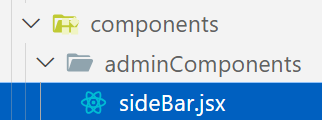
Atelier 6 NextJS : Authentification

# SideBar menu

Créer sous le répertoire /src/components un dossier intitulé adminComponents

Dans ce dossier créer un fichier appelé sideBar.jsx



On va se référer à ce site pour le code correspondant à ce composant :

<https://www.npmjs.com/package/react-pro-sidebar>

Installer le package :

npm i react-pro-sidebar

Les icônes à partir de ce site :

<https://mui.com/material-ui/material-icons/>

'use client';

import { Sidebar, Menu, MenuItem } from 'react-pro-sidebar';

import Link from 'next/link';

import AdminPanelSettingsIcon from '@mui/icons-material/AdminPanelSettings';

import DashboardIcon from '@mui/icons-material/Dashboard';

import ArticleIcon from '@mui/icons-material/Article';

import CategoryIcon from '@mui/icons-material/Category';

import ContentCopyIcon from '@mui/icons-material/ContentCopy';

import GroupIcon from '@mui/icons-material/Group';

import ReceiptIcon from '@mui/icons-material/Receipt';

import CalendarMonthIcon from '@mui/icons-material/CalendarMonth';

import BarChartIcon from '@mui/icons-material/BarChart';

import LightModeIcon from '@mui/icons-material/LightMode';

const SideBar = () => {

  return (

    <Sidebar>

        <Menu

            menuItemStyles={{

            button: ({ level, active, disabled }) => {

                // only apply styles on first level elements of the tree

                if (level === 0)

                return {

                    color: disabled ? '#b4c2c0' : '#718280',

                    backgroundColor: active ? '#4f5d73' : '#071121',

                };

            },

            }}

        >

      <MenuItem>

<div className="card" style={{"backgroundColor": "#FF00FF"}}>

        <div className="card-body">

          <AdminPanelSettingsIcon /> Administrateur

        </div>

      </div>

      </MenuItem>

<MenuItem component={<Link href="/admin/dashboard" />}> <DashboardIcon/> Dashboard </MenuItem>

      <MenuItem component={<Link href="/admin/livres" />}> <ArticleIcon/> Livres </MenuItem>

      <MenuItem component={<Link href="/admin/auteurs" />}> <CategoryIcon/> Auteurs </MenuItem>

      <MenuItem component={<Link href="/admin/editeurs" />}> <ContentCopyIcon/> Editeurs</MenuItem>

      <MenuItem component={<Link href="#" />}> <ReceiptIcon/> Spécialités </MenuItem>

      <MenuItem component={<Link href="#" />}> <GroupIcon/> Utilisateurs </MenuItem>

      <MenuItem component={<Link href="#" />}> <CalendarMonthIcon/> Calendrier </MenuItem>

      <MenuItem component={<Link href="#" />}> <BarChartIcon/> Chart </MenuItem>

      <MenuItem component={<Link href="#" />}> <LightModeIcon/> Thème </MenuItem>

    </Menu>

  </Sidebar>

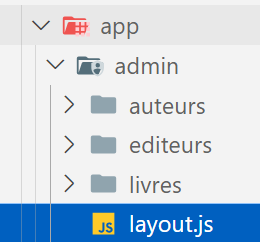
  )

}

export default SideBar

Ensuite, créer sous /src/app/admin puis le fichier layout.js

Layout est une interface utilisateur partagée entre les routes.



import SideBar from "@/components/adminComponents/sideBar";

export default function AdminLayout({ children }) {

  return (

    <div className="row mt-4">

    <div className="col-md-2">

    <SideBar />

    </div>

    <div className="col-md-10">

    {children}

    </div>

    </div>

  )

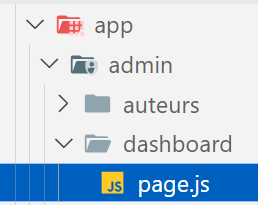
}

Ainsi, le sideBar sera visible dans tous les enfants de admin. On lui réservé 3 colonnes et les 9 autres pour l’enfant qui sera appelé via les de sidebar.

Avec mt-4 on a défini une marge en haut égale à 4.

# Page Dashboard

Par la suite, sous /src/admin, créer un dossier dashbord. Ce dernier contenant un fichier page.js



import React from 'react'

const page = () => {

  return (

  <div className="container">

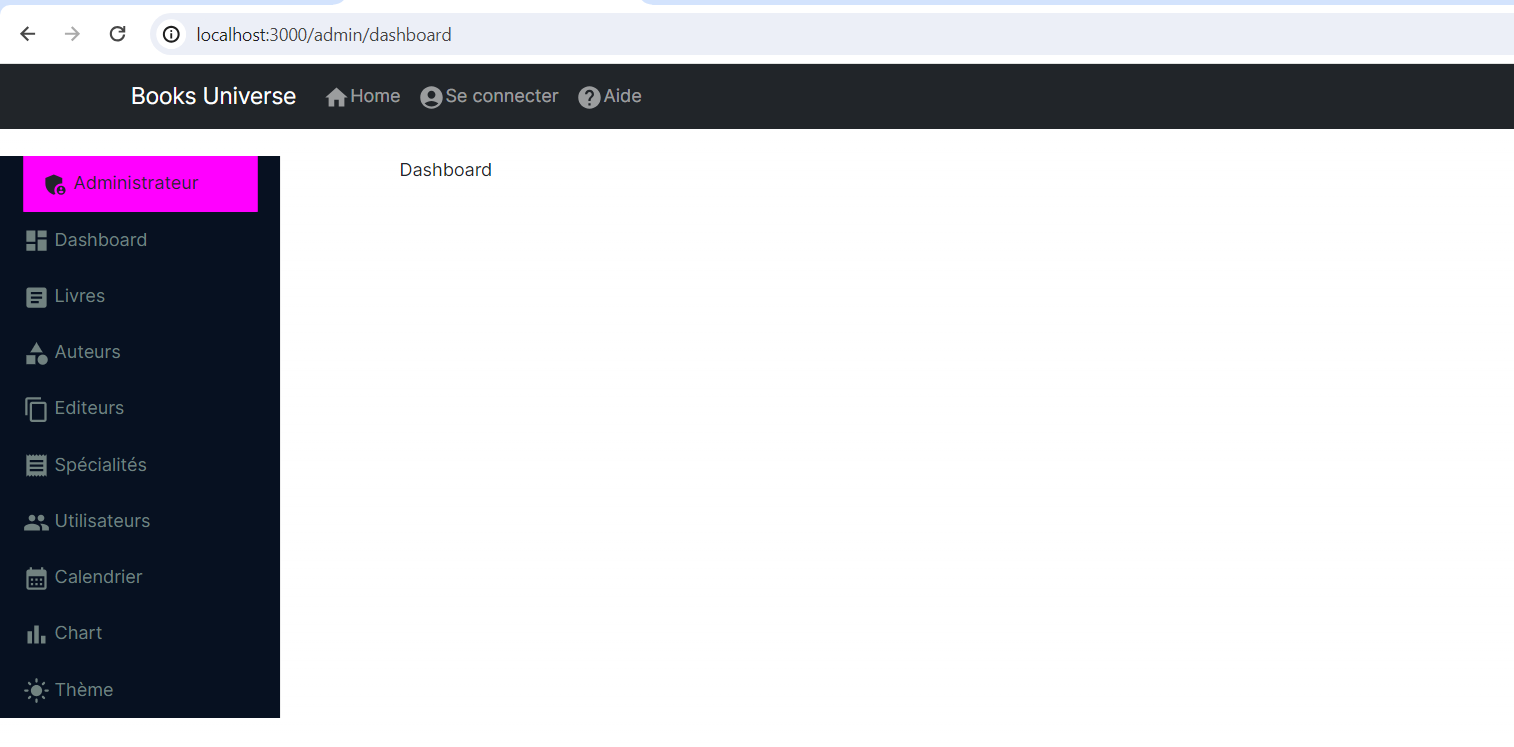
    Dashboard

</div>

  )

}

export default page



# Next-auth

Lorsque vous souhaitez un système d’authentification dans votre application Next.js, NextAuth.js est une excellente solution pour apporter la sécurité sans avoir à le construire vous-même. Il vient avec une longue liste de fournisseurs pour ajouter rapidement de l’authentification OAuth et fournit des adaptateurs pour de nombreuses bases de données et ORM.

Site : <https://next-auth.js.org/>

Référence : <https://next-auth.js.org/configuration/initialization#route-handlers-app>

## Package next-auth

Dans le projet next installer

npm i next-auth

# Provider Google

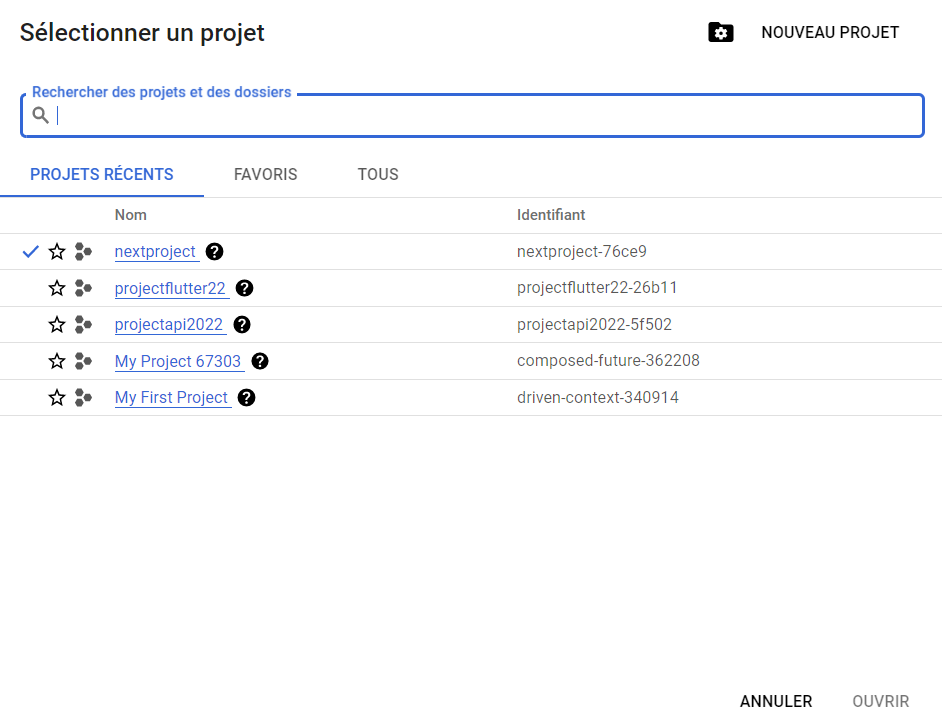
Avant de commencer à intégrer la connexion Google dans notre application, nous devrons d'abord obtenir un identifiant client Google.

Pour obtenir un identifiant client de Google, accédez à :

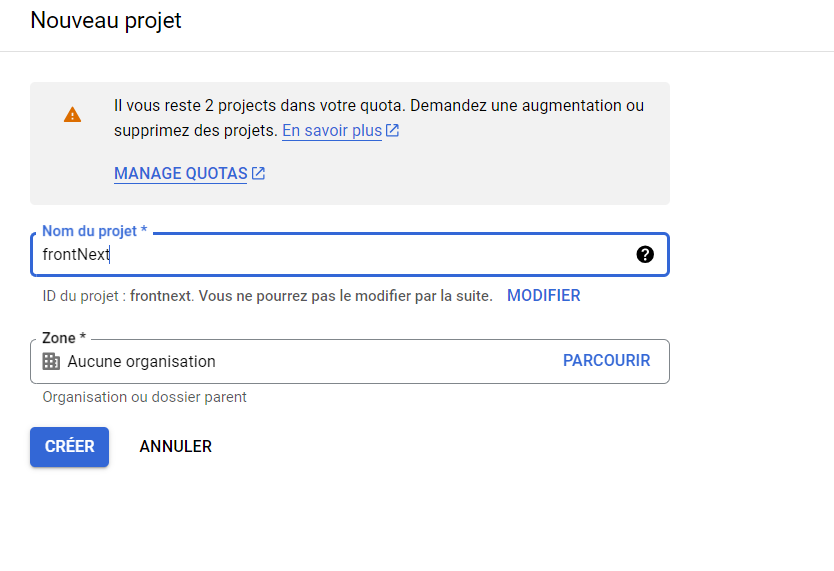
<https://console.cloud.google.com/>



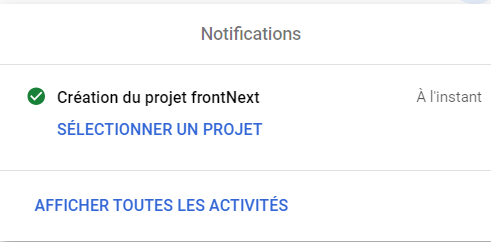
Sélectionnez un projet ou en créer un nouveau



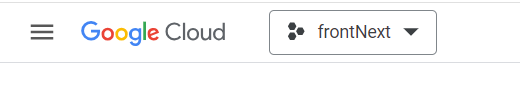




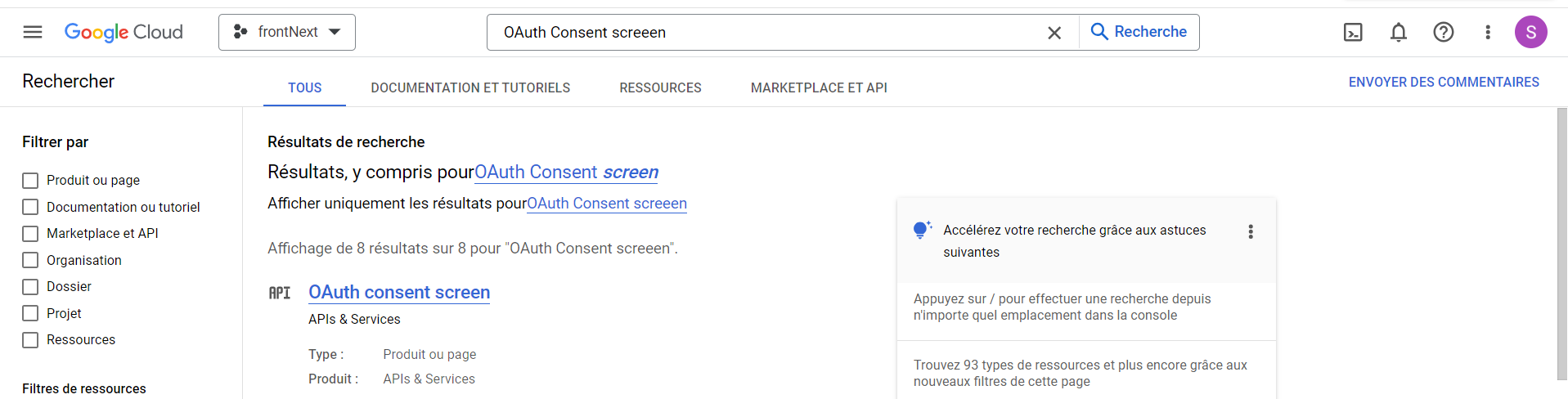


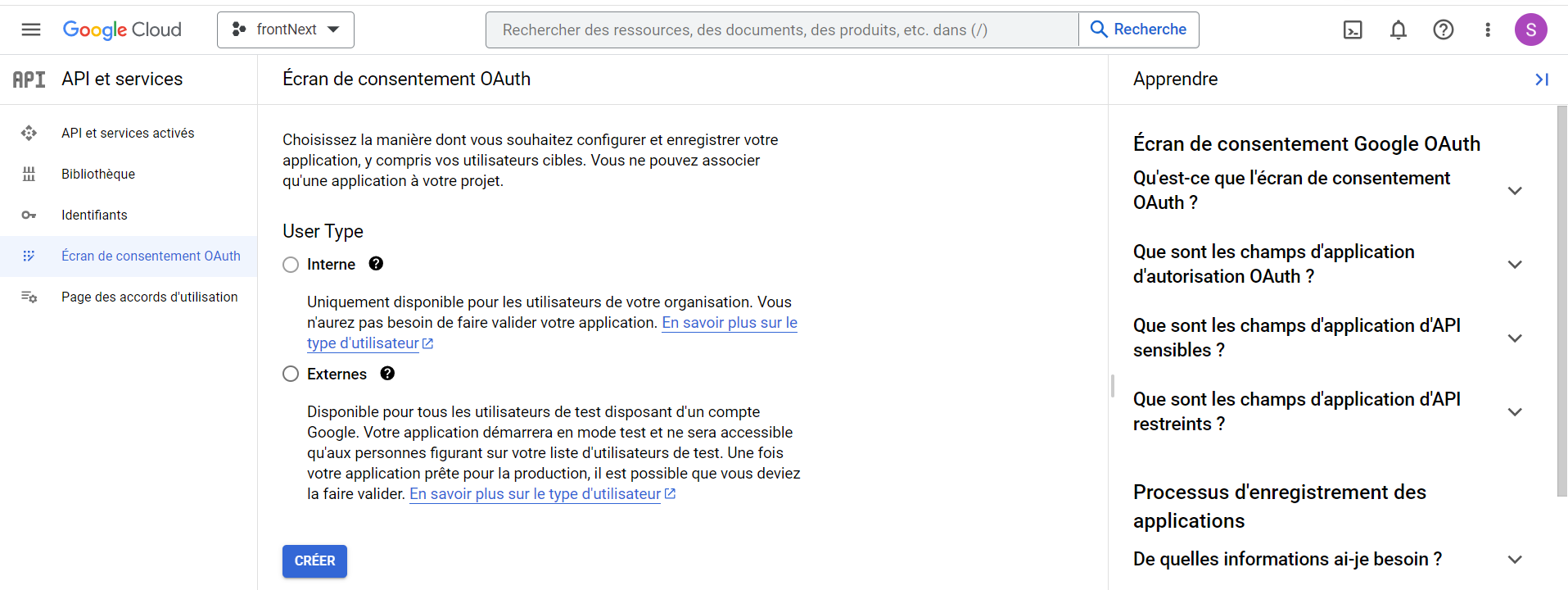


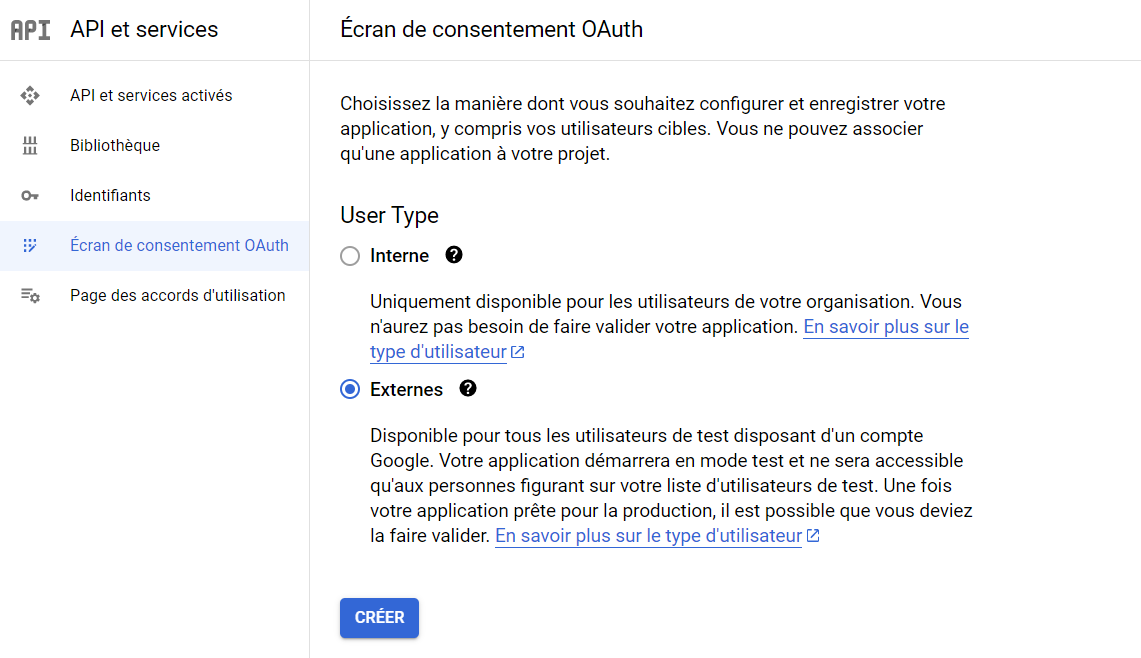


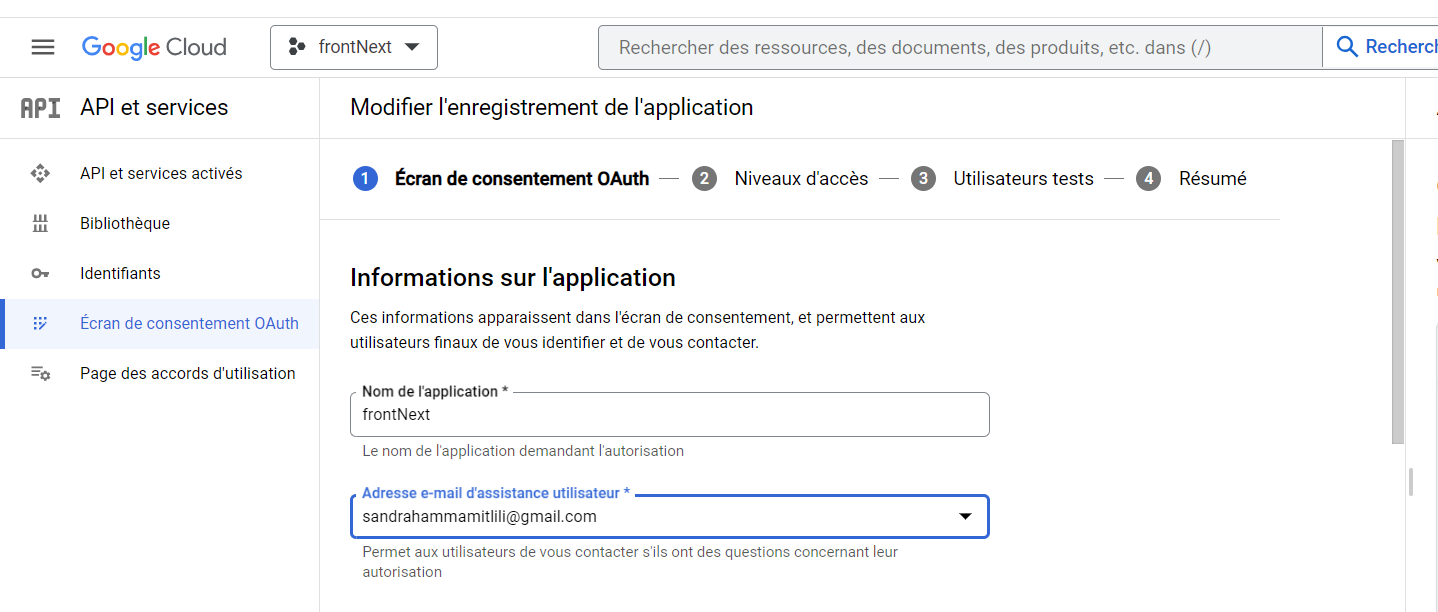


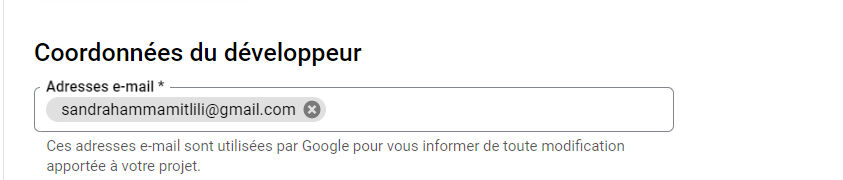




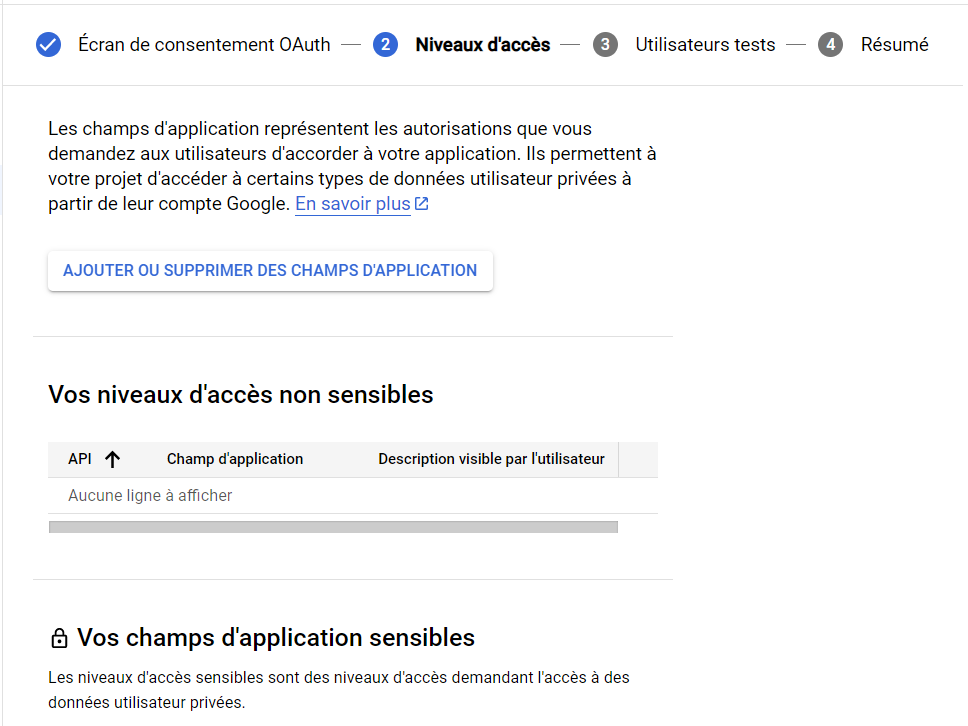


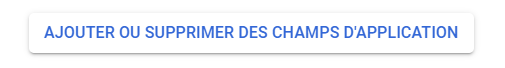


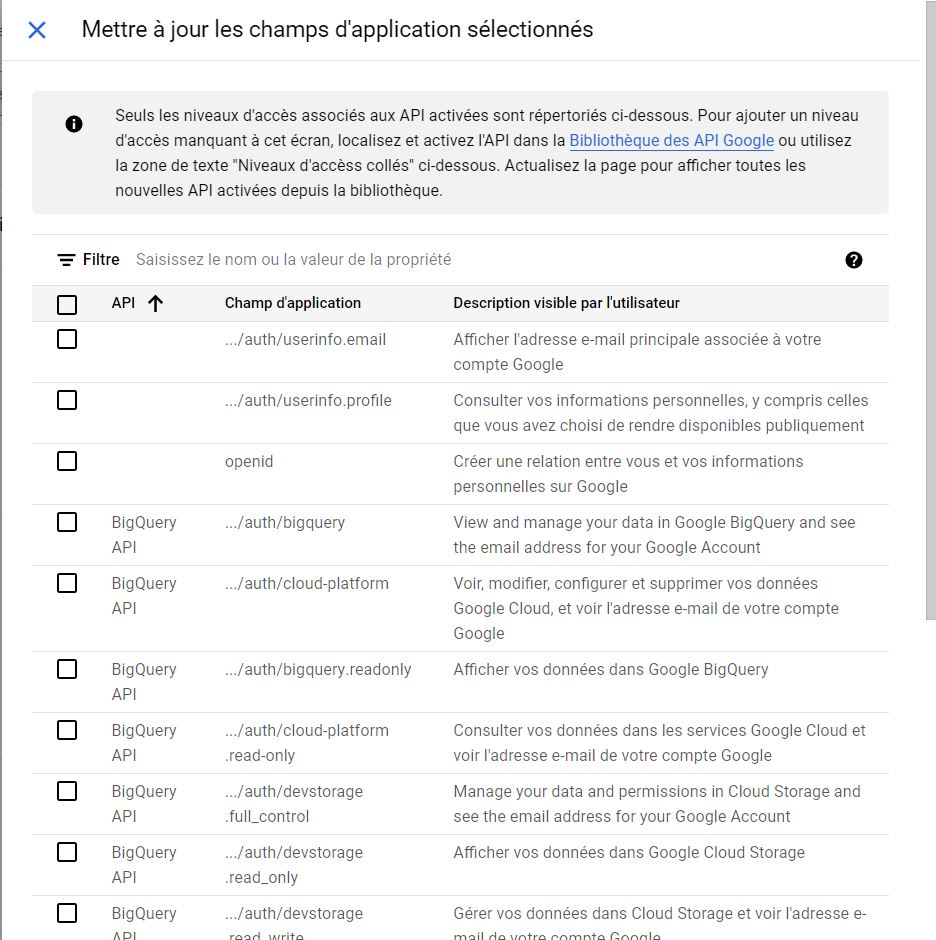


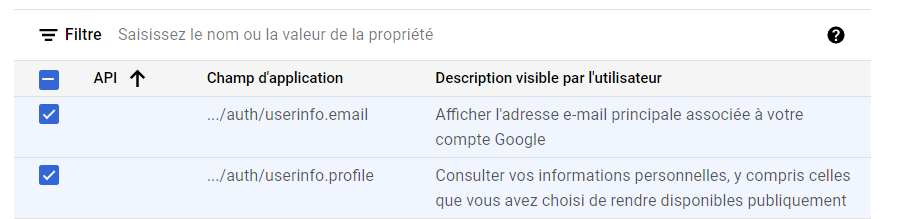


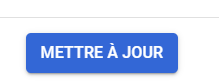


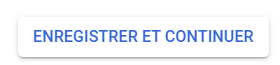


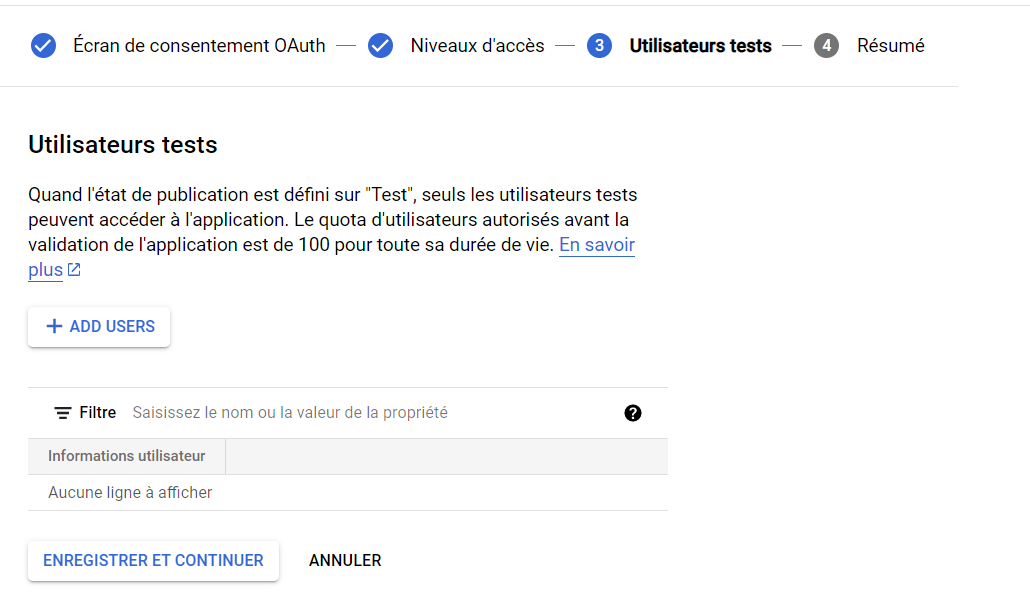


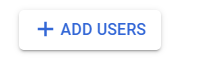


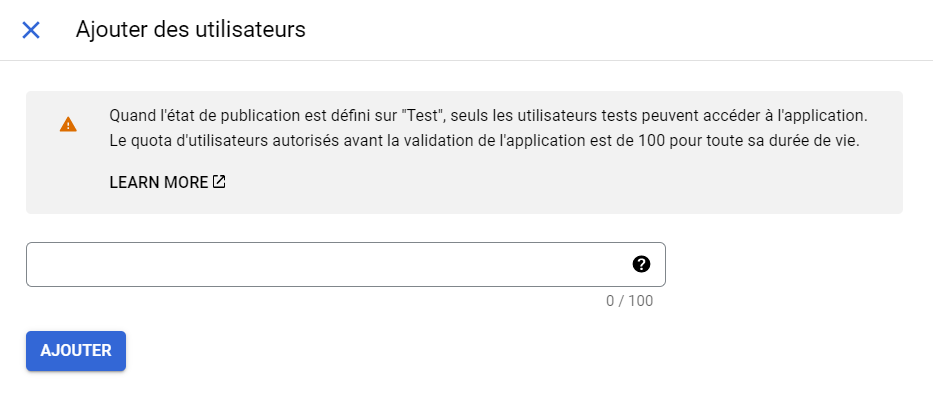








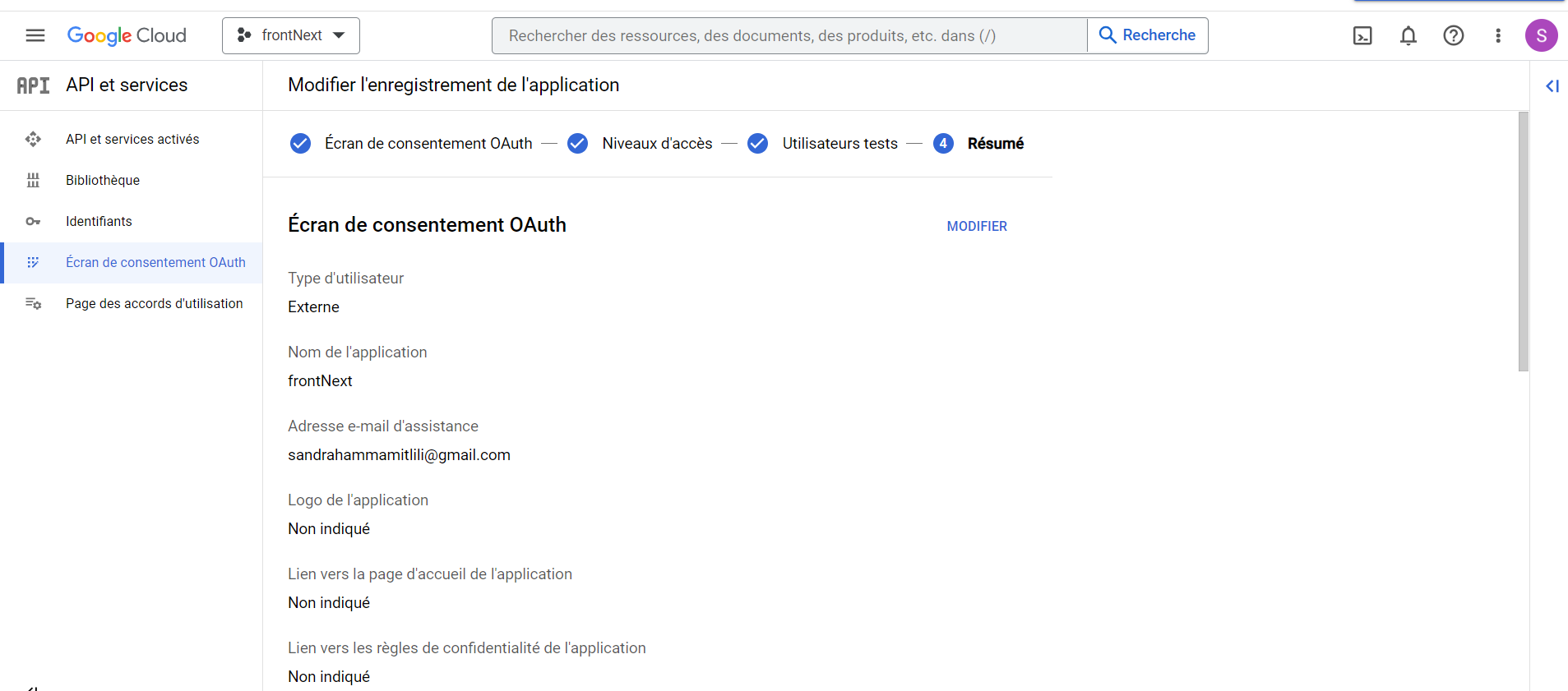


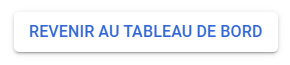


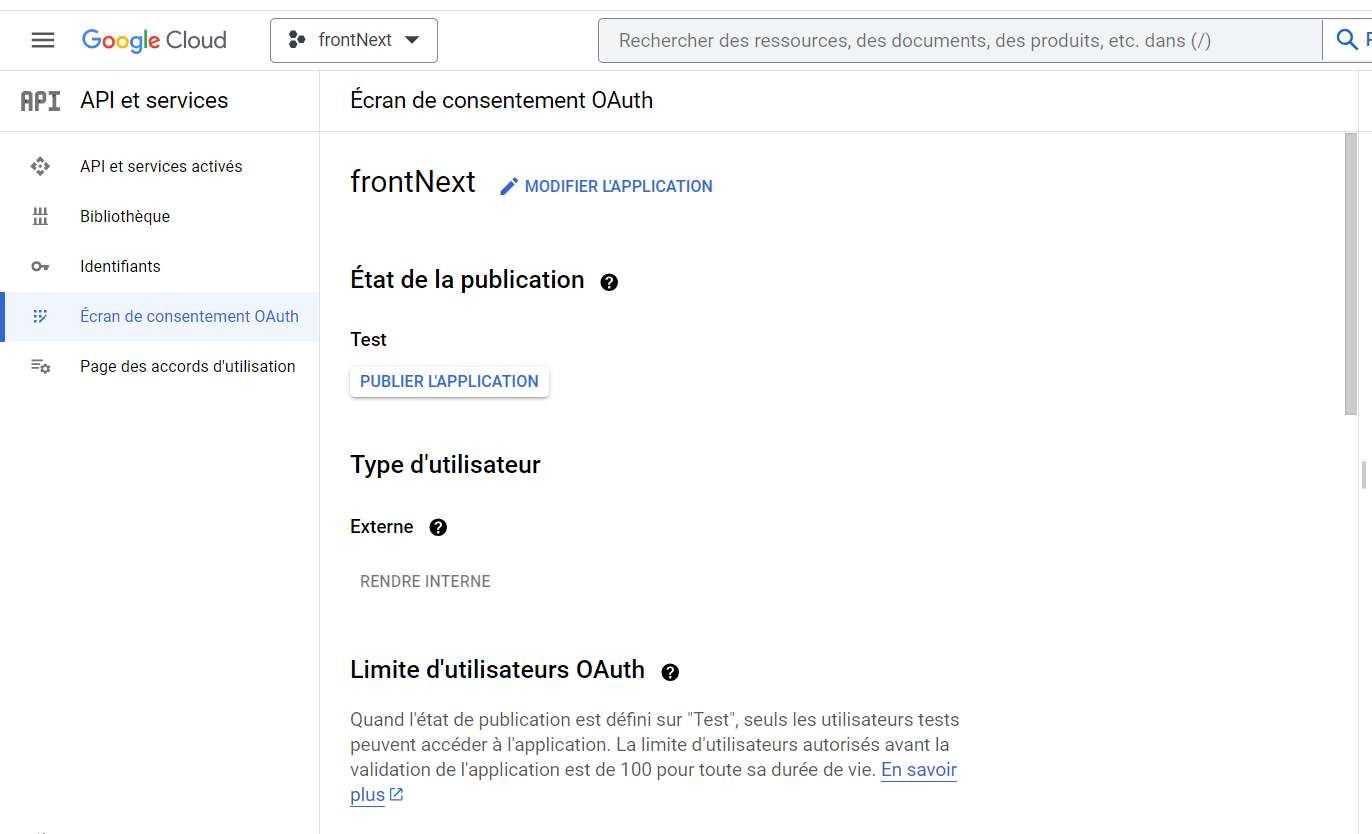




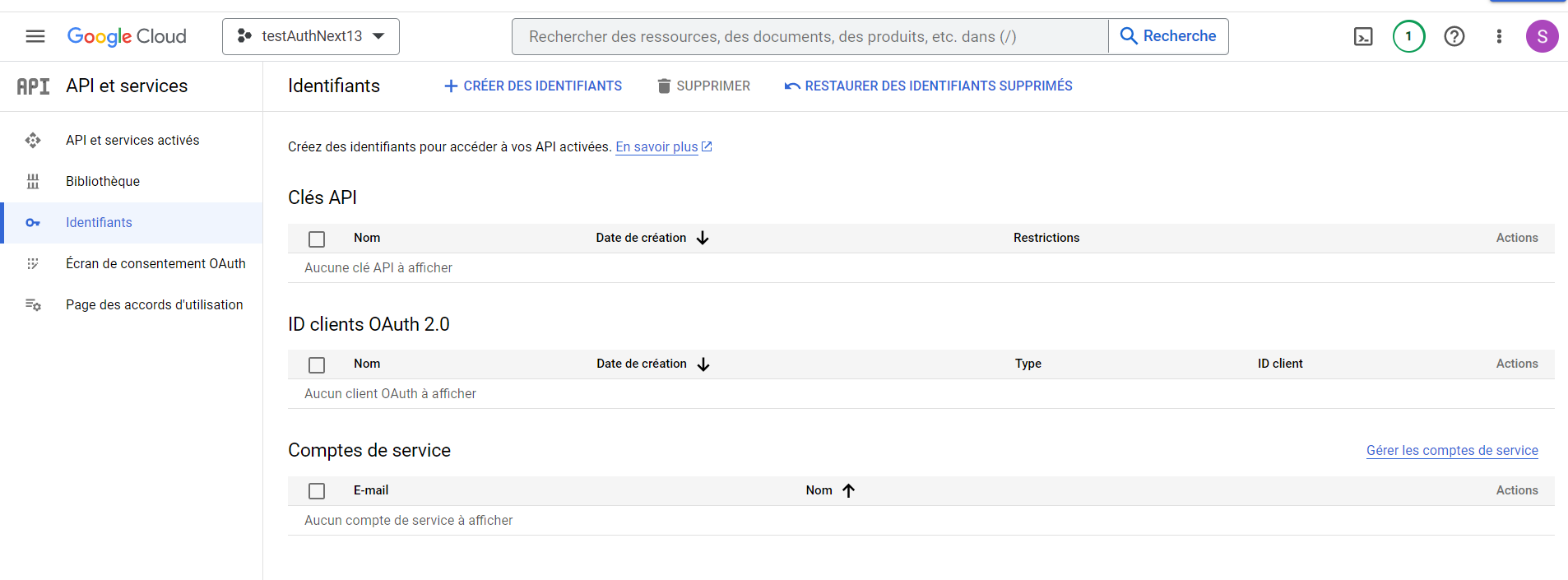




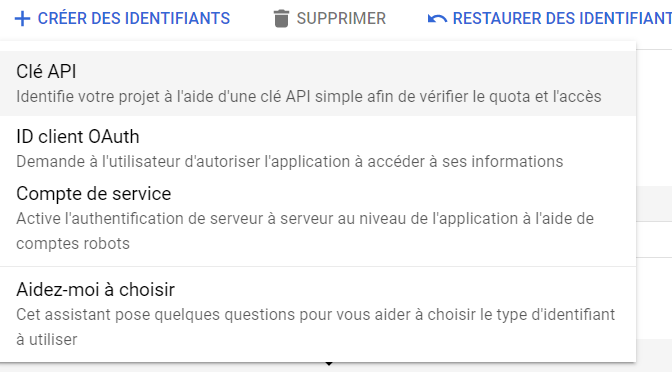




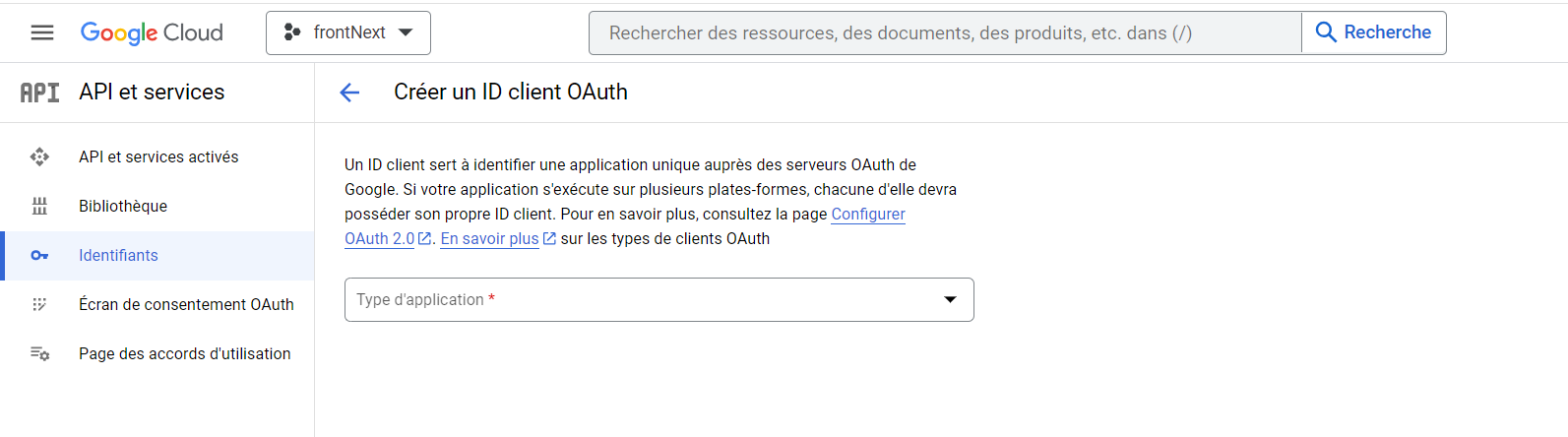




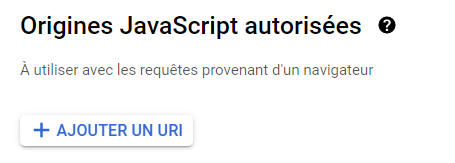


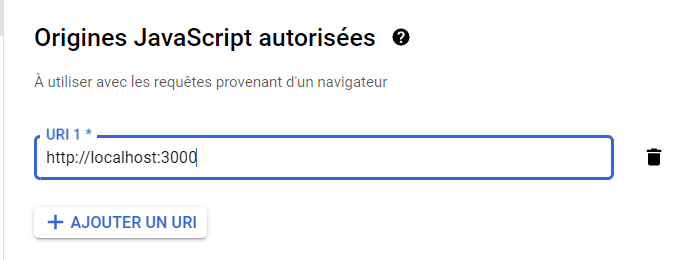






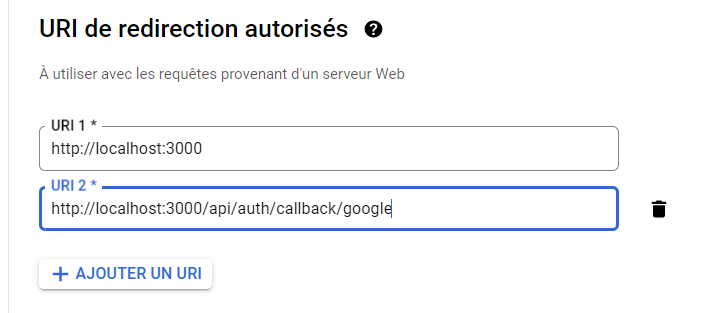
















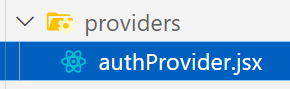
Veuillez noter :

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

Code secret du client : **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**

# Provider

Créer sous /src/ créer le dossier providers puis le fichier authProvider.jsx





'use client'

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

const AuthProvider = ({ children }) => {

  return <SessionProvider>{children}</SessionProvider>

}

export default AuthProvider

Puis lui faire appel dans le layout principal :



import { Inter } from "next/font/google";

import "./globals.css";

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

import Menu from '@/components/client/menu';

import connectDB from '@/lib/connectDB';

import AuthProvider from '@/providers/authProvider';

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

export const metadata = {

  title: "Create Next App",

  description: "Generated by create next app",

};

export default function RootLayout({ children }) {

await connectDB();

  return (

    <html lang="en">

      <body className={inter.className}>

      <AuthProvider>

      <Menu/>

        {children}

        </AuthProvider>

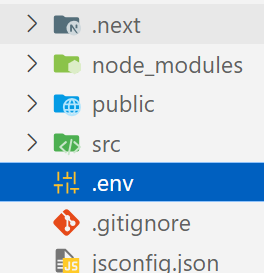
      </body>

    </html>

  );

}

# .env



Dans le fichier .env créer les variables d’environnement suivantes. Veuillez remplir les valeurs par vos credentials.

NEXT\_PUBLIC\_GOOGLE\_CLIENTID = 31017781977-c2mev6djp0jd9dte3hcqal7v8ha2tp27.apps.googleusercontent.com

NEXT\_PUBLIC\_GOOGLE\_CLIENTSECRET = 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

NEXT\_PUBLIC\_AUTH\_SECRET = secret

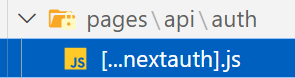
# /src/pages/api/auth/[…nextauth].js

Sous /src/ créer le dossier pages

Pus sous /src/pages le dossier api

Puis sous /src/pages/api le dossier auth

puis sous /src/pages/api/auth le fichier […nextauth].js



import NextAuth from "next-auth";

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

export const authOptions = {

    providers: [

        GoogleProvider({

            clientId: process.env.NEXT\_PUBLIC\_GOOGLE\_CLIENTID,

            clientSecret: process.env.NEXT\_PUBLIC\_GOOGLE\_CLIENTSECRET

        })

    ],

**secret: process.env.NEXT\_PUBLIC\_AUTH\_SECRET,**

}

export default NextAuth(authOptions)

Remarques :

1/ **secret** est indispensable pour récupérer la session par la suite.

2/ Pour des versions antérieures cette page devient :

***src/app/api/auth/[…nextauth]/route.js***

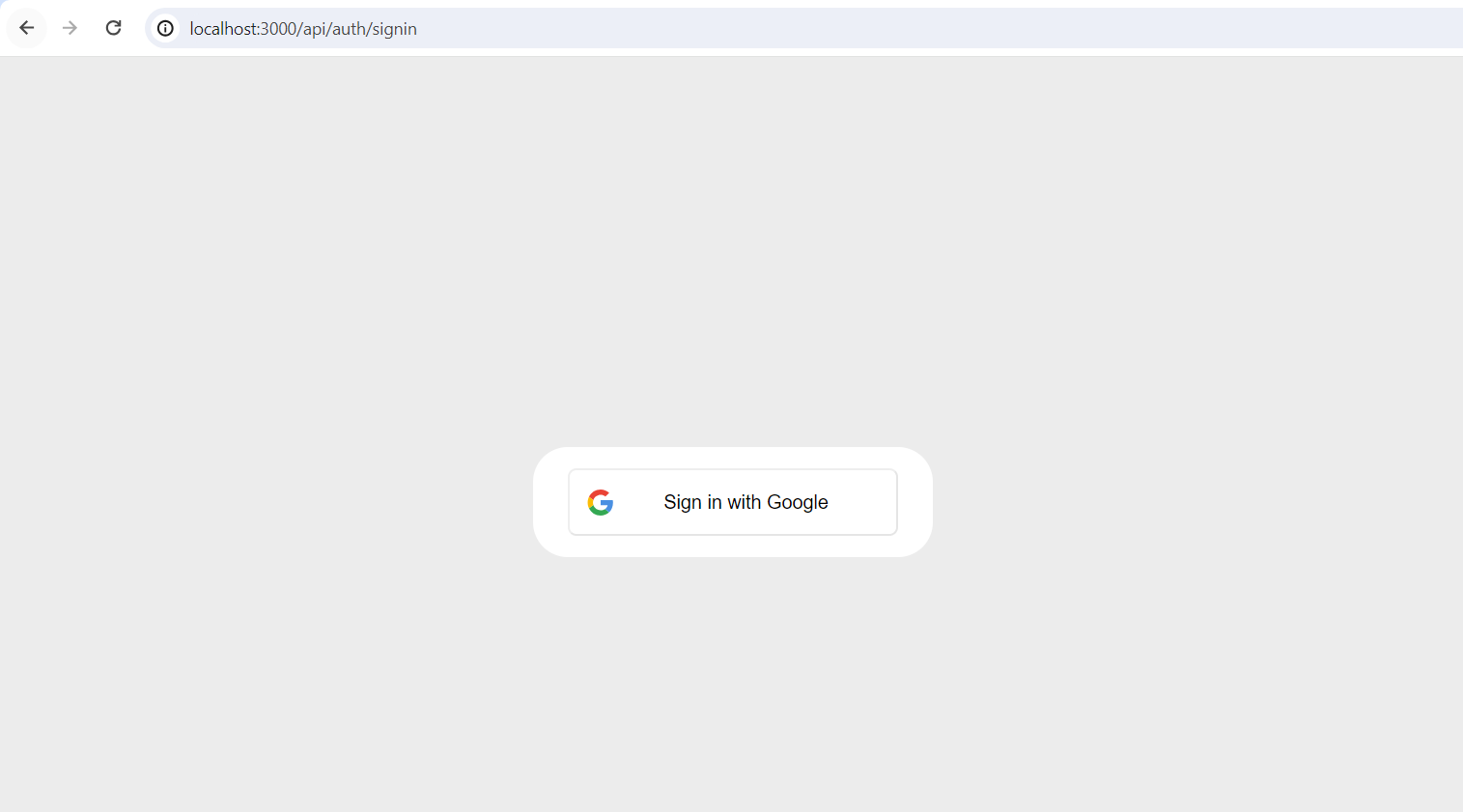
Et l’export devient :

const handler = NextAuth(authOptions)

export { handler as GET, handler as POST }

Dans le navigateur exécuter l’url :

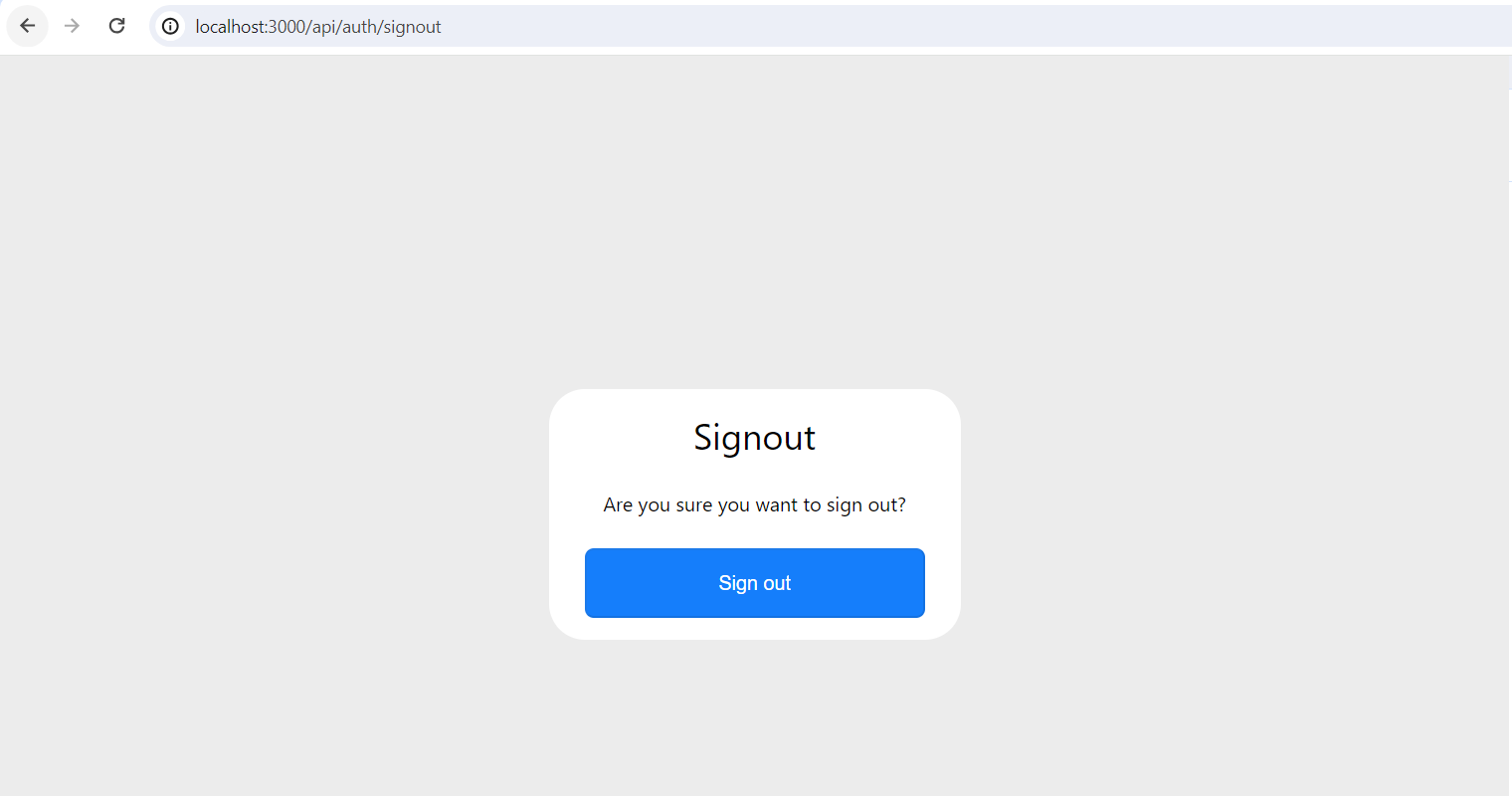
# <http://localhost:3000/api/auth/signin>



Si tout va bien on va se rediriger vers la page d’accueil

Pour se déconnecter :

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



# Le menu

On va ajouter le test de la session et afficher le menu correspondant selon qu’on est connecté ou pas.



'use client';

import Container from 'react-bootstrap/Container';

import Nav from 'react-bootstrap/Nav';

import Navbar from 'react-bootstrap/Navbar';

import Link from 'next/link';

import HomeIcon from '@mui/icons-material/Home';

import AccountCircleIcon from '@mui/icons-material/AccountCircle';

import HelpIcon from '@mui/icons-material/Help';

import LogoutIcon from '@mui/icons-material/Logout';

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

function Menu() {

  const { data: session } = useSession()

  return (

    <>

      <Navbar bg="dark" data-bs-theme="dark">

        <Container>

          <Navbar.Brand href="#home">Books Universe</Navbar.Brand>

          <Nav className="me-auto">

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

  {session? <Nav.Link onClick={() => signOut()}><LogoutIcon/>Se déconnecter</Nav.Link>

            : <Nav.Link onClick={() => signIn()}><AccountCircleIcon/>Se connecter</Nav.Link>

            }

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

          </Nav>

        </Container>

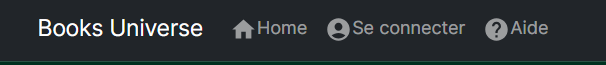
      </Navbar>

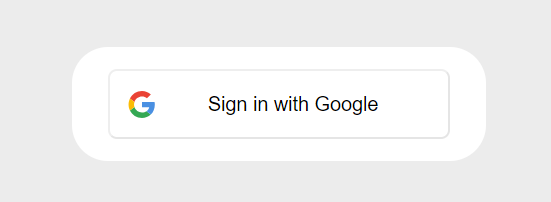
     </>

  );

}

export default Menu;





Après connexion



# Contrôle d’accès



import SideBar from "@/components/adminComponents/sideBar";

import { getServerSession } from 'next-auth'

import { authOptions } from '@/pages/api/auth/[...nextauth]'

export default **async** function AdminLayout({ children }) {

  const session = await getServerSession(authOptions)

  if (!session) {

    return (

      <section className='py-24'>

        <div className='container'>

          <h1 className='text-2xl font-bold'>

            You are not authorized to view this page

          </h1>

        </div>

      </section>

    )

  }

  return (

    <div className="row mt-4">

    <div className="col-md-3">

    <SideBar />

    </div>

    <div className="col-md-9">

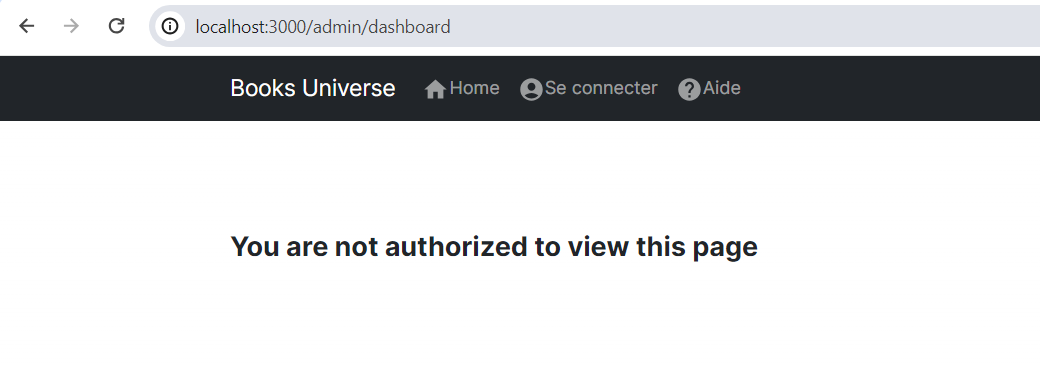
    {children}

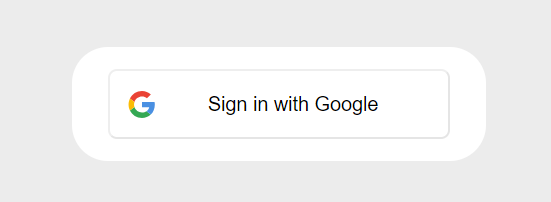
    </div>

    </div>

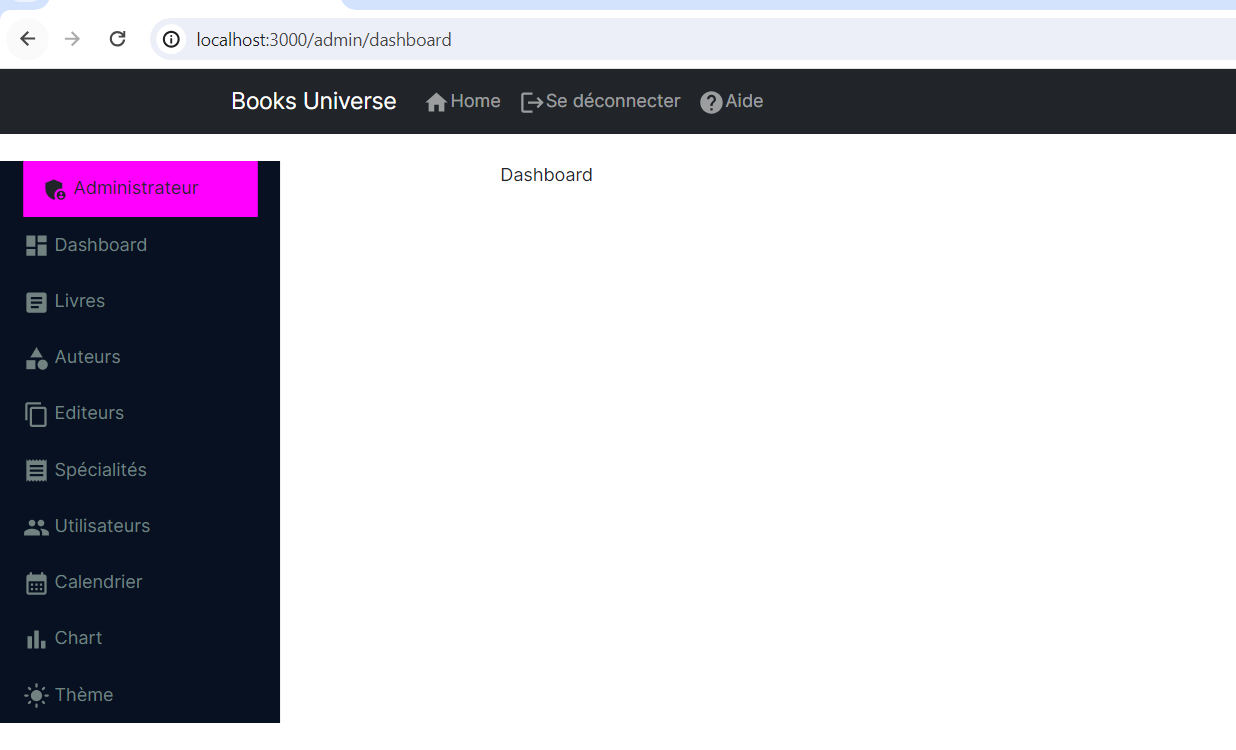
  )

}





Après connexion



# L’admin sideBar

On va ajouter le test de la session et afficher le menu admin correspondant selon celui qui s’est connecté



'use client';

import { Sidebar, Menu, MenuItem } from 'react-pro-sidebar';

import Link from 'next/link';

import AdminPanelSettingsIcon from '@mui/icons-material/AdminPanelSettings';

import DashboardIcon from '@mui/icons-material/Dashboard';

import ArticleIcon from '@mui/icons-material/Article';

import CategoryIcon from '@mui/icons-material/Category';

import ContentCopyIcon from '@mui/icons-material/ContentCopy';

import GroupIcon from '@mui/icons-material/Group';

import ReceiptIcon from '@mui/icons-material/Receipt';

import CalendarMonthIcon from '@mui/icons-material/CalendarMonth';

import BarChartIcon from '@mui/icons-material/BarChart';

import LightModeIcon from '@mui/icons-material/LightMode';

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

const SideBar = () => {

  const { data: session } = useSession()

  return (

    <Sidebar>

        <Menu

            menuItemStyles={{

            button: ({ level, active, disabled }) => {

                // only apply styles on first level elements of the tree

                if (level === 0)

                return {

                    color: disabled ? '#b4c2c0' : '#718280',

                    backgroundColor: active ? '#4f5d73' : '#071121',

                };

            },

            }}

        >

      <MenuItem>

<div className="card" style={{"backgroundColor": "#FF00FF"}}>

        <div className="card-body">

          <AdminPanelSettingsIcon /> Administrateur

        </div>

      </div>

      </MenuItem>

      <MenuItem>

     <div className="card" style={{"backgroundColor": "#44ebe2"}}>

      <div className="card-body">

          {session?.user?.image ? (

              <div>

                <img

                  src={session.user.image}

                  alt={session.user.name}

                  className='inline-block rounded-full'

                  width="30"

                  height="30"

               />

                {session.user.name}

              </div>

            ) : (

             null

            )}

        </div>

        </div>

      </MenuItem>

      <MenuItem component={<Link href="/admin/dashboard" />}> <DashboardIcon/> Dashboard </MenuItem>

      <MenuItem component={<Link href="/admin/livres" />}> <ArticleIcon/> Livres </MenuItem>

      <MenuItem component={<Link href="/admin/auteurs" />}> <CategoryIcon/> Auteurs </MenuItem>

      <MenuItem component={<Link href="/admin/editeurs" />}> <ContentCopyIcon/> Editeurs</MenuItem>

      <MenuItem component={<Link href="#" />}> <ReceiptIcon/> Spécialités </MenuItem>

      <MenuItem component={<Link href="#" />}> <GroupIcon/> Utilisateurs </MenuItem>

      <MenuItem component={<Link href="#" />}> <CalendarMonthIcon/> Calendrier </MenuItem>

      <MenuItem component={<Link href="#" />}> <BarChartIcon/> Chart </MenuItem>

      <MenuItem component={<Link href="#" />}> <LightModeIcon/> Thème </MenuItem>

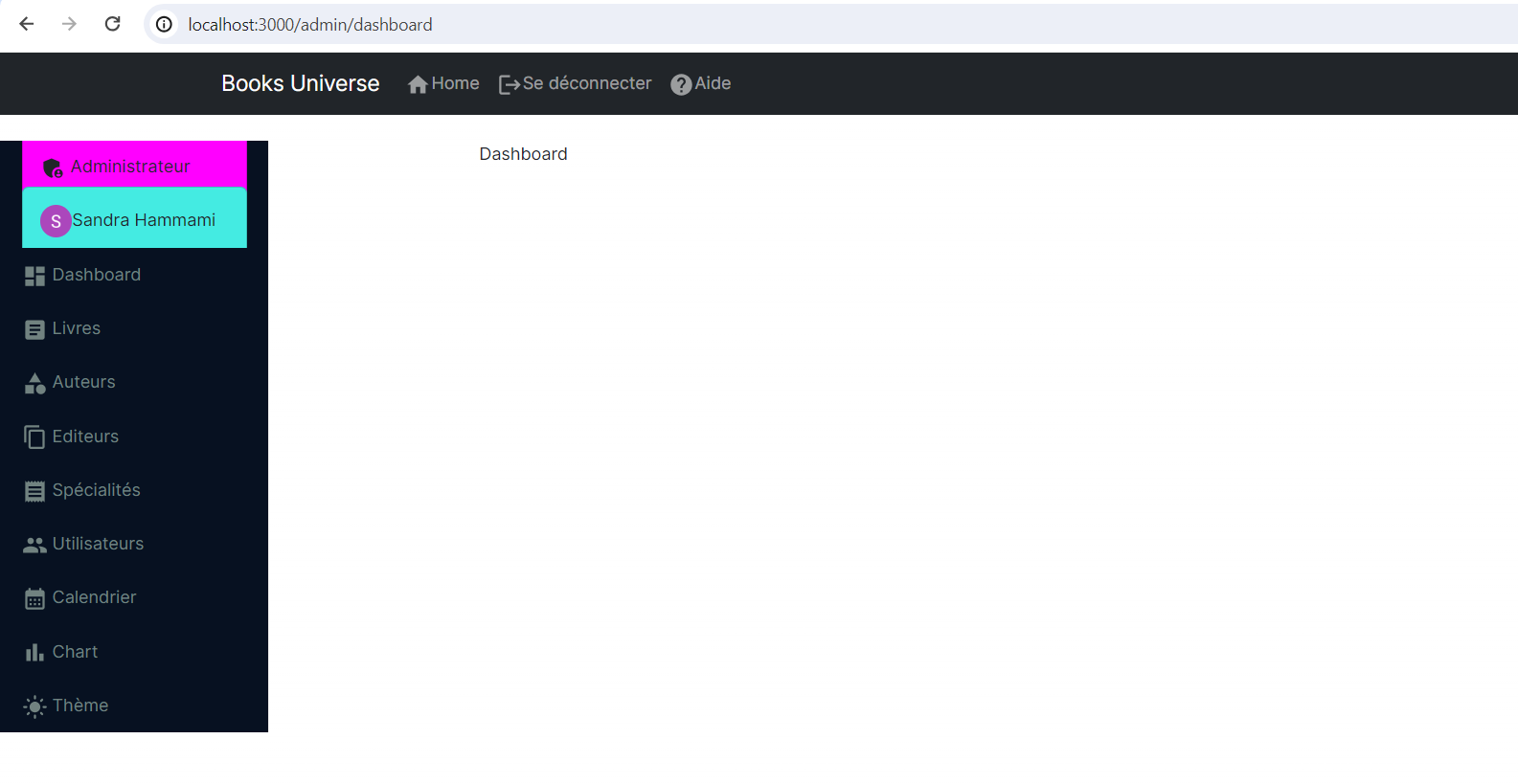
    </Menu>

  </Sidebar>

  )

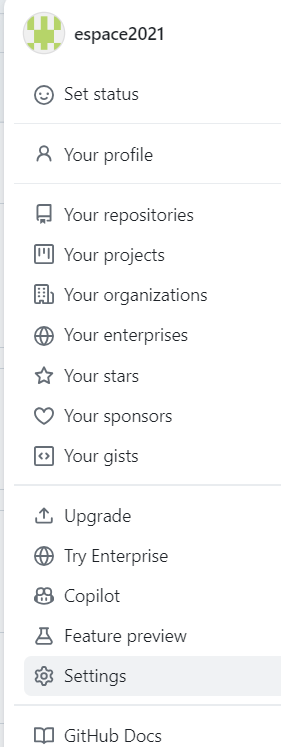
}

export default SideBar



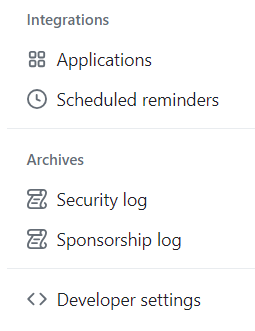
# GitHub Credentials

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

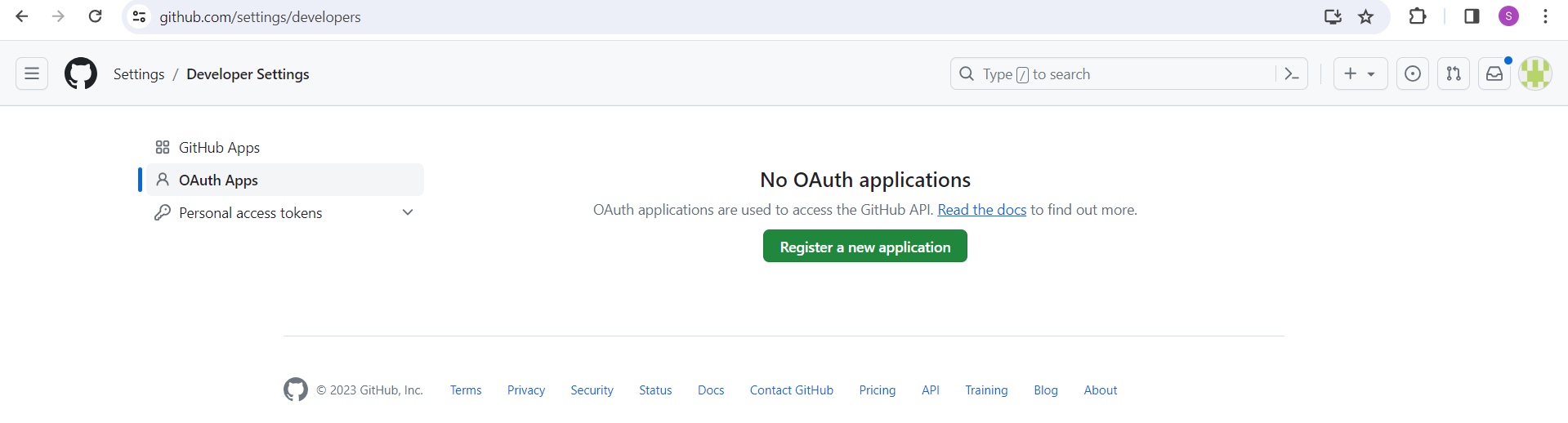


Sélectionnez Settings dans le menu déroulant :

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 :

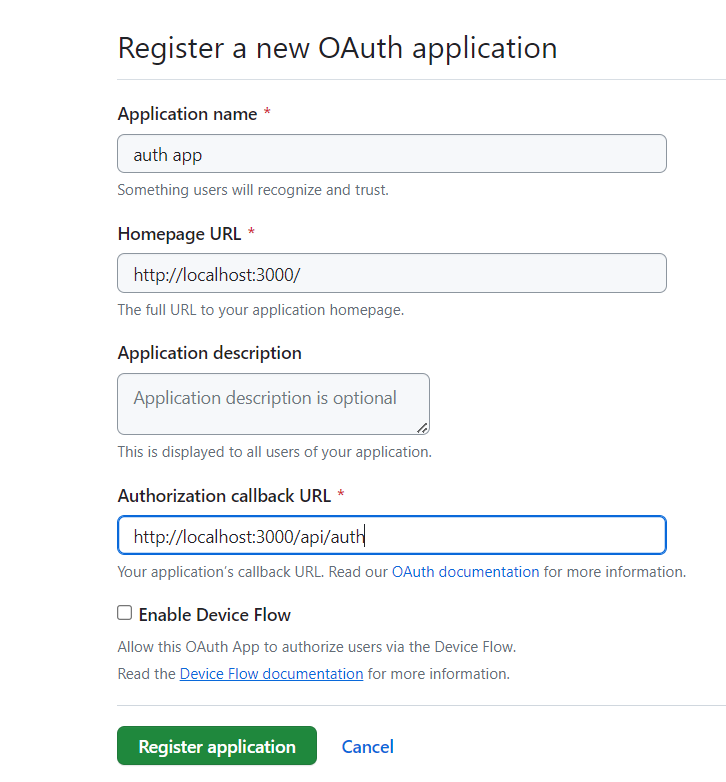


Remplissez le formulaire que vous voyez sur la page qui suit :

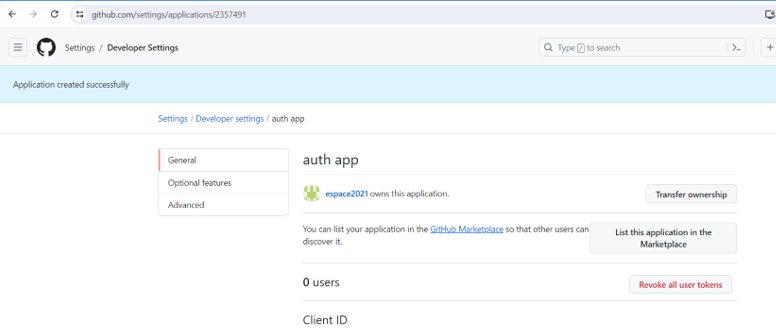
Application name - auth app

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 :

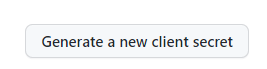


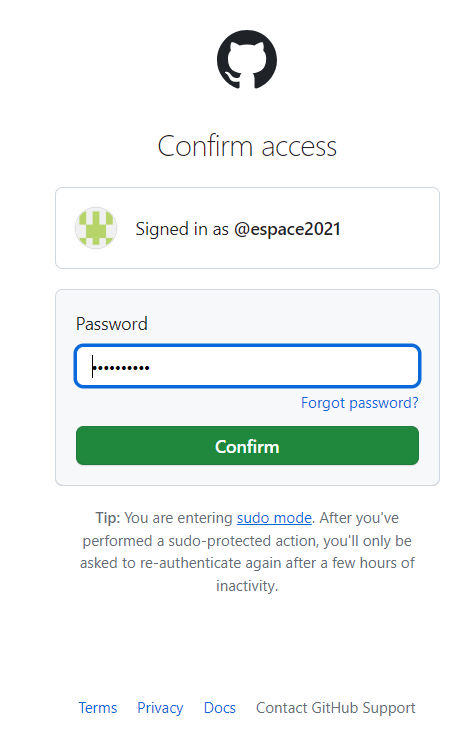
Veuillez noter

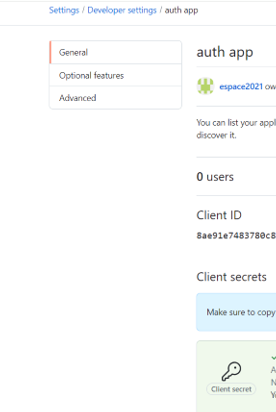
## Client ID

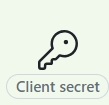
8ae91e7483780c86e578

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









Veuillez noter Client secret

a10f093c2d1eb2253b7e3149e8f42b11dafe9fc0

# Ajouter le provider GitHub



NEXT\_PUBLIC\_GITHUB\_CLIENTID = 8ae91e7483780c86e578

NEXT\_PUBLIC\_GITHUB\_CLIENTSECRET = a10f093c2d1eb2253b7e3149e8f42b11dafe9fc0



import NextAuth from "next-auth";

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

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

export const authOptions = {

    providers: [

        GoogleProvider({

            clientId: process.env.NEXT\_PUBLIC\_GOOGLE\_CLIENTID,

            clientSecret: process.env.NEXT\_PUBLIC\_GOOGLE\_CLIENTSECRET

        }),

        GitHubProvider({

            clientId: process.env.NEXT\_PUBLIC\_GITHUB\_CLIENTID,

            clientSecret: process.env.NEXT\_PUBLIC\_GITHUB\_CLIENTSECRET

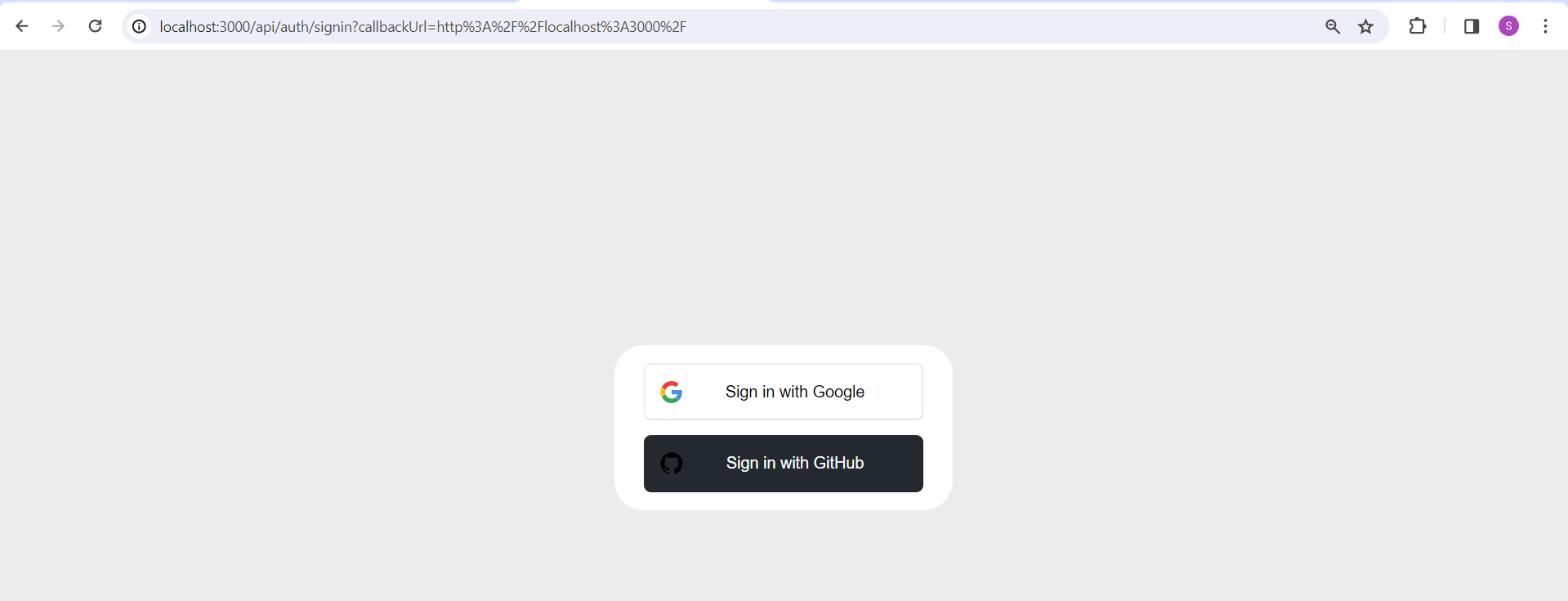
          })

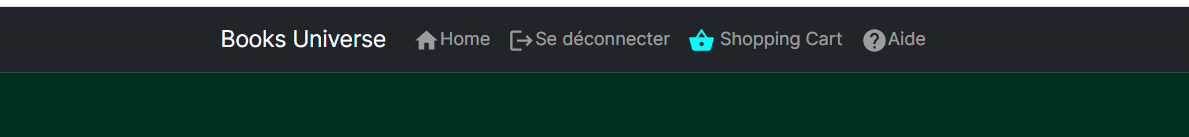
    ],

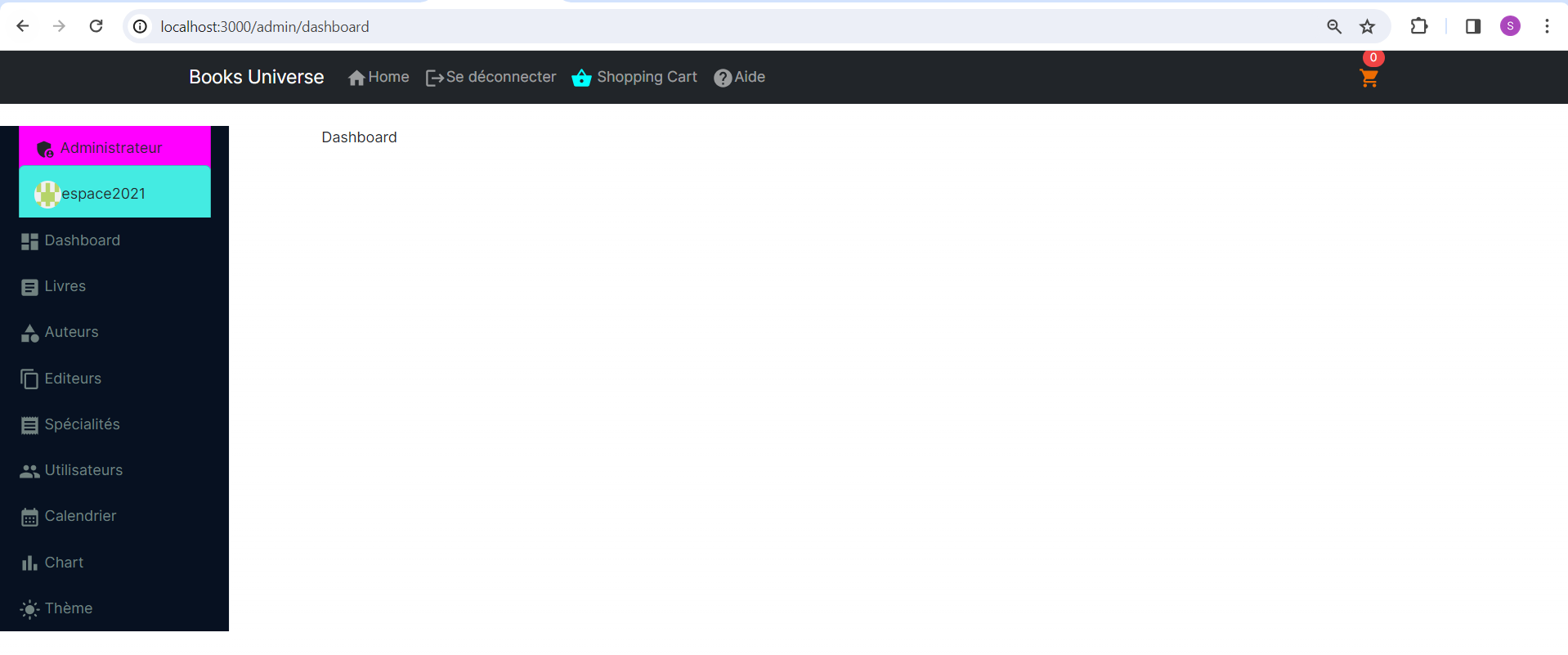
    secret: process.env.NEXT\_PUBLIC\_AUTH\_SECRET,

}

export default NextAuth(authOptions)

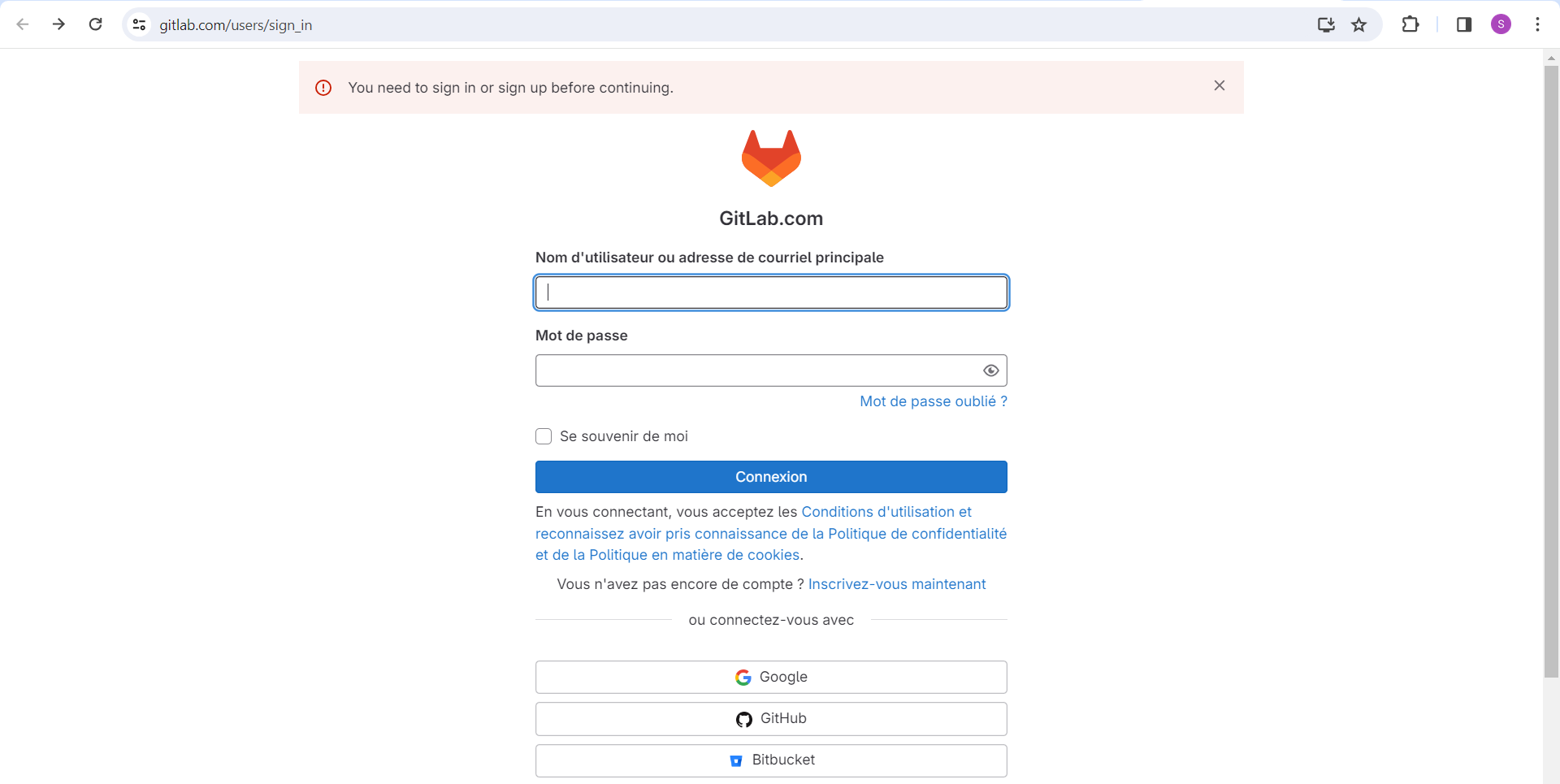


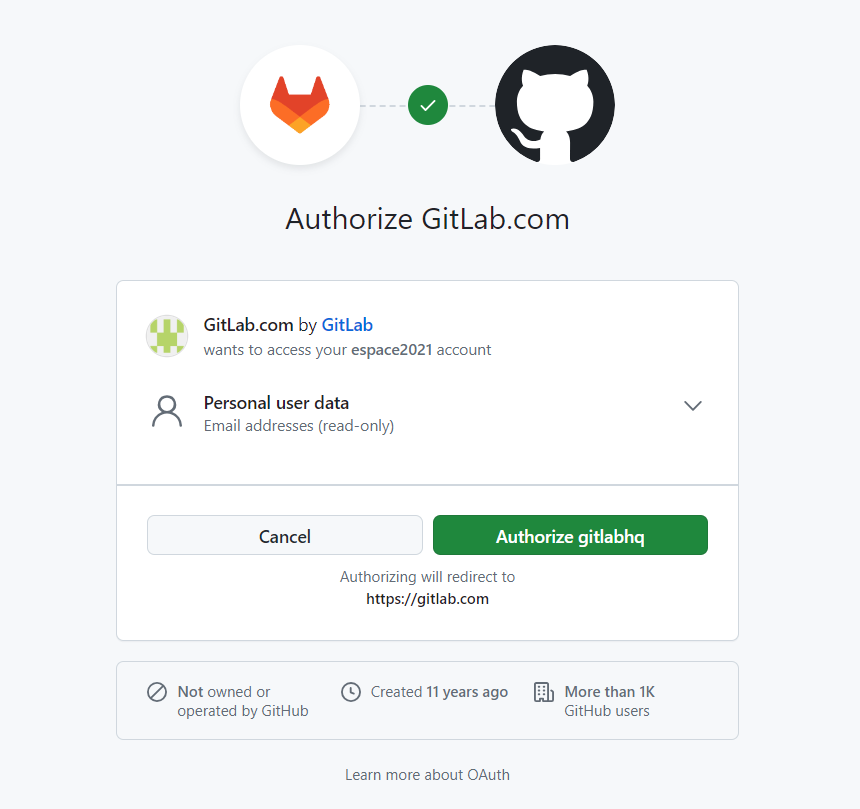




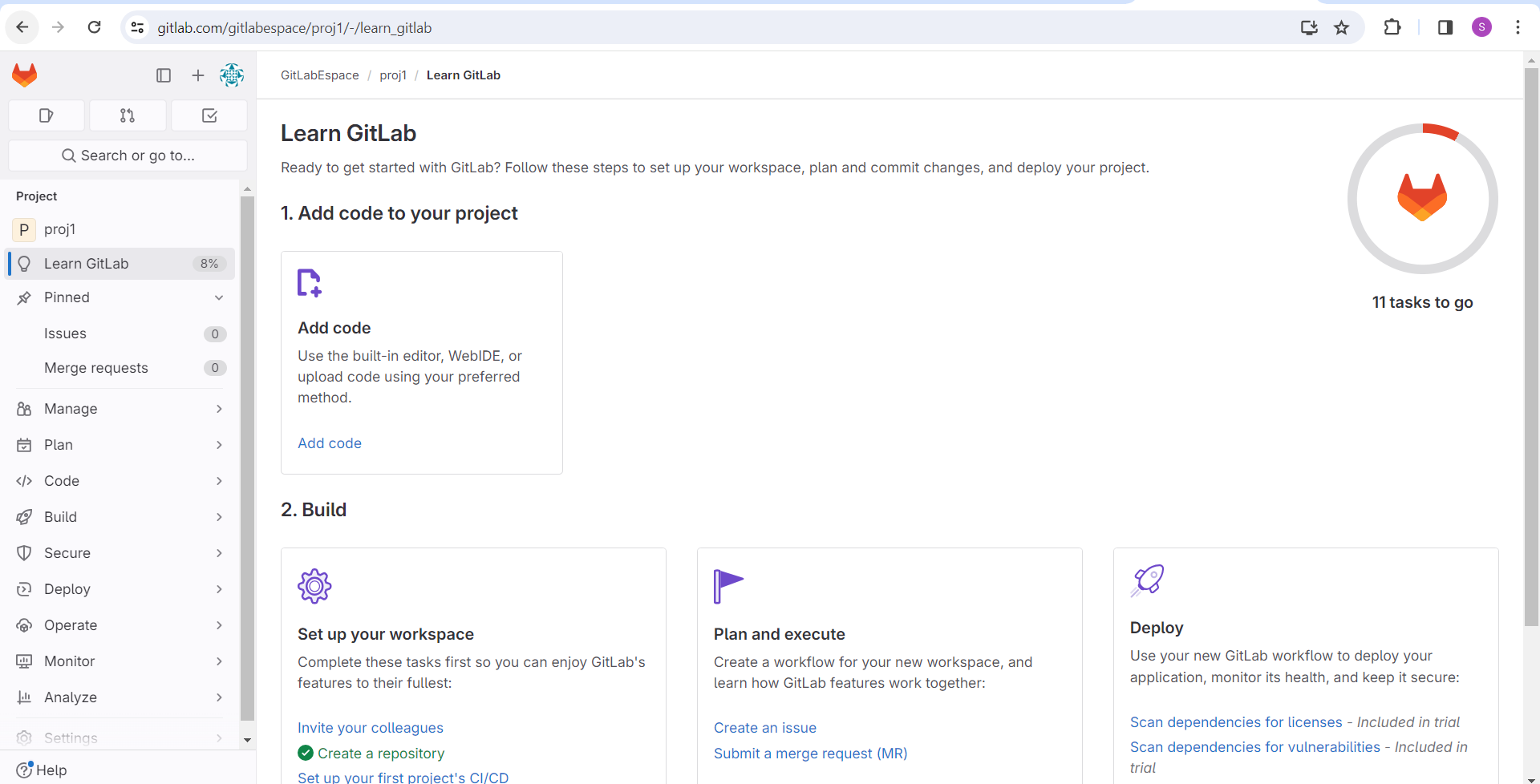
# GitLab Credentials

<https://gitlab.com/users/sign_in>

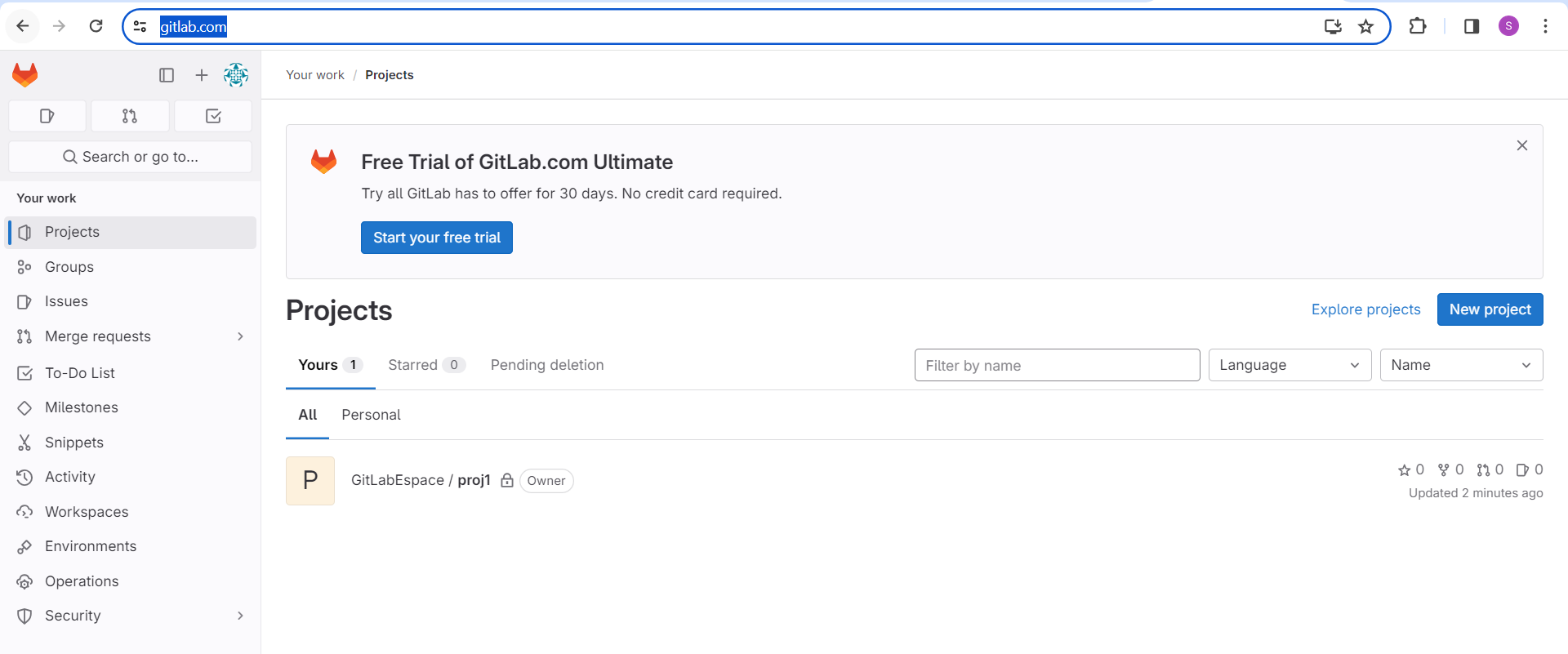


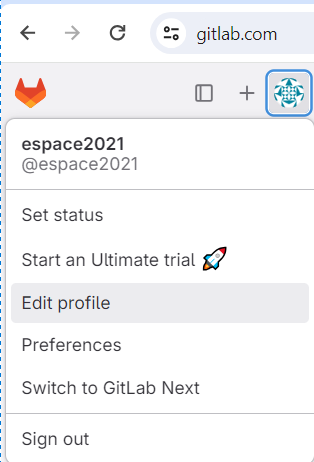


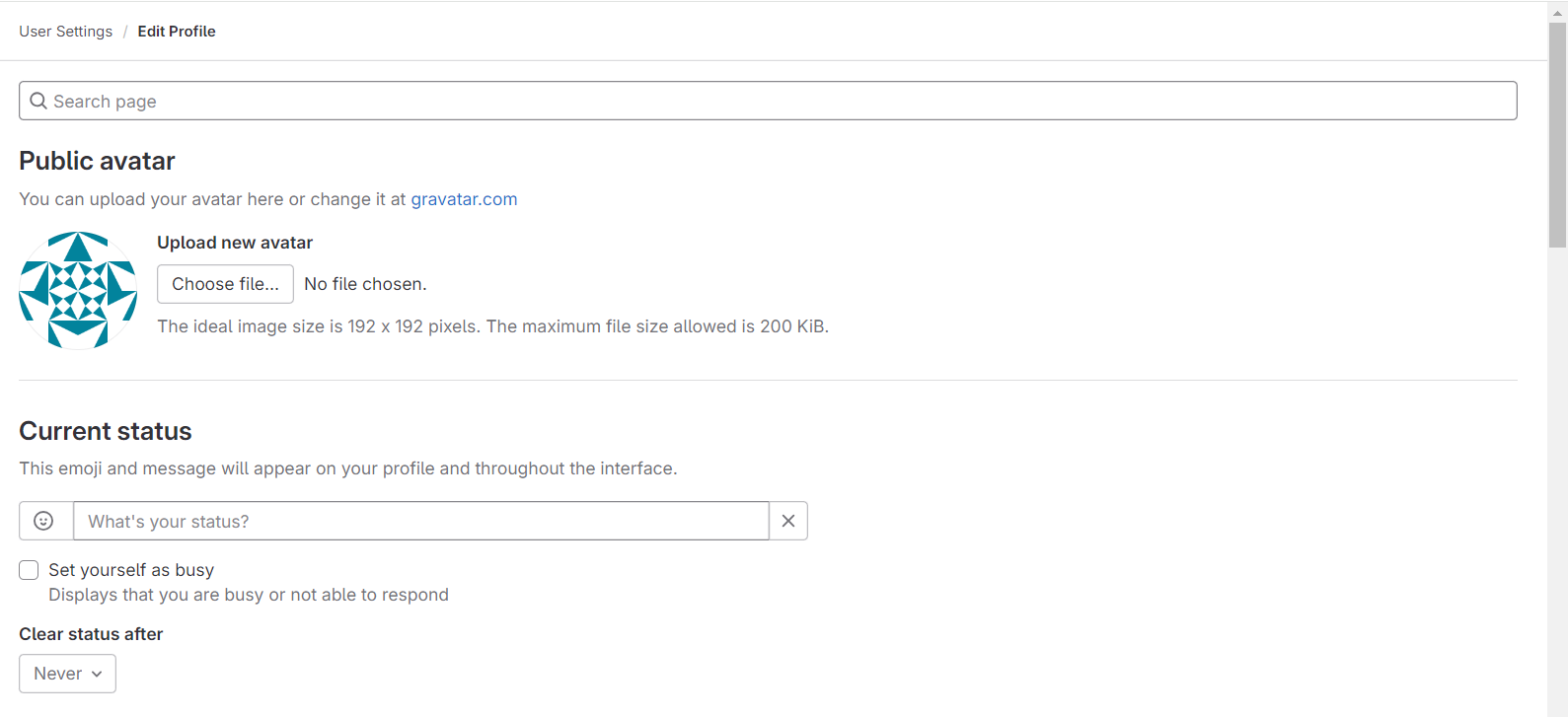


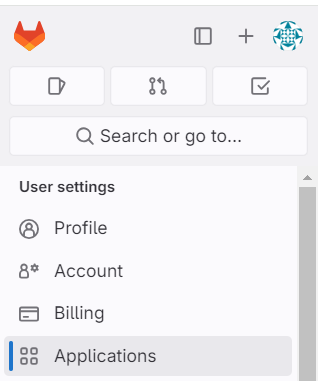


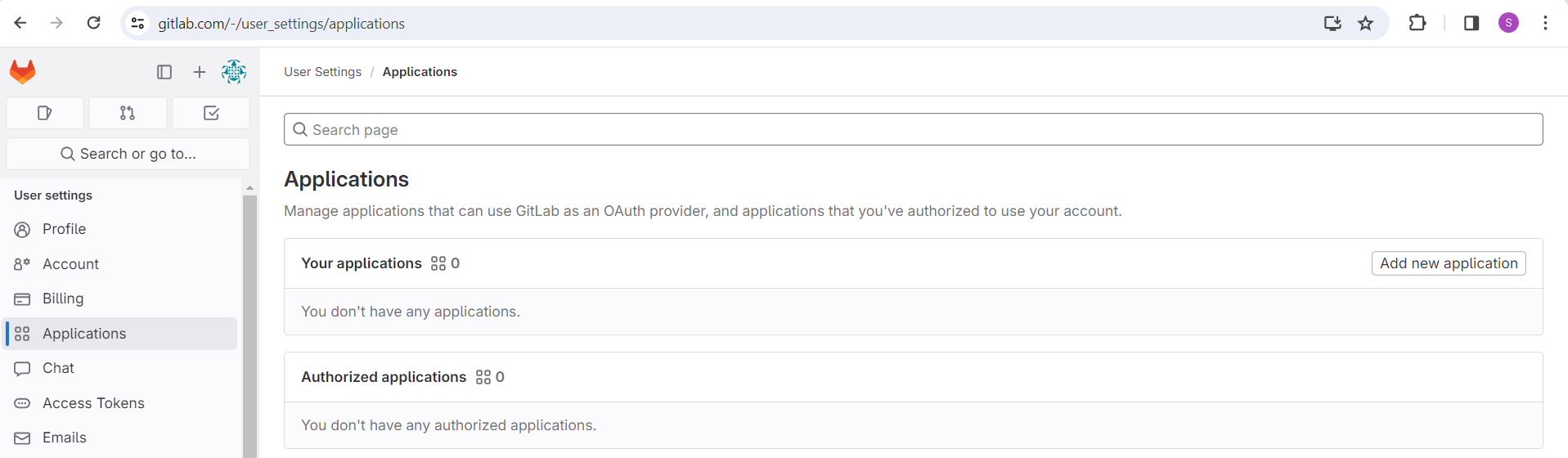
<https://gitlab.com/>



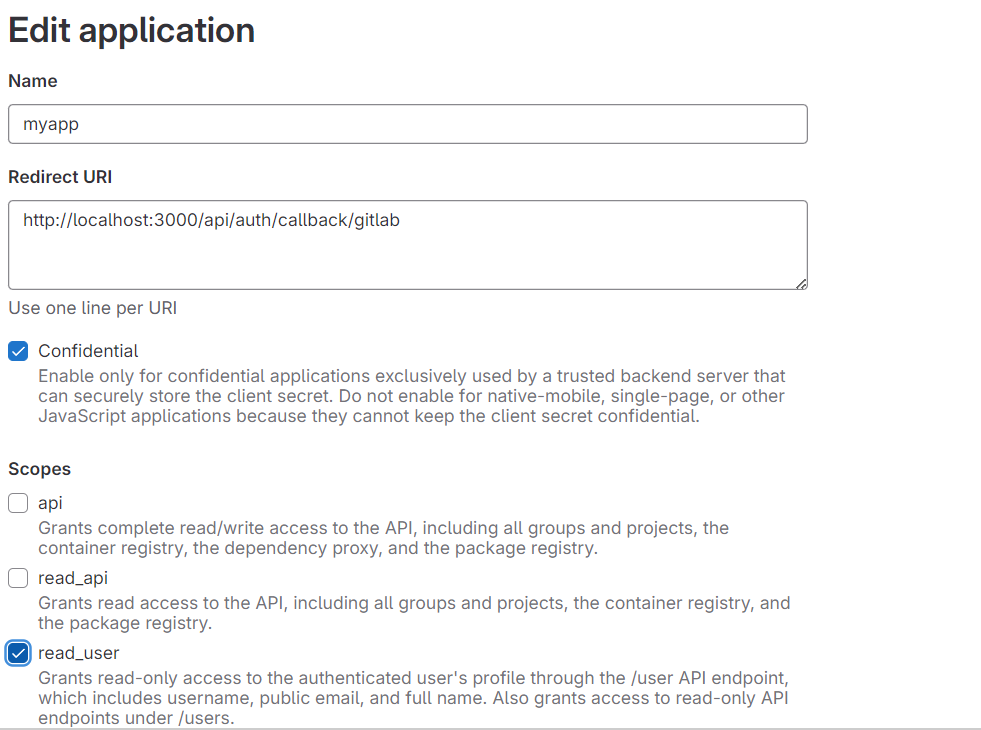


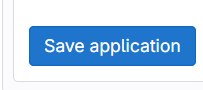


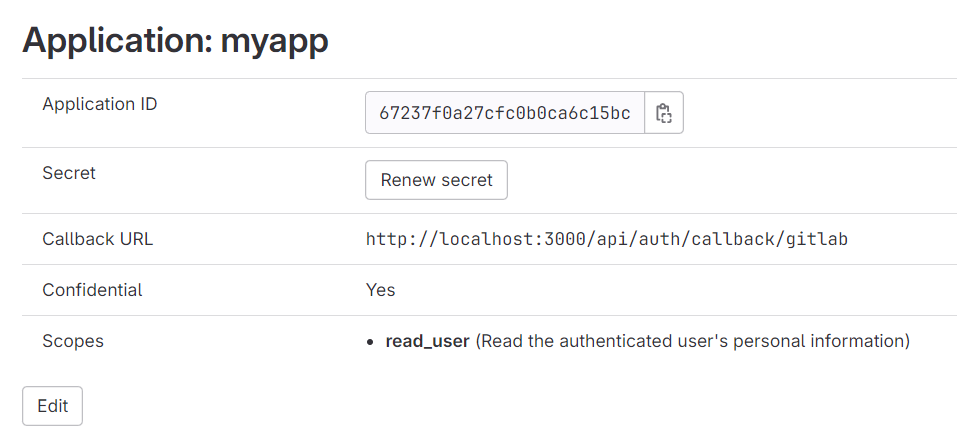








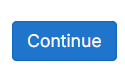


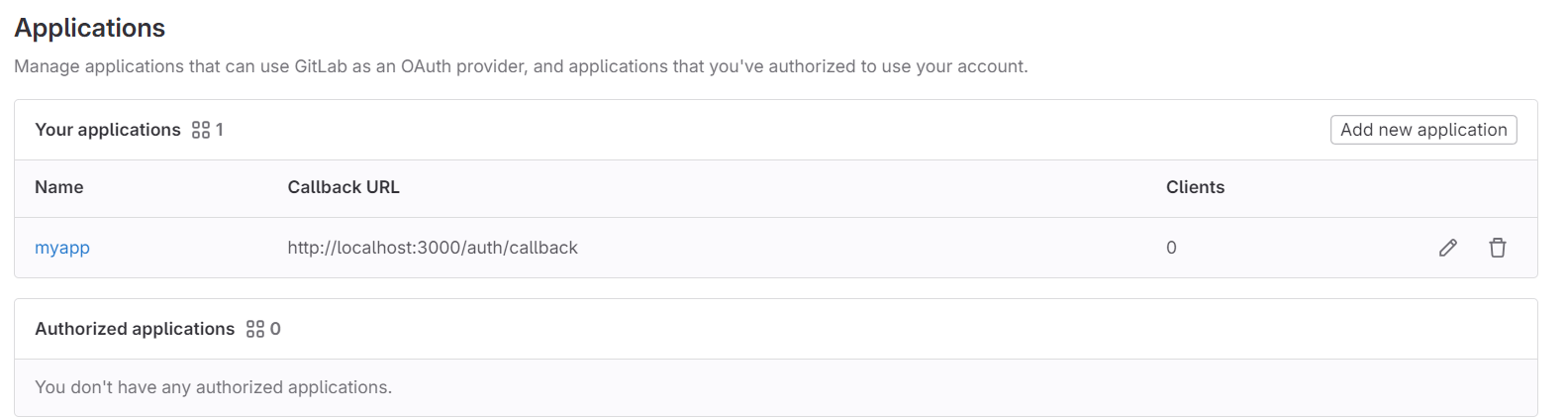


User ID (Application ID) : 67237f0a27cfc0b0ca6c15bc0cf033647c308b45f4c75d492ee8bfeaeb391786

Secret ID (Secret) : gloas-d5dbf07e5c6fb2f26495b34bc2910cff04b4b1e15b97bfa54b27829d393fe45d

http://localhost:3000/





# Ajoutet GitLab Provider



NEXT\_PUBLIC\_GITLAB\_CLIENT\_ID = 2142963467237f0a27cfc0b0ca6c15bc0cf033647c308b45f4c75d492ee8bfeaeb391786

NEXT\_PUBLIC\_GITLAB\_CLIENT\_SECRET = gloas-d5dbf07e5c6fb2f26495b34bc2910cff04b4b1e15b97bfa54b27829d393fe45d



import NextAuth from "next-auth";

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

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

import GitlabProvider from "next-auth/providers/gitlab";

export const authOptions = {

    providers: [

        GoogleProvider({

            clientId: process.env.NEXT\_PUBLIC\_GOOGLE\_CLIENTID,

            clientSecret: process.env.NEXT\_PUBLIC\_GOOGLE\_CLIENTSECRET

        }),

        GitHubProvider({

            clientId: process.env.NEXT\_PUBLIC\_GITHUB\_CLIENTID,

            clientSecret: process.env.NEXT\_PUBLIC\_GITHUB\_CLIENTSECRET

          }),

        GitlabProvider({

            clientId: process.env.NEXT\_PUBLIC\_GITLAB\_CLIENT\_ID,

            clientSecret: process.env.NEXT\_PUBLIC\_GITLAB\_CLIENT\_SECRET

          })

    ],

    secret: process.env.NEXT\_PUBLIC\_AUTH\_SECRET,

}

export default NextAuth(authOptions)

