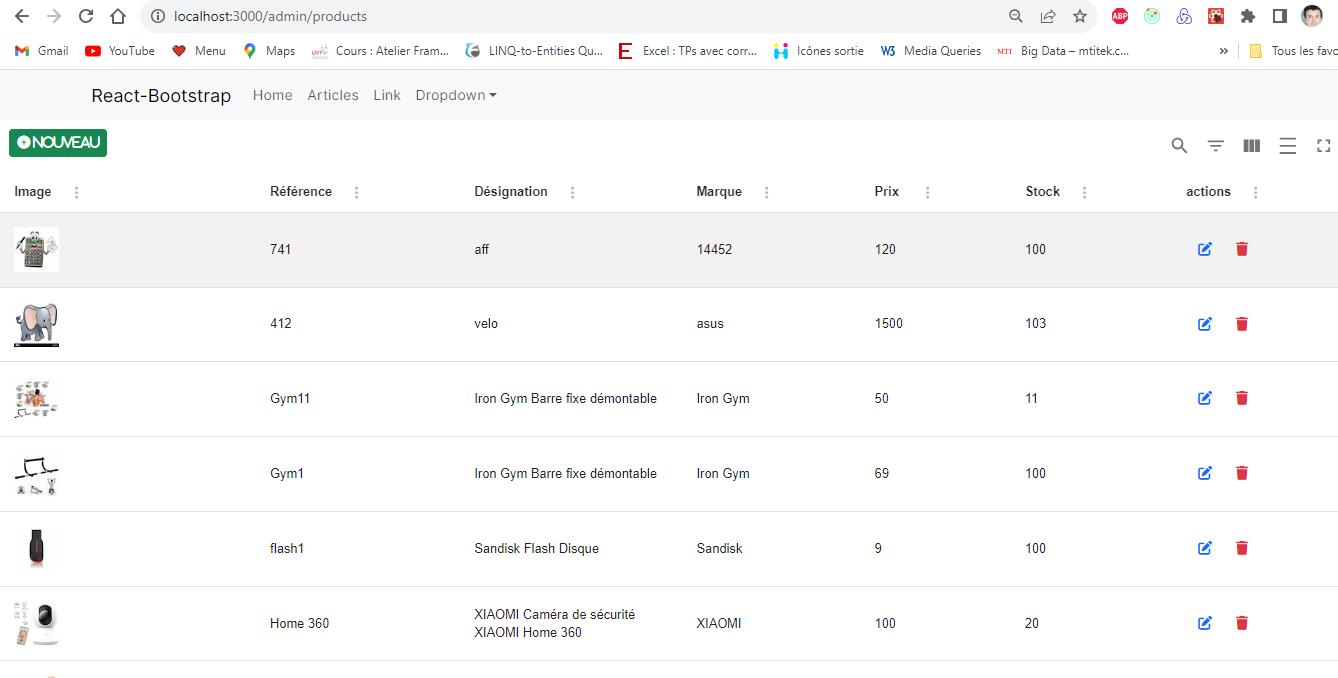
</Form>

 </div>

  )

}

export default UpdateProduct





Client Side

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

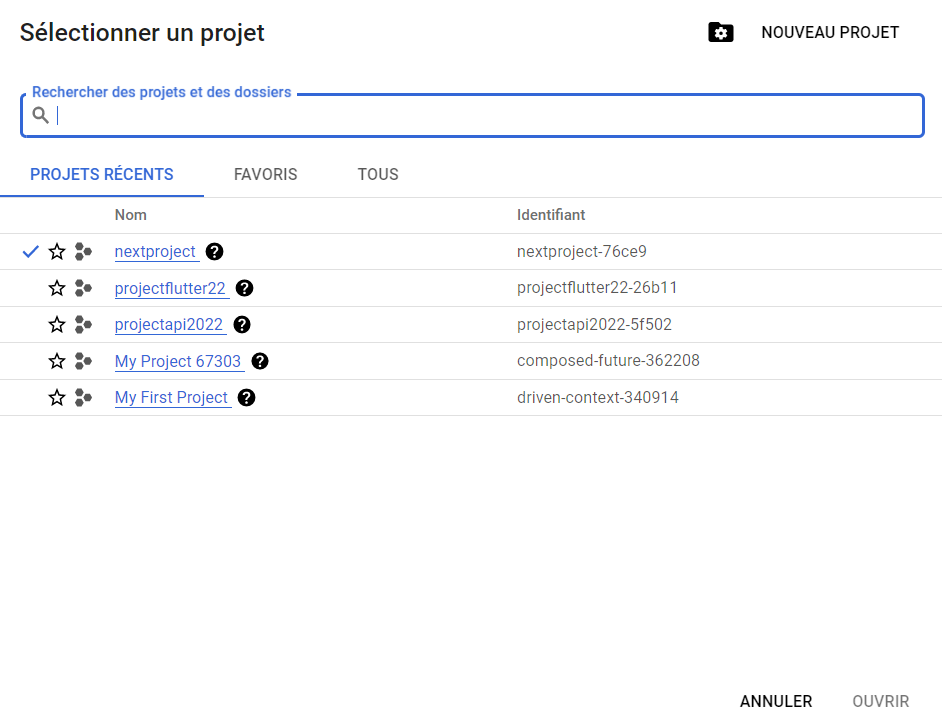
npm i next-auth

## Google Login

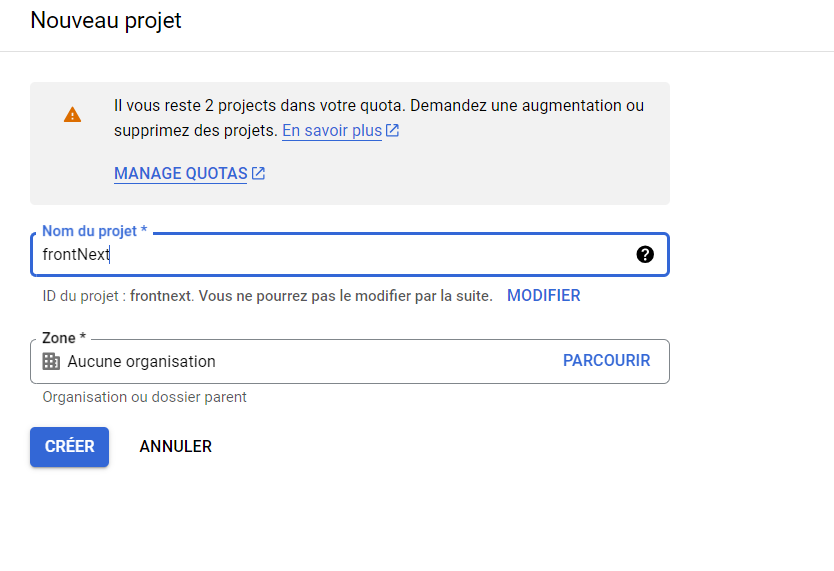
Aller au site :

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

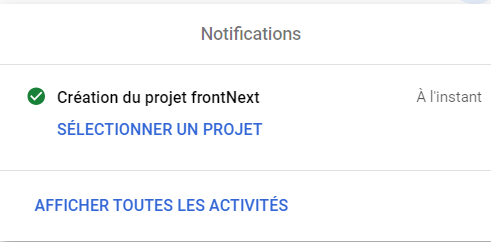




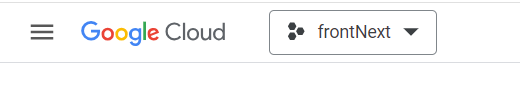




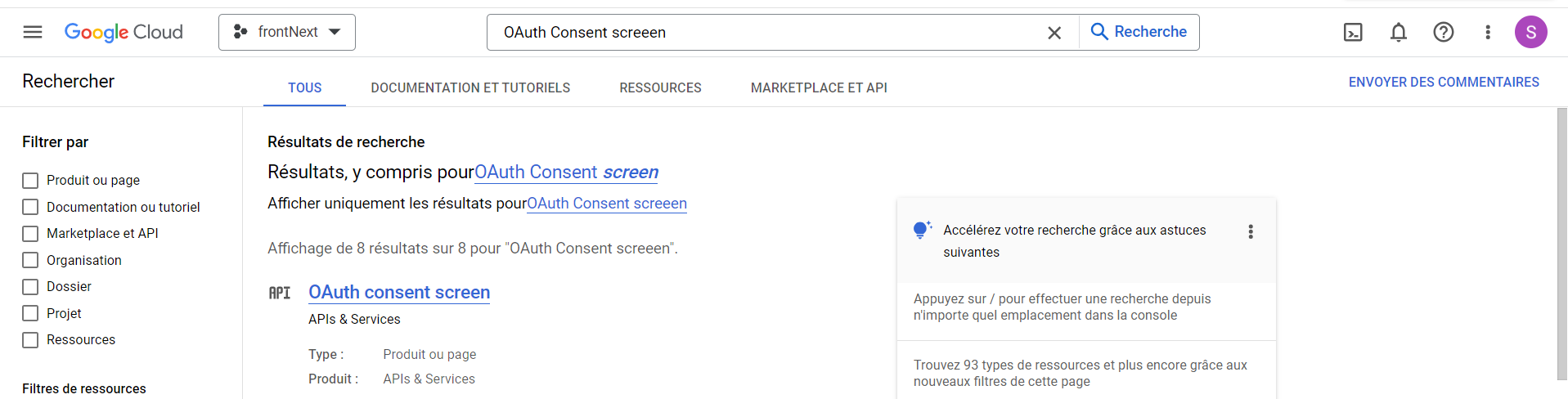


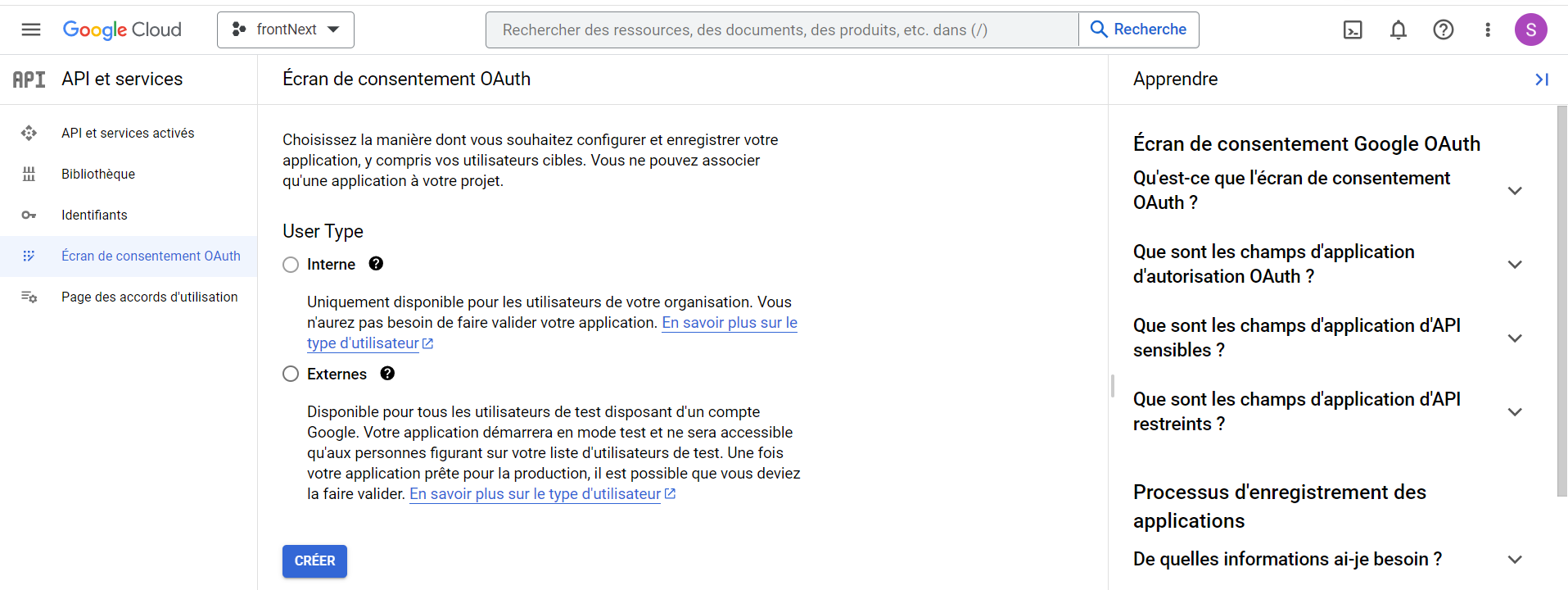


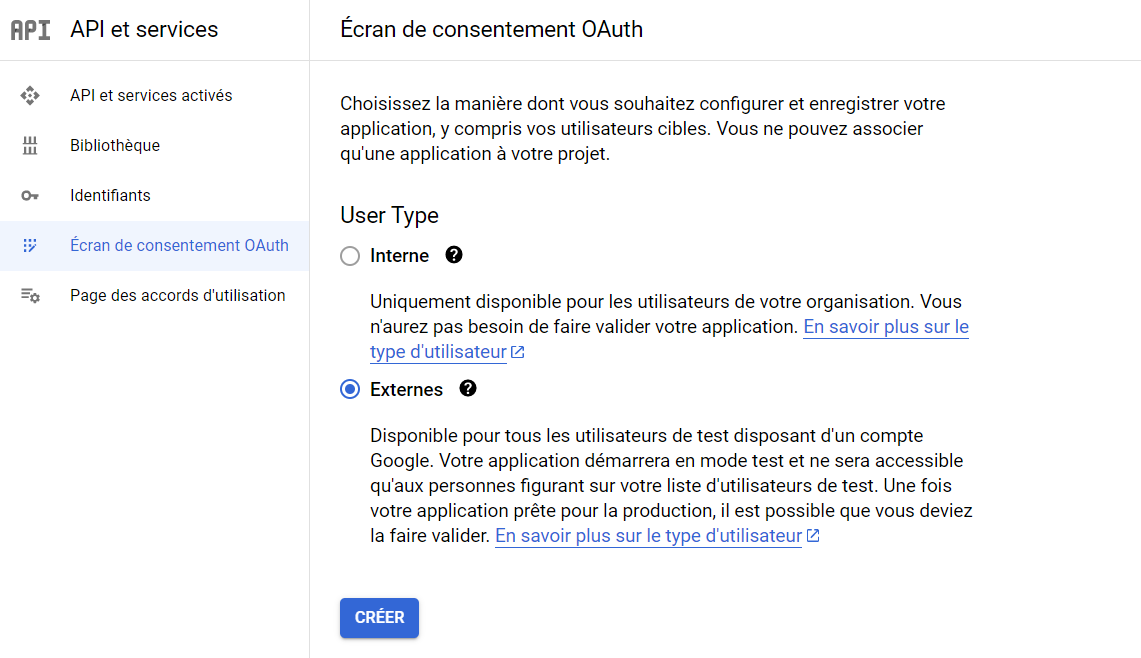


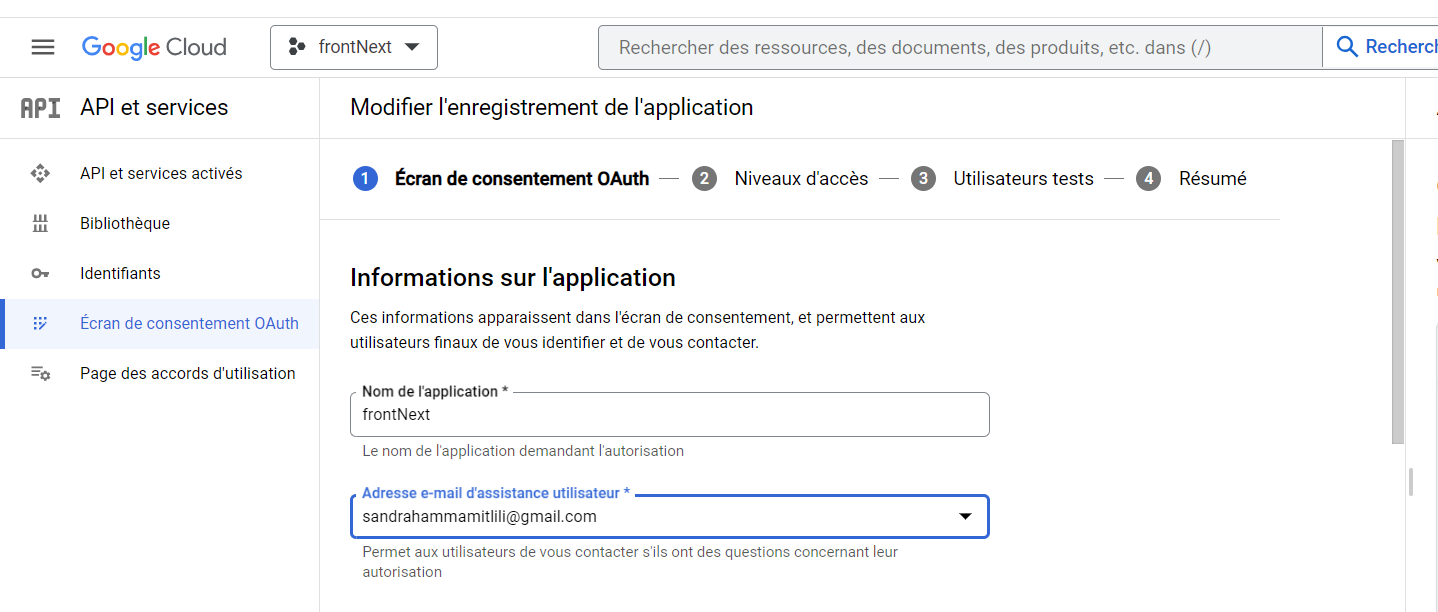






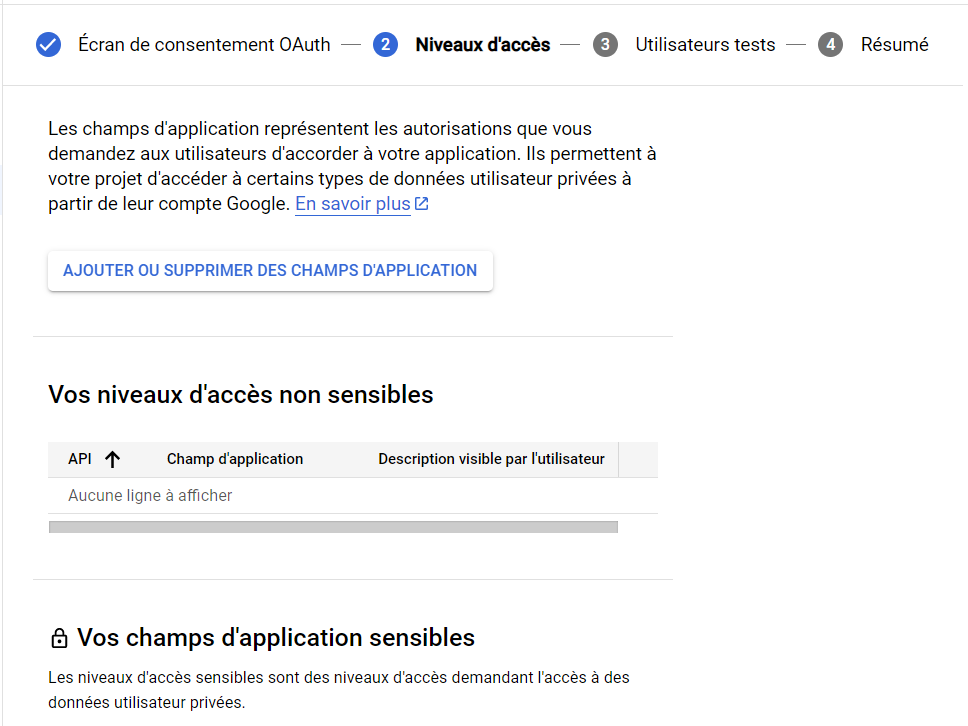


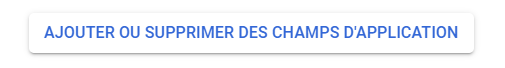


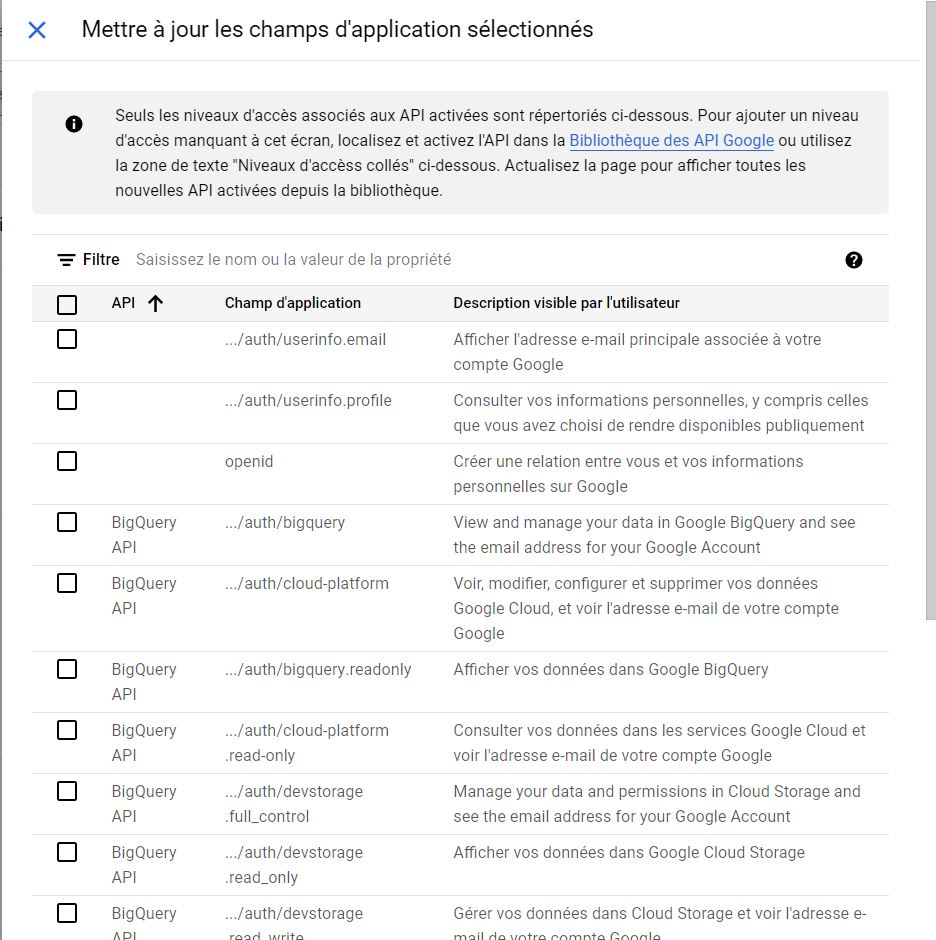


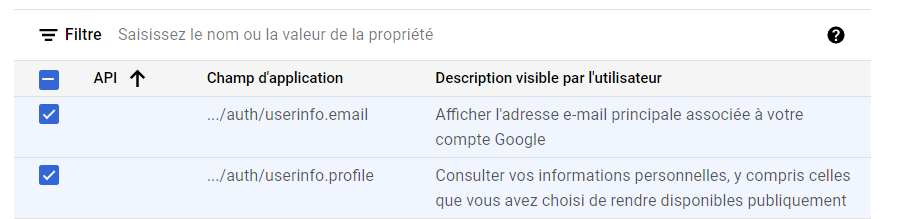


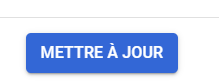


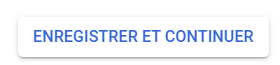


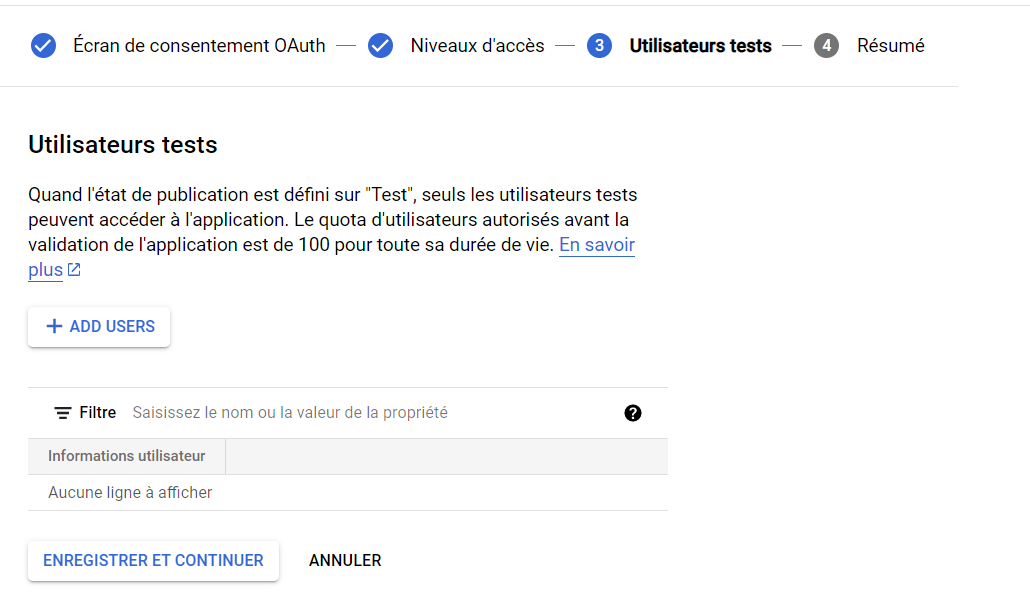


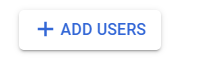


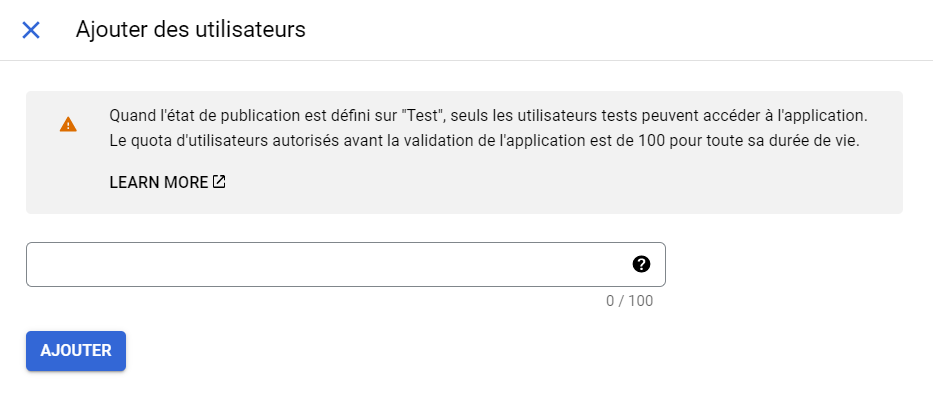








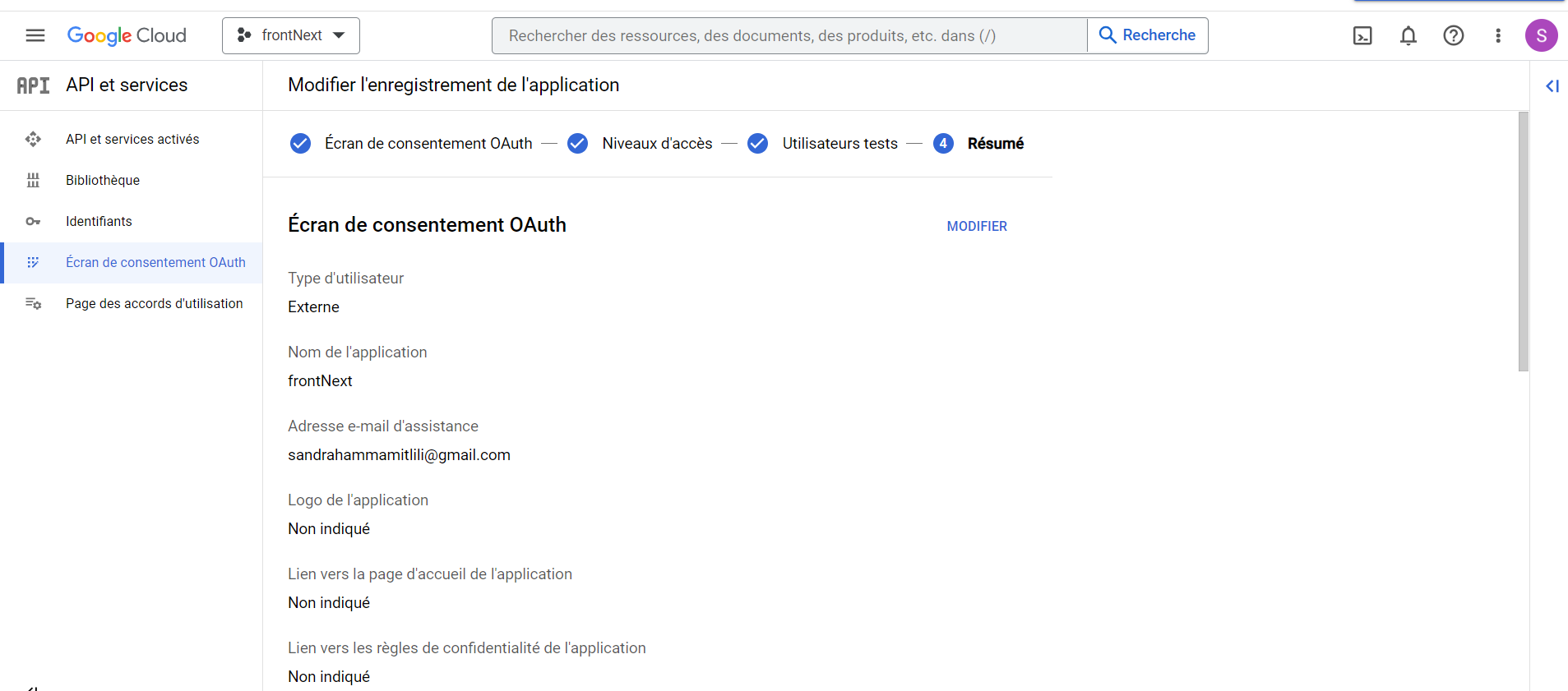


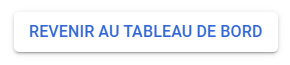


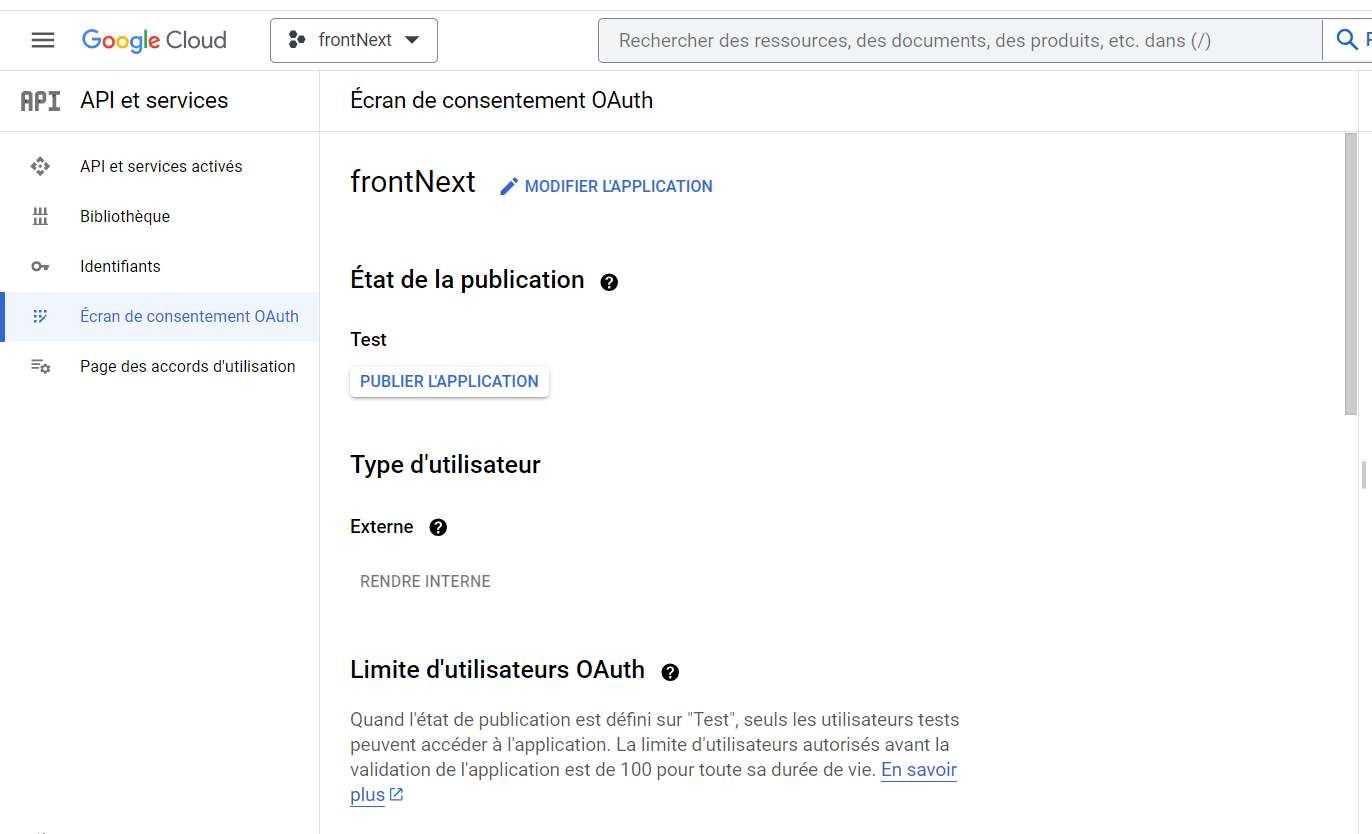




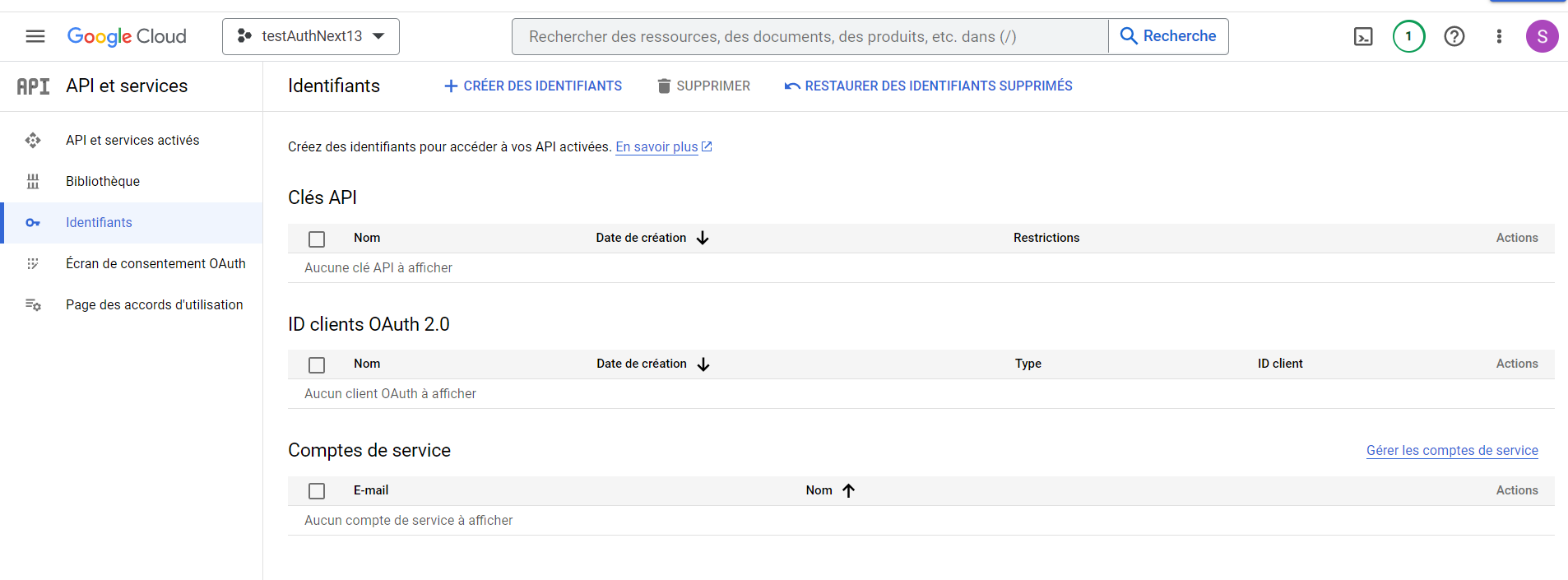




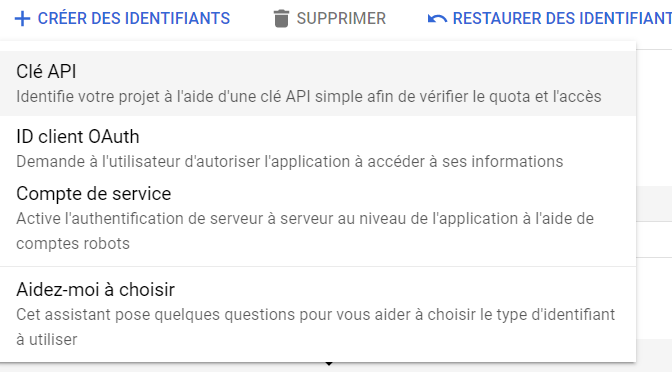




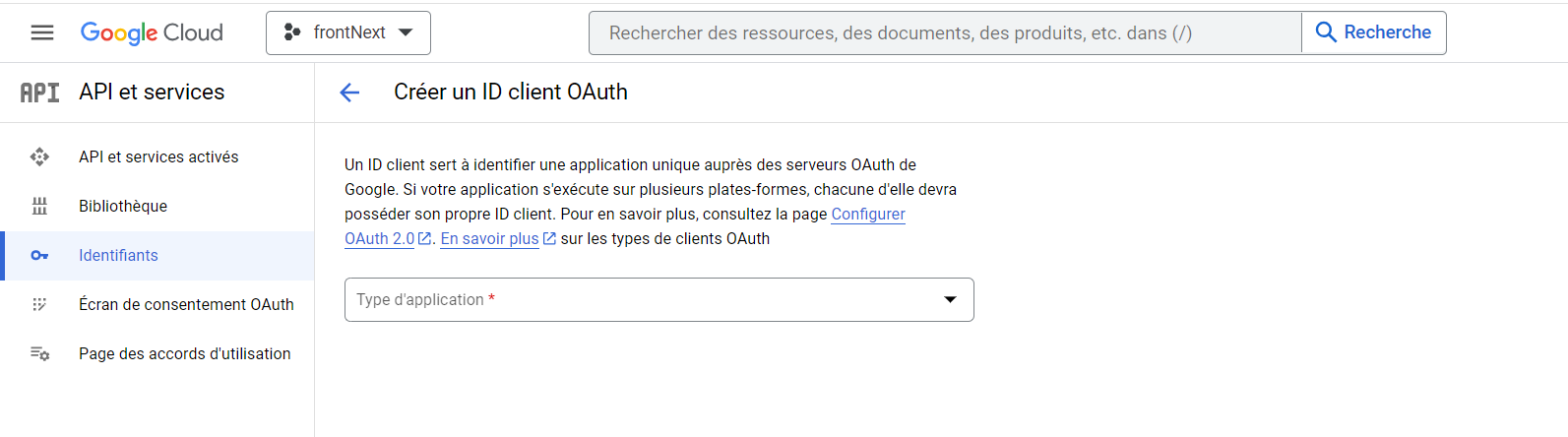




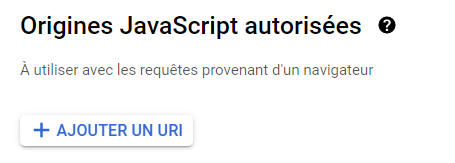


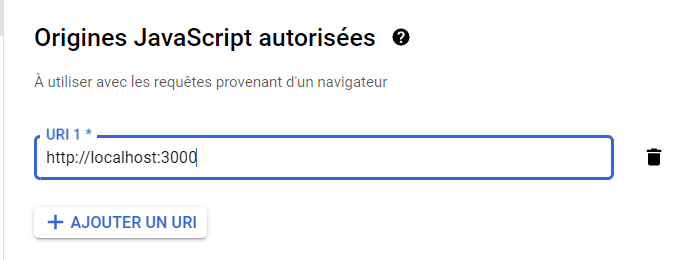






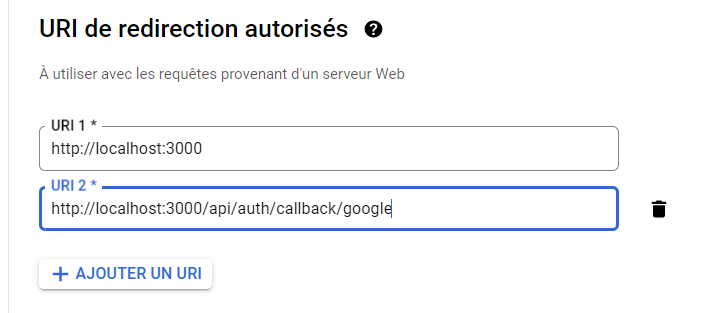




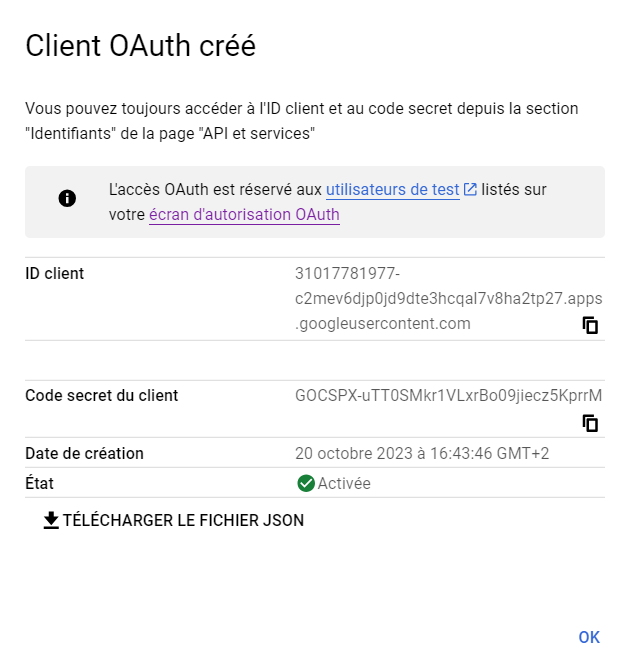










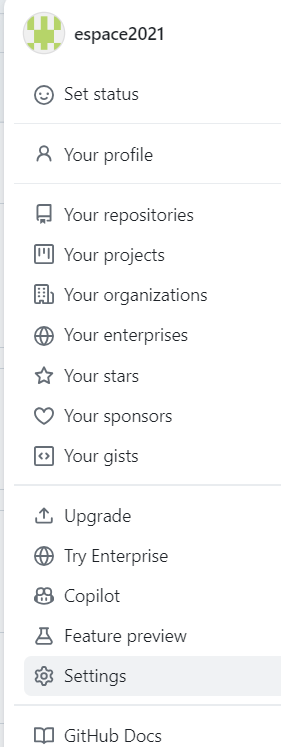


ID client : 31017781977-c2mev6djp0jd9dte3hcqal7v8ha2tp27.apps.googleusercontent.com

Code secret du client : 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

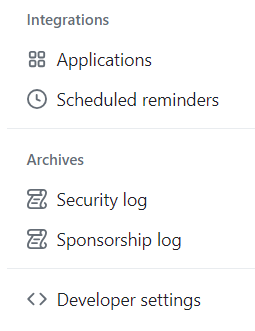
## GitHub Credentials

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

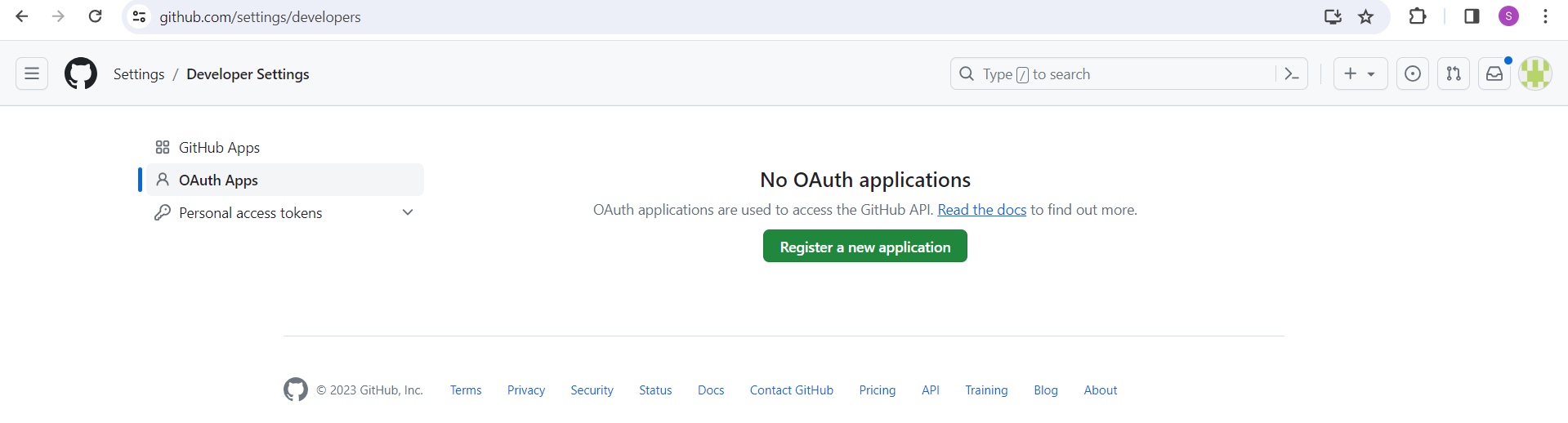


Sélectionnez Settings dans le menu déroulant :

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 :

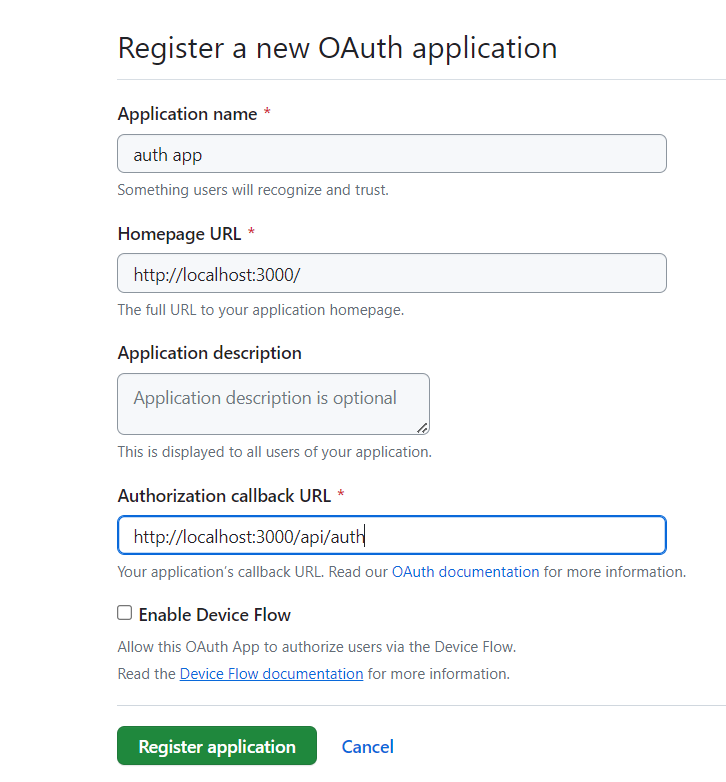


Remplissez le formulaire que vous voyez sur la page qui suit :

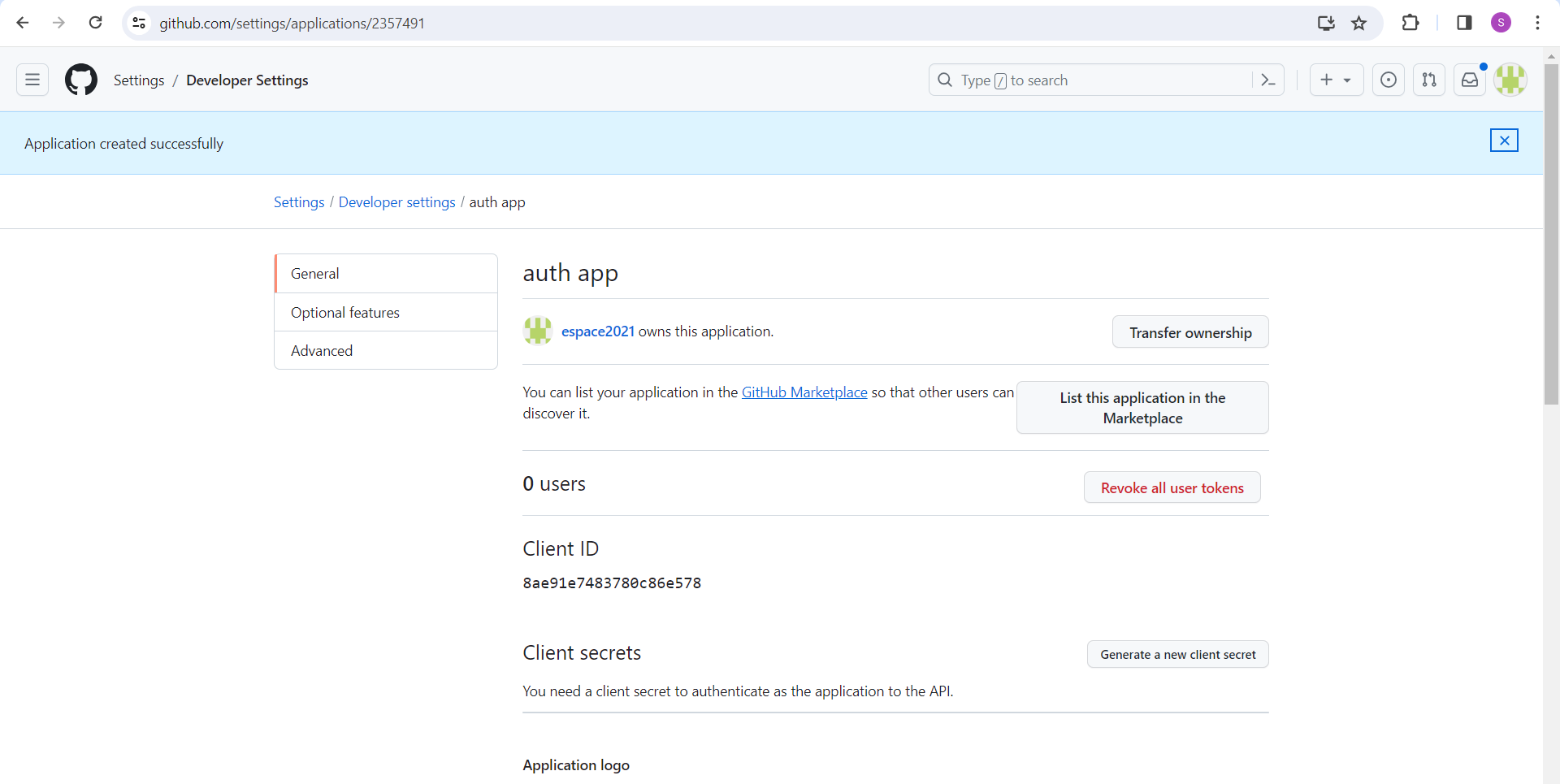
Application name - auth app

Homepage URL - http://localhost:3000/

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



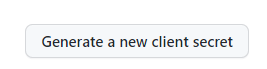
Cliquez sur le bouton Register Application. Une nouvelle page apparaît avec le GITHUB\_ID et le GITHUB\_SECRET :

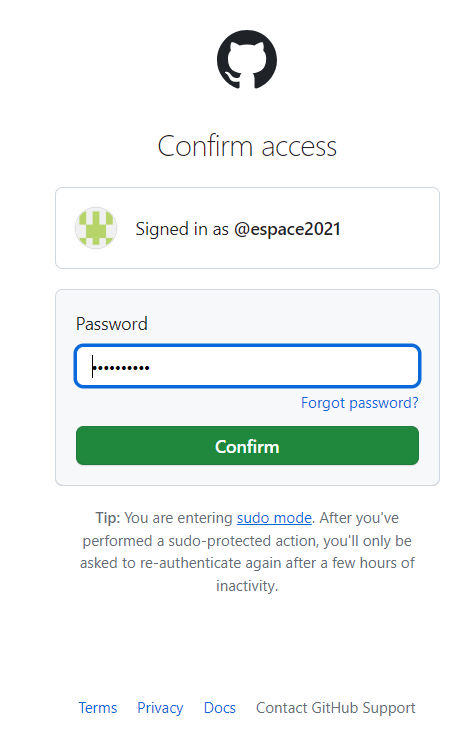


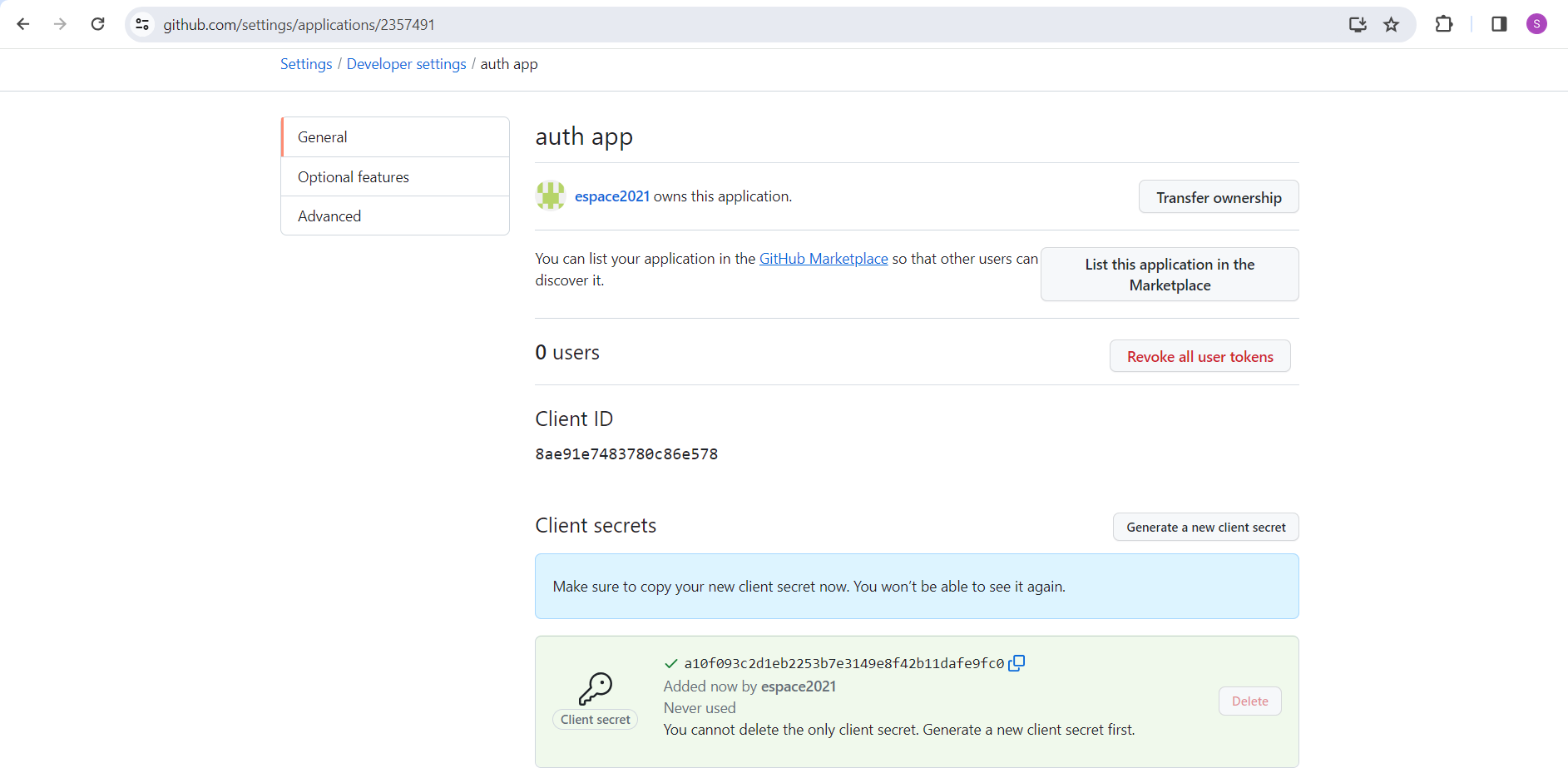
## Client ID

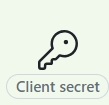
8ae91e7483780c86e578

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>

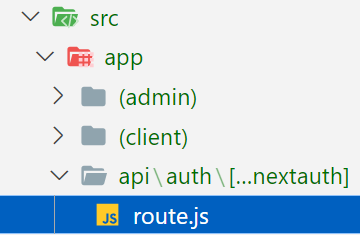
    </>

  )

}

## /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>

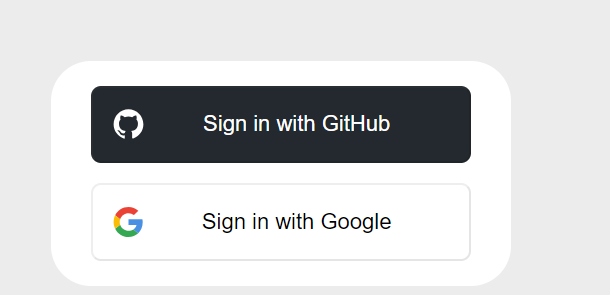
</nav>

);

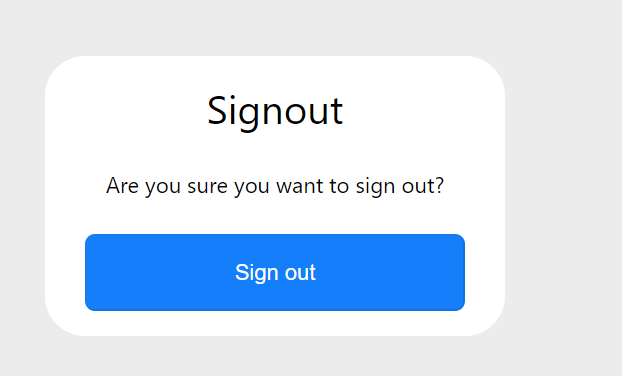
};

export default NavBar;

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



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



# Email Credentials

Vous pouvez utiliser n'importe quel fournisseur de messagerie proposant des services SMTP. Ces étapes utiliseront Gmail. Vous pouvez consulter Mailgun, SendGrid et autres.

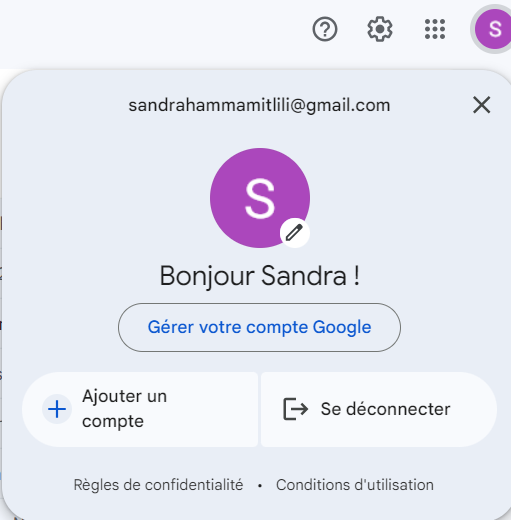
Vous aurez besoin des détails suivants :

* EMAIL\_SERVER\_HOST
* EMAIL\_SERVER\_PORT
* EMAIL\_SERVER\_USER
* EMAIL\_SERVER\_PASSWORD
* EMAIL\_FROM

Les étapes suivantes vous mèneront à votre EMAIL\_SERVER\_PASSWORD :

Connectez-vous à votre compte Gmail.

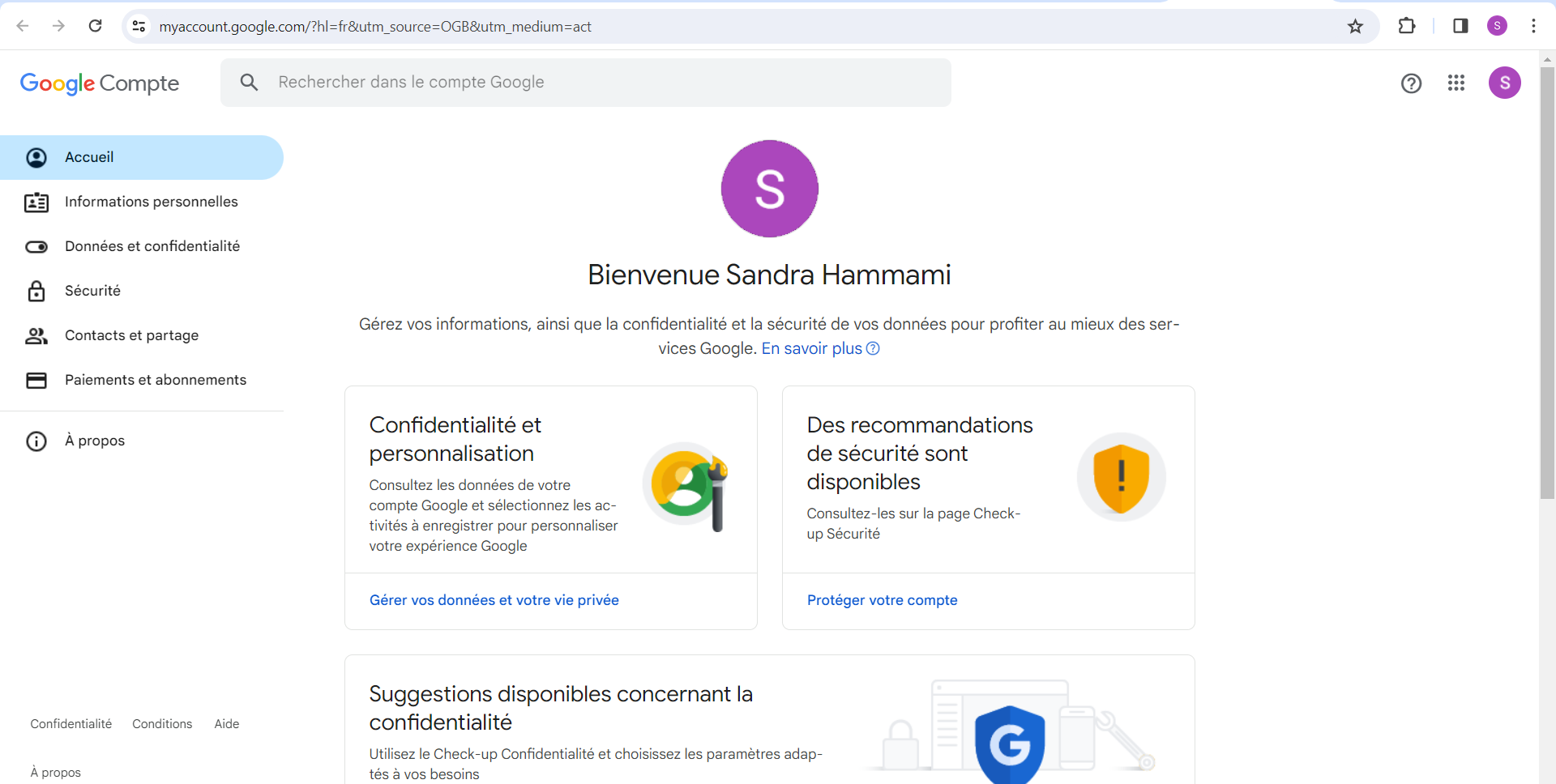
Cliquez sur votre image de profil. Il déroulera un menu.



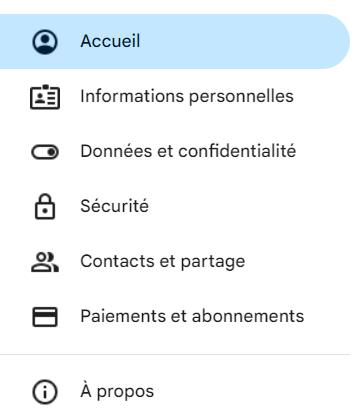
Cliquez sur le bouton Gérer votre compte Google



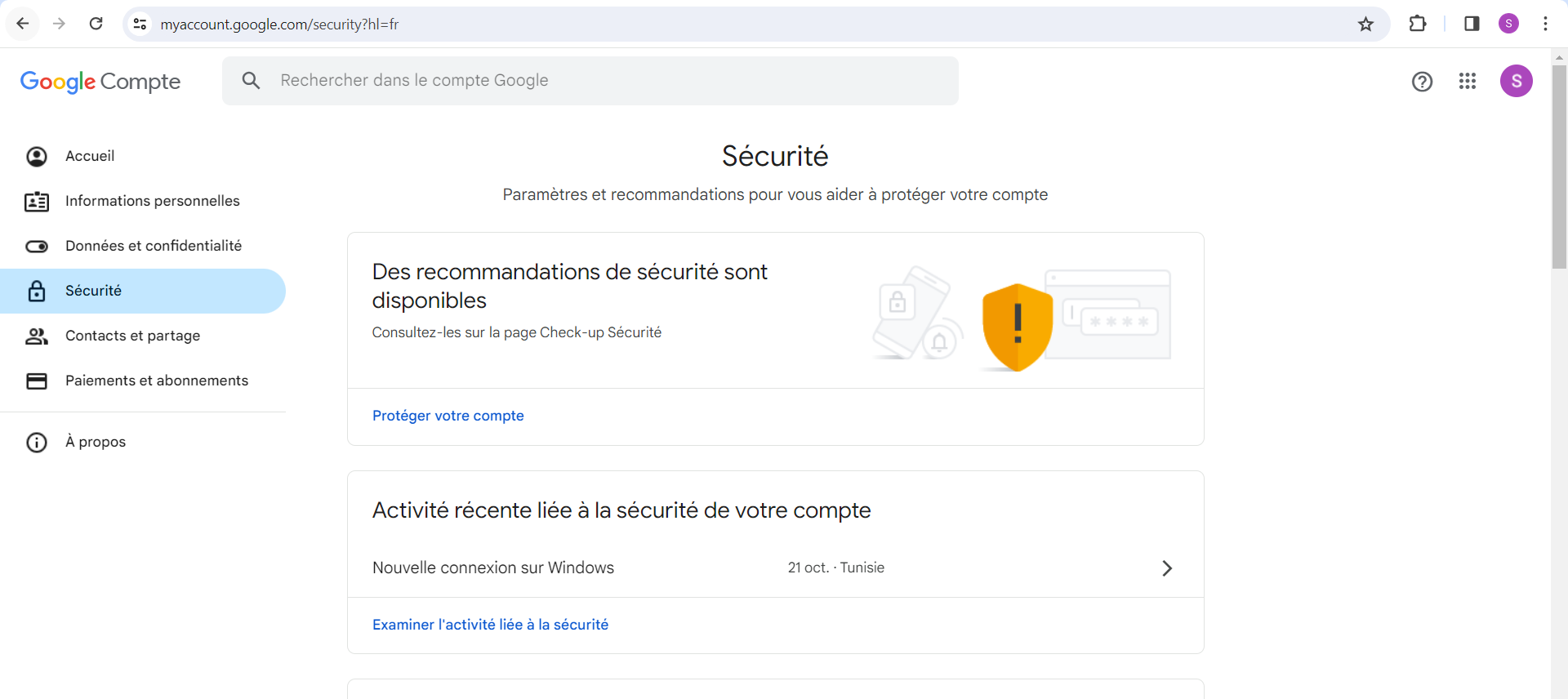
Vous serez redirigé vers la page ci-dessous :



Cliquez sur Sécurité dans le menu latéral.



La page suivante apparaîtra.

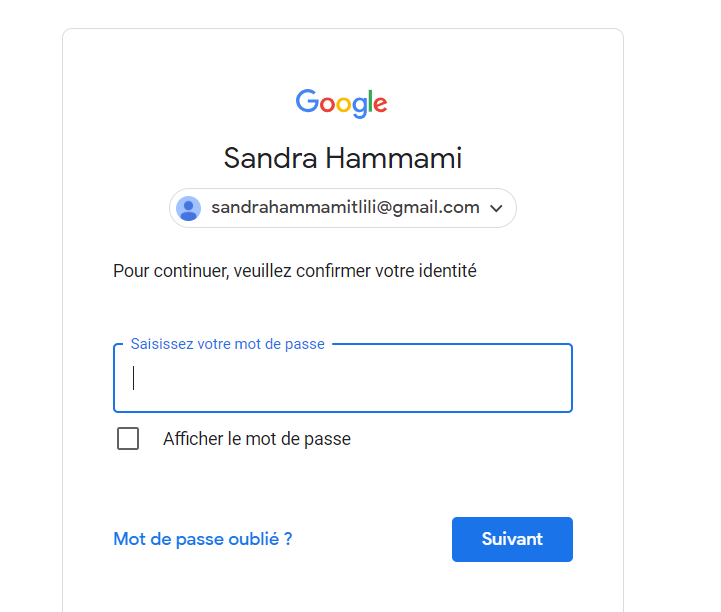


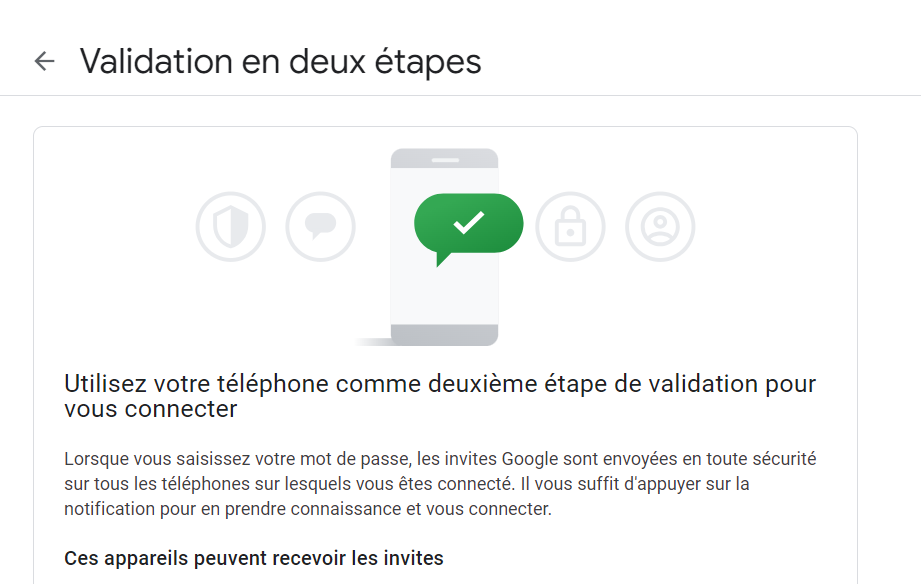
Vous devrez configurer la vérification en deux étapes pour que ce processus fonctionne.





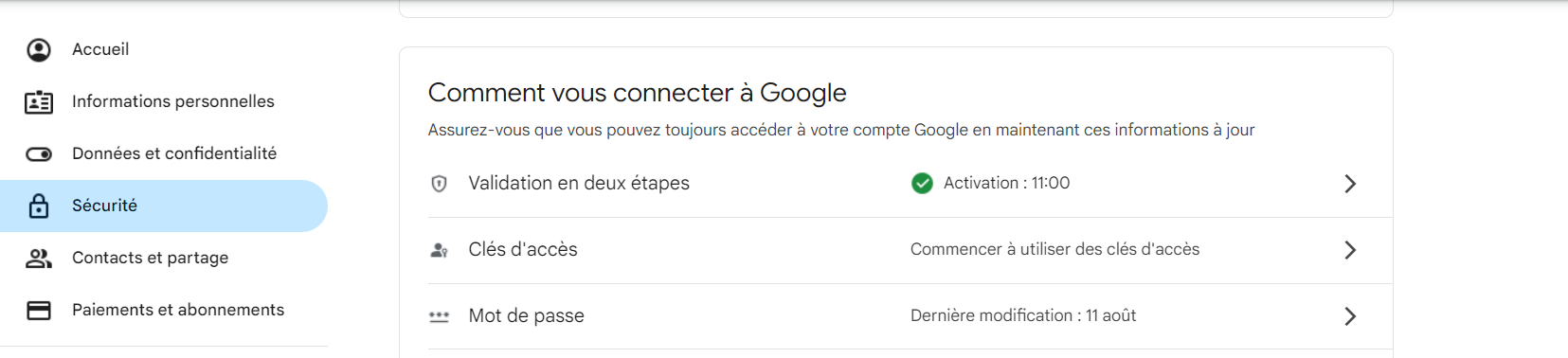
Cliquer sur commencer puis saisir le mot de passe







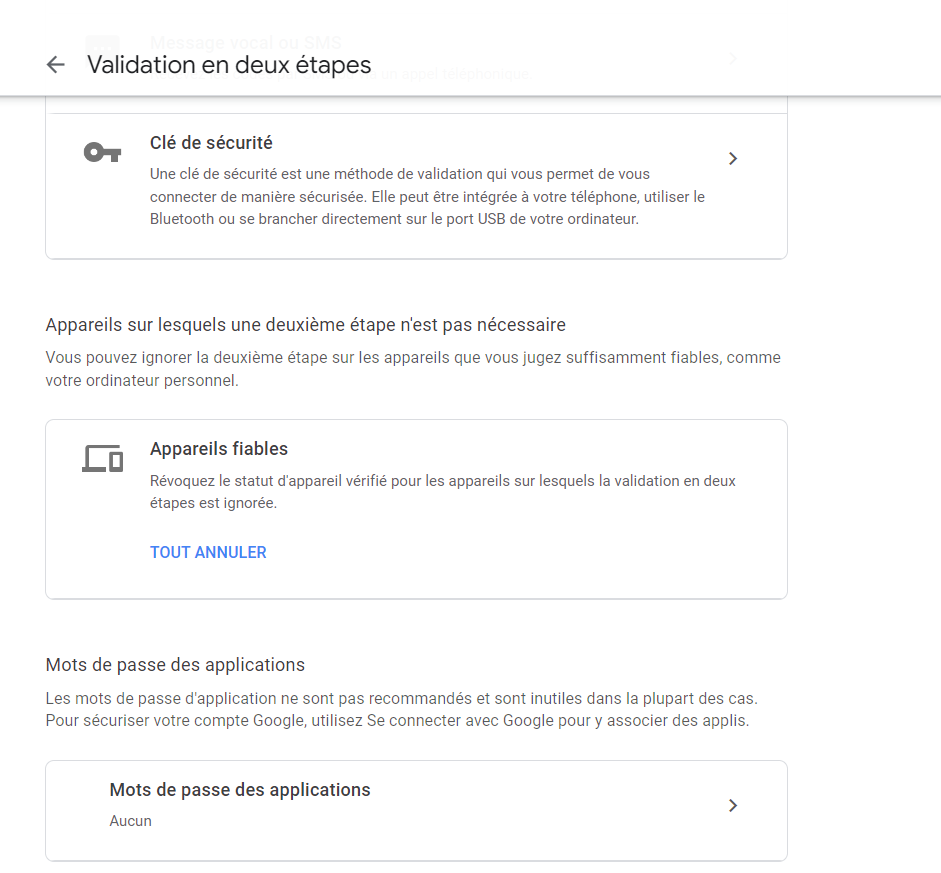
Maintenant dans la rubrique sécurité on voit que la validation en deux étapes est activée



Cliquer sur :



Après quoi vous serez redirigé vers la page suivante :

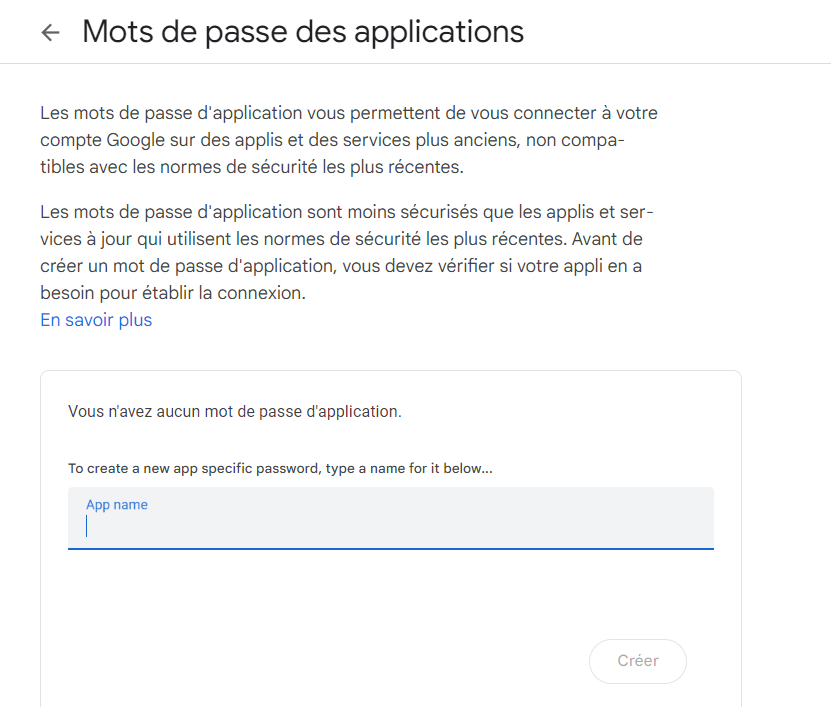


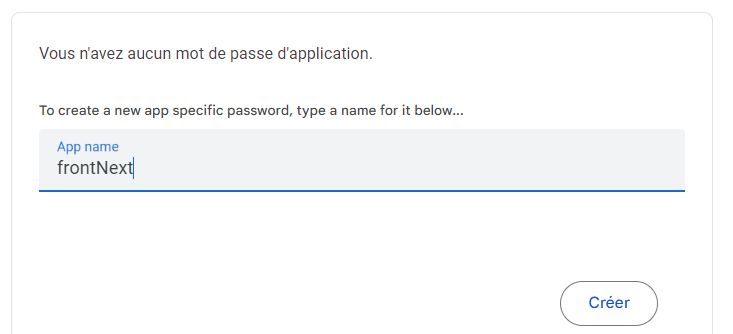
Faites défiler vers le bas et cliquez sur Mots de passe d'application, comme dans l'image ci-dessus.

Cliquer sur

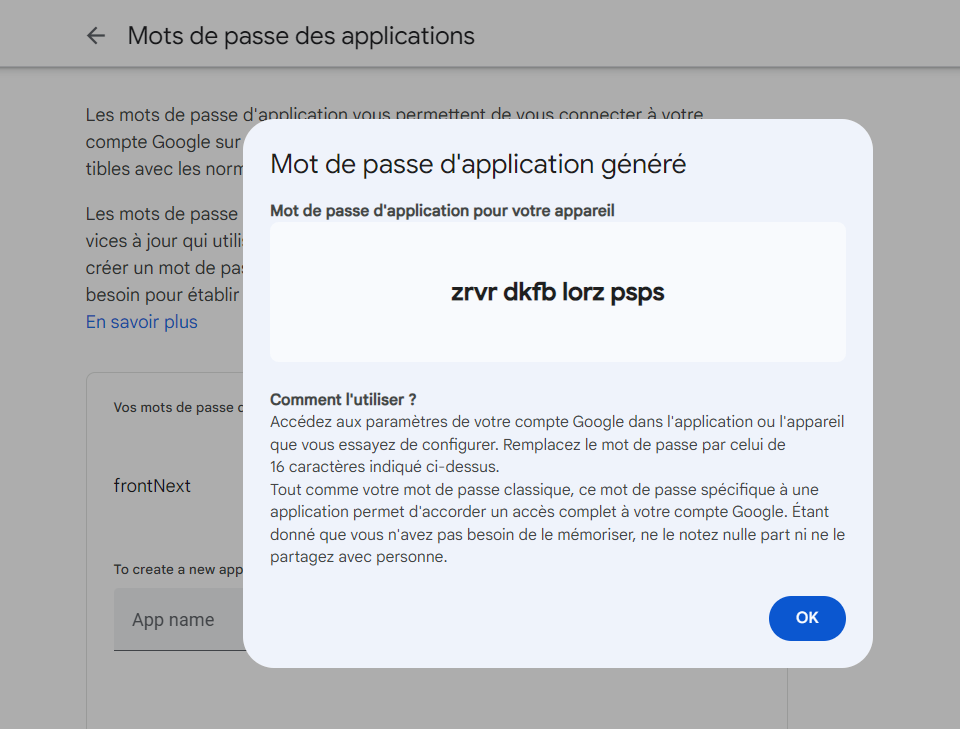


Ce qui affiche

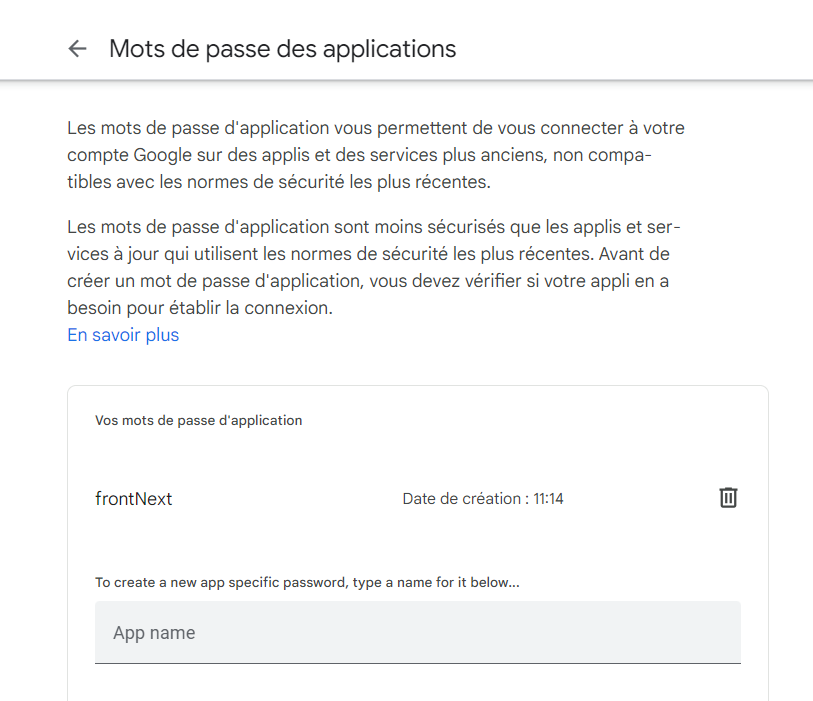




Cliquer sur créer



**zrvr dkfb lorz psps**



L’Email credentials sera comme suit :

EMAIL\_SERVER\_HOST="smtp.gmail.com"

EMAIL\_SERVER\_PORT=465

EMAIL\_SERVER\_USER="<email\_address>". Exemple “sandrahammamitlili@gmail.com”

EMAIL\_SERVER\_PASSWORD="zrvrdkfblorzpsps"

EMAIL\_FROM="<email\_address\_you\_want\_recipient\_to\_see\_as\_the\_sender>". Il n’est pas nécessaire que cet e-mail existe. Exemple “ABC@gmail.com”

## Installer package nodemailer

Pour utiliser NodeMailer avec NextAuth.js, vous devrez généralement configurer NodeMailer pour qu'il utilise un service SMTP de votre choix (par exemple, Gmail, SendGrid) et créer des fonctions pour envoyer des e-mails à partir de vos actions NextAuth.js, par exemple lors de l'inscription d'un nouvel utilisateur ou de la réinitialisation d'un mot de passe.

NodeMailer est souvent utilisé dans les applications NextAuth.js pour gérer les communications par e-mail liées à l'authentification, offrant ainsi un moyen flexible et personnalisable d'interagir avec les utilisateurs via des e-mails.

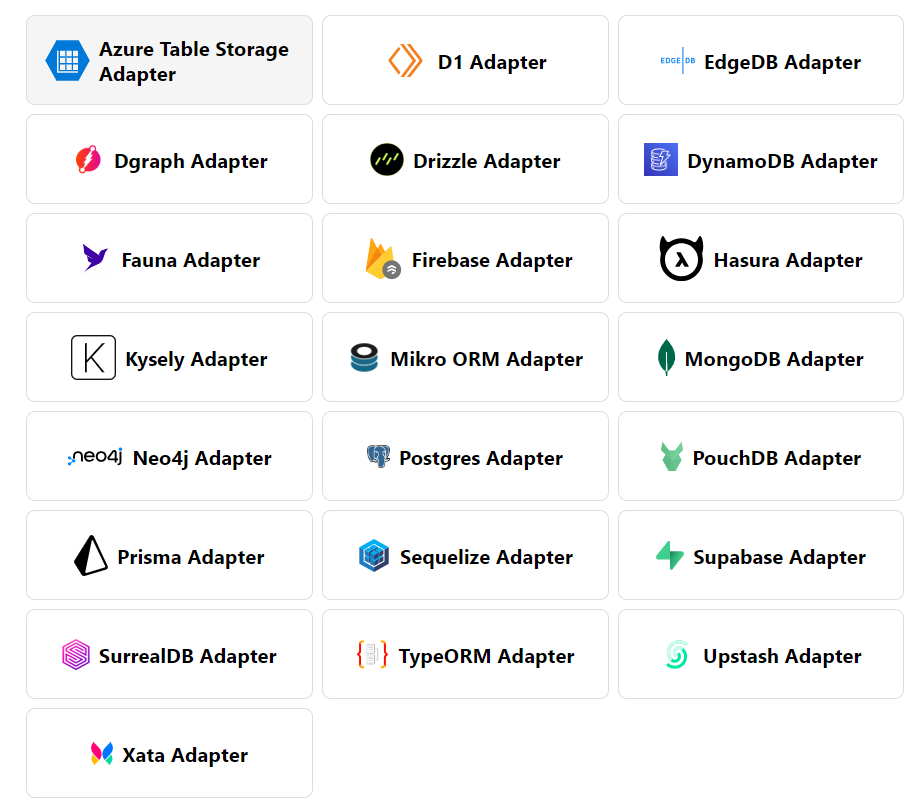
npm install nodemailer

## Adapter

Un adaptateur (ou adapter en anglais) est un composant logiciel qui sert à connecter ou à faire le lien entre deux parties d'un système informatique qui ont des interfaces incompatibles ou différentes. Les adaptateurs sont couramment utilisés pour rendre compatible l'interaction entre deux composants logiciels ou matériels qui ont été conçus de manière indépendante ou qui utilisent des protocoles différents.

Dans le contexte de l'authentification et de l'autorisation, un adaptateur peut être utilisé pour connecter un système d'authentification existant à une application.

À l'aide d'un adaptateur Auth.js / NextAuth.js, vous pouvez vous connecter à n'importe quel service de base de données ou même à plusieurs services différents en même temps. Les adaptateurs officiels répertoriés suivants sont créés et gérés par la communauté :



Nous allons choisir **MongoDB**

Installer package :

npm install @auth/mongodb-adapter mongodb

Créer .env :

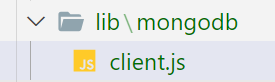


ENV=DEVELOPMENT

MONGODB\_URI\_LOCAL=mongodb://127.0.0.1:27017/DatabaseCommerce

MONGODB\_URI=mongodb+srv://hassan:messk2015@cluster0.yfhhu.mongodb.net/DatabaseCommerce?retryWrites=true&w=majority

Puis créer un simple MongoDB client. Créer dossier lib **sous src**, dans lequel créer le dossier mongodb. Puis y créer le fichier client.js



import { MongoClient } from "mongodb"

const uri = process.env.MONGODB\_URI

const options = {

  useUnifiedTopology: true,

  useNewUrlParser: true,

}

let client

let clientPromise

if (!process.env.MONGODB\_URI) {

  throw new Error("Please add your Mongo URI to .env.local")

}

if (process.env.NODE\_ENV === "development") {

  // In development mode, use a global variable so that the value

  // is preserved across module reloads caused by HMR (Hot Module Replacement).

  if (!global.\_mongoClientPromise) {

    client = new MongoClient(uri, options)

    global.\_mongoClientPromise = client.connect()

  }

  clientPromise = global.\_mongoClientPromise

} else {

  // In production mode, it's best to not use a global variable.

  client = new MongoClient(uri, options)

  clientPromise = client.connect()

}

// Export a module-scoped MongoClient promise. By doing this in a

// separate module, the client can be shared across functions.

export default clientPromise

## Modifier [..nextauth]/route.js



import NextAuth from "next-auth";

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

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

import EmailProvider from "next-auth/providers/email";

import clientPromise from '@/lib/mongodb/client';

import { MongoDBAdapter } from "@auth/mongodb-adapter";

export const authOptions = {

    adapter: MongoDBAdapter(clientPromise),

    providers: [

        EmailProvider({

            server: {

              host: "smtp.gmail.com",

              port: 587,

              auth: {

                user: "sandrahammamitlili@gmail.com",

                pass: "zrvrdkfblorzpsps",

              },

            },

            from: "ABC@gmail.com",

          }),

        GitHubProvider({

            clientId: "8ae91e7483780c86e578",

            clientSecret: "a10f093c2d1eb2253b7e3149e8f42b11dafe9fc0",

          }),

        GoogleProvider({

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

            clientSecret: "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"

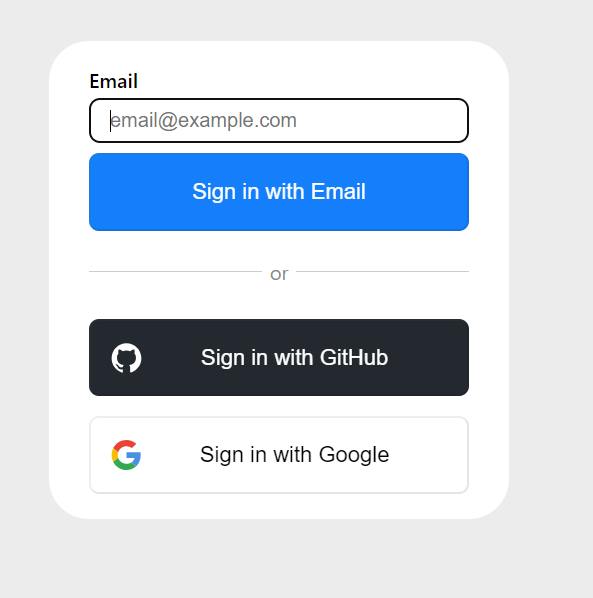
        })

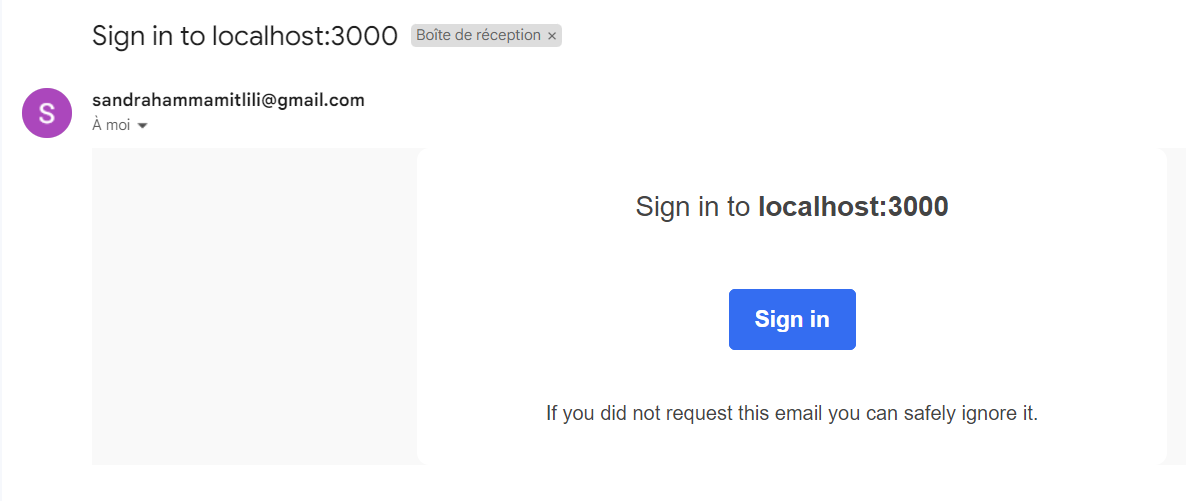
    ],

}

const handler = NextAuth(authOptions)

export { handler as GET, handler as POST }







import React from 'react'

import Link from "next/link";

const page = () => {

  return (

    <div>

      <h1>APPLICATION</h1>

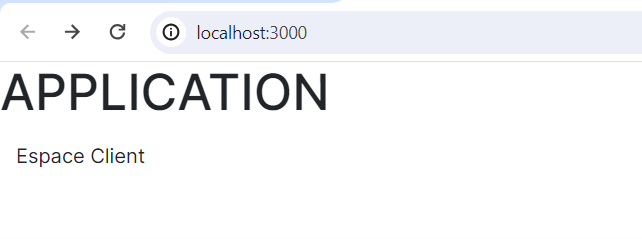
      <Link className="btn btn-default" href={'/client'}> Espace Client </Link>

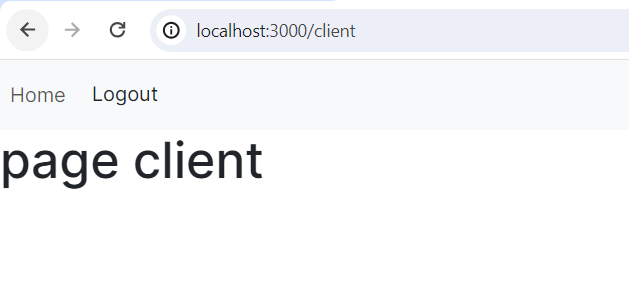
    </div>

  )

}

export default page





# Shopping Cart

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**



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

import './globals.css'

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

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

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

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

 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>

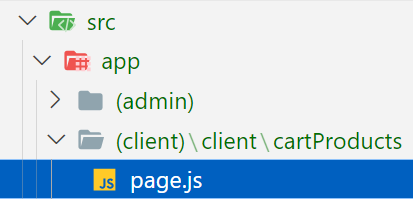
  )

}

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} />

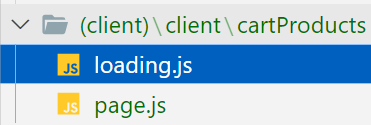
     </>

    )

}

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>

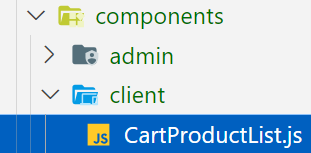
)

}

export default loading

# Component CartProductList

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



import CartProductItem from "./CartProductItem";

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

       return (

        <div className="row">

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

              <CartProductItem key={product?.\_id} product={product} />

            ))}

          </div>

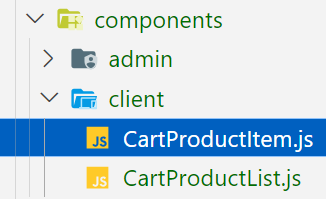
       );

}

export default CartProductList;

# Component CartProductItem

Sous le dossier /components/client/ créer le composant CartProductItem.js



"use client";

import React,{ useState } from "react";

const CartProductItem= ({product})=> {

    const [quantity, setQuantity] = useState(1);

    const decreaseQuantity = () => {

      if (quantity > 1) {

        setQuantity(quantity - 1);

      }

    };

    const increaseQuantity = () => {

      setQuantity(quantity + 1);

    };

    return (

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

  <div className="card">

    <img

      src={product?.imageart}

      className="card-img-top p-5"

      alt={product.designation}

    />

  </div>

  <div>

 <div>{product.designation}</div>

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

 <div>

    <button

      onClick={decreaseQuantity}

      className="btn btn-default"

    >

      -

    </button>

    <span>{quantity}</span>

    <button

      onClick={increaseQuantity}

      className="btn btn-default"

    >

      +

    </button>

  </div>

  </div>

  <div className="text-center">

  <button className="btn btn-warning">

    Add to cart

  </button>

  </div>

</article>

    );

}

export default CartProductItem;

# 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>

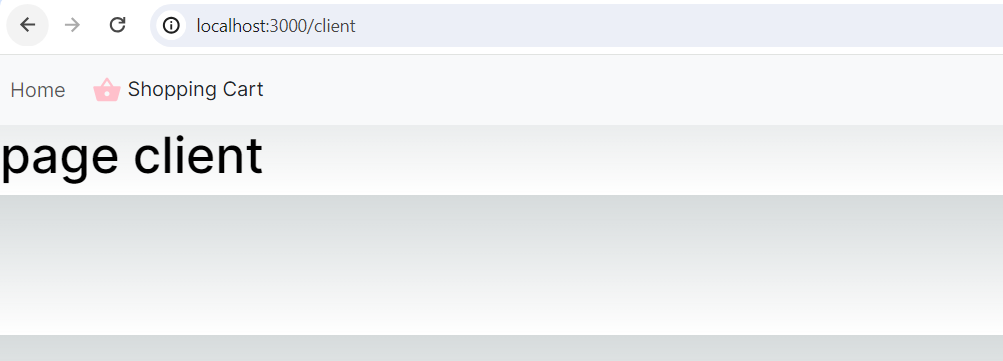
</nav>

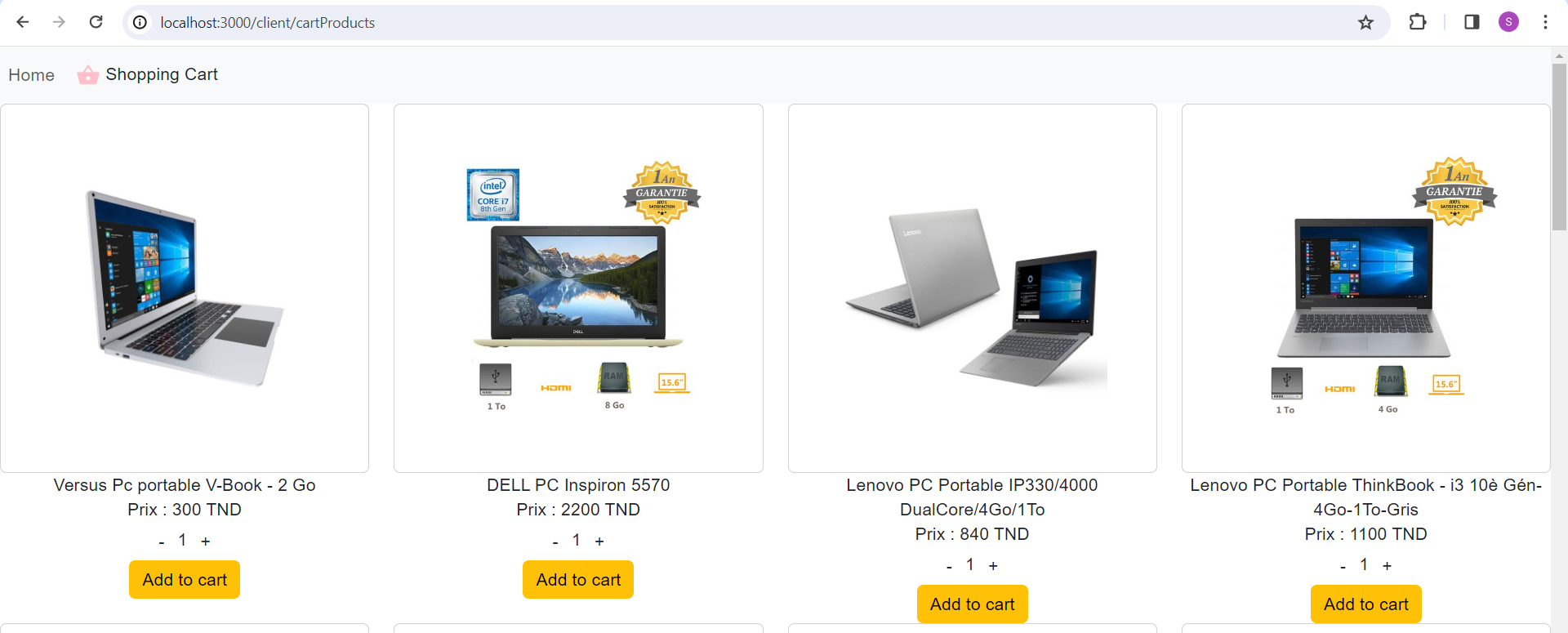
);

};

export default NavBar;

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





# Protéger la page



"use client";

import React,{ useState } from "react";

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

import { redirect } from 'next/navigation'

const CartProductItem= ({product})=> {

  const { data: session } = useSession({

    required: true,

    onUnauthenticated() {

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

    }

  })

    const [quantity, setQuantity] = useState(1);

    const decreaseQuantity = () => {

      if (quantity > 1) {

        setQuantity(quantity - 1);

      }

    };

    const increaseQuantity = () => {

      setQuantity(quantity + 1);

    };

    return (

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

  <div className="card">

    <img

      src={product?.imageart}

      className="card-img-top p-5"

      alt={product.designation}

    />

  </div>

  <div>

 <div>{product.designation}</div>

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

 <div>

    <button

      onClick={decreaseQuantity}

      className="btn btn-default"

    >

      -

    </button>

    <span>{quantity}</span>

    <button

      onClick={increaseQuantity}

      className="btn btn-default"

    >

      +

    </button>

  </div>

  </div>

  <div className="text-center">

  <button className="btn btn-warning">

    Add to cart

  </button>

  </div>

</article>

    );

}

export default CartProductItem;

## signIn

Sous le répertoire /src/app/(user) 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 })

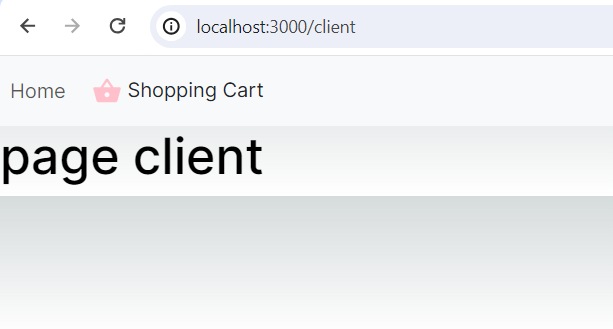
  return (

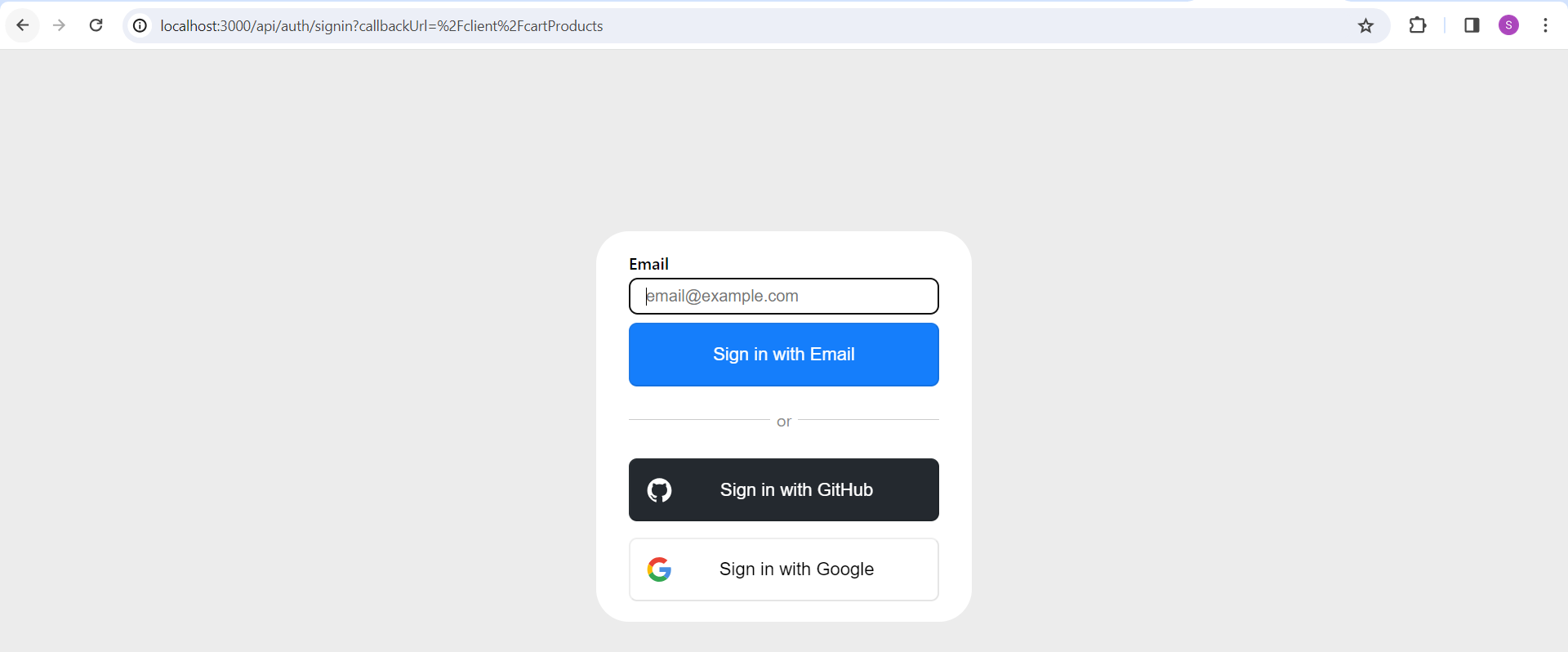
<></>

  )

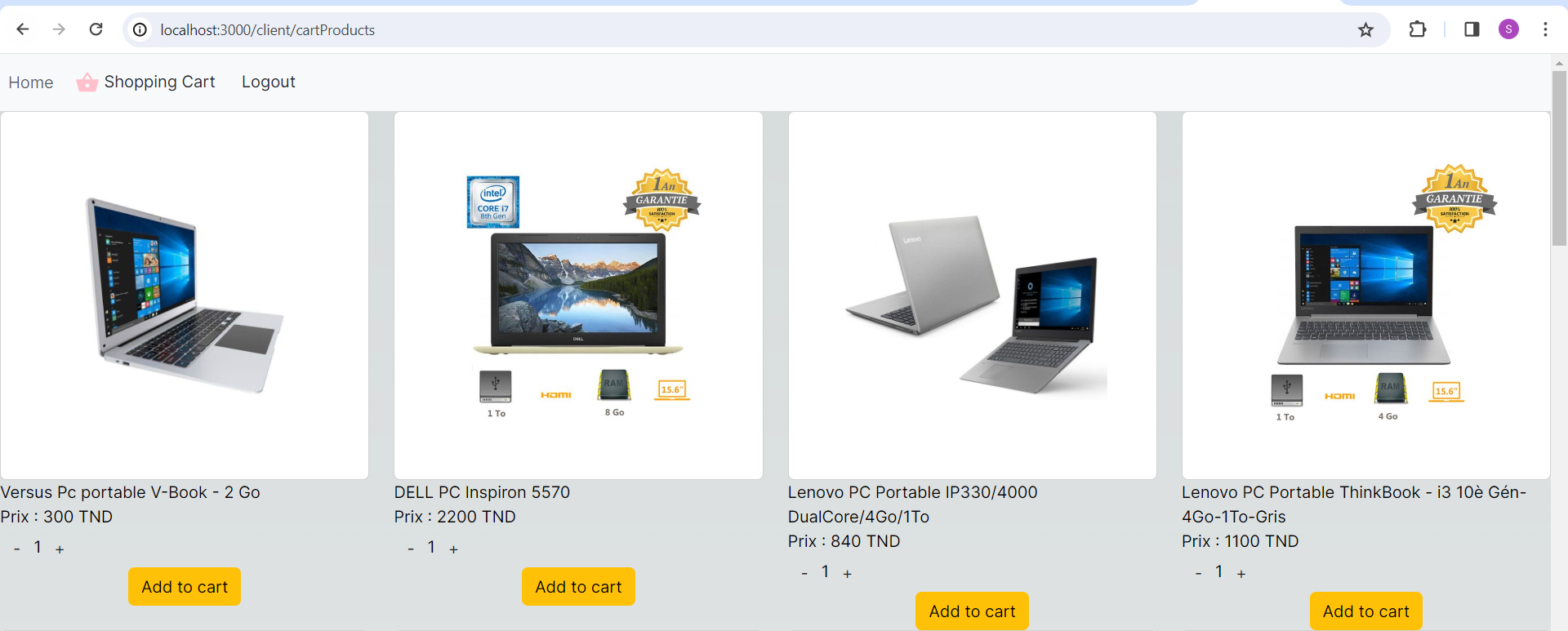
}

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

