

Stockage des images dans Firebase et affichage avec Filepond dans ajout et modification des articles

Firebase est le nom d'une plateforme mobile de Google qui facilite la création de back-end à la fois scalable et performant. En d'autres termes, il s'agit d'une plateforme qui permet de développer rapidement des applications pour mobile et pour le web.

L'objectif de la création de Firebase.google.com en 2011 par James Tamplin et Andrew Lee est d'éviter aux professionnels et aux particuliers de s'engager dans un processus complexe de création et de maintenance d'une architecture serveur.

De plus, la plateforme peut être exploitée par plusieurs utilisateurs en même temps sans connaître un quelconque bug. La praticité est également au rendez-vous grâce à ses fonctionnalités intuitives. Depuis le rachat de la plateforme par Google en 2014, Firebase sdks a connu de nombreuses améliorations et n'a de cesse de satisfaire ses utilisateurs.

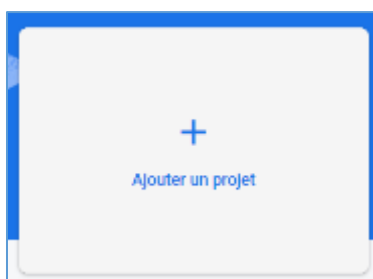
Dans Firebase, vous trouverez des API intuitives regroupées dans un SDK unique. Ces API, en plus de vous faire gagner du temps, vous permettent de réduire le nombre d'intégrations que vous devez gérer par le biais de votre application.

Configuration d'un projet Firebase

Accéder au site :

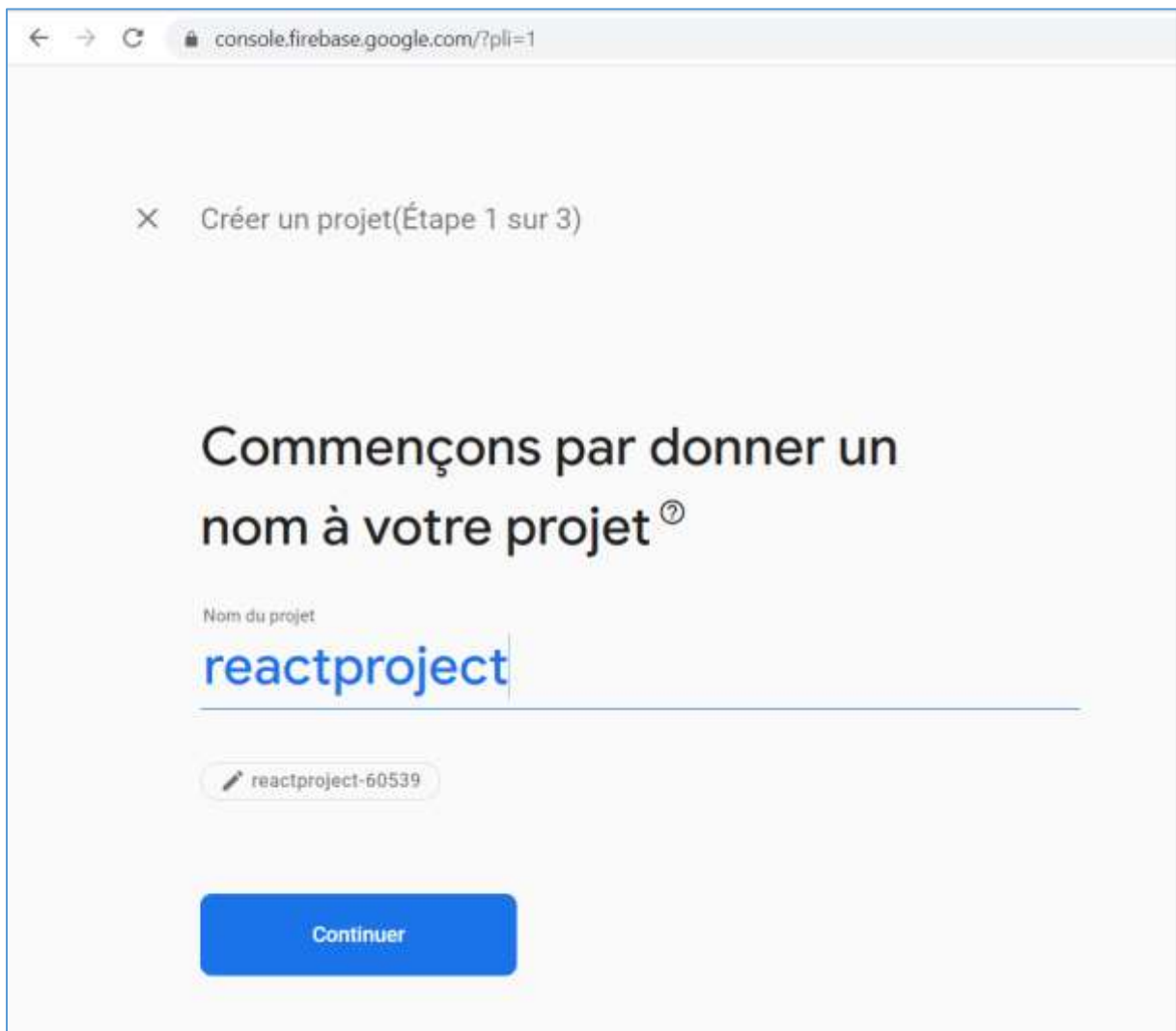
<https://console.firebase.google.com/>

Cliquer sur





Donner un nom pour le projet puis continuer :



Décocher Google Analytics puis valider

← → ↺

console.firebase.google.com/?pli=1

×

Créer un projet(Étape 2 sur 2)

Google Analytics est une solution d'analyse gratuite et illimitée qui permet le ciblage, la création de rapports et bien plus dans Firebase Crashlytics, Cloud Messaging, Remote Config, A/B Testing, Cloud Functions et la messagerie dans l'application.

Google Analytics permet d'exploiter les fonctionnalités suivantes :

×

Tests A/B ⓘ

×

Segmentation et ciblage des utilisateurs dans les produits Firebase ⓘ

×

Utilisateurs n'ayant pas subi de plantage ⓘ

×

Déclencheurs de fonctions Cloud basés sur des événements ⓘ

×

Rapports gratuits et illimités ⓘ

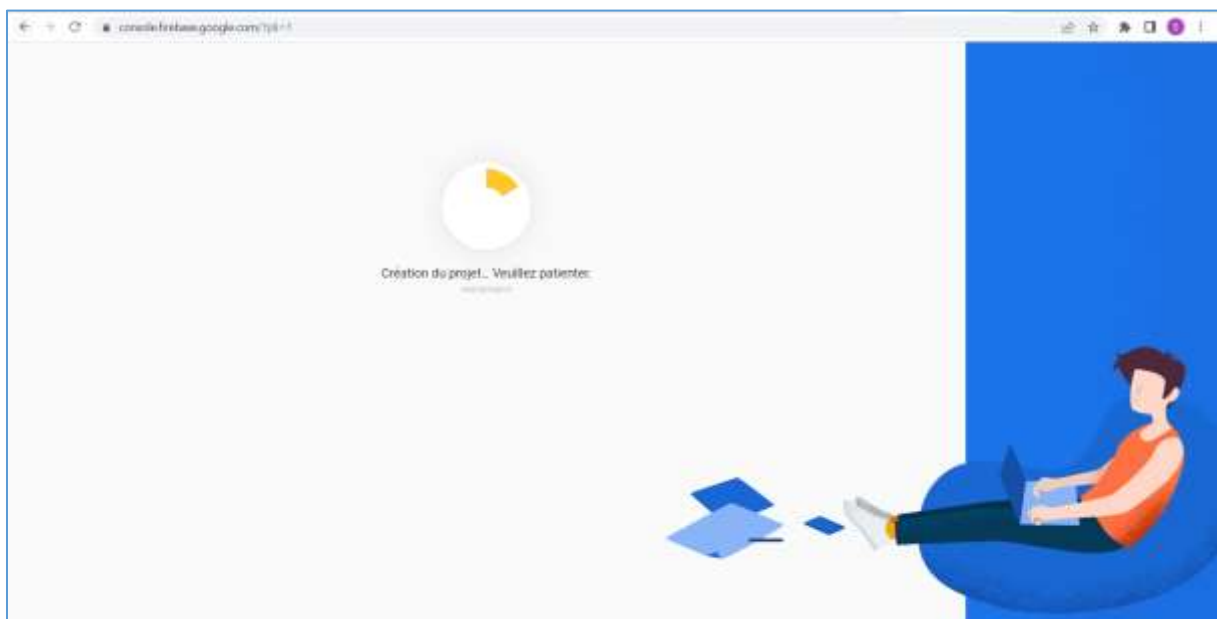
☐

Activer Google Analytics pour ce projet

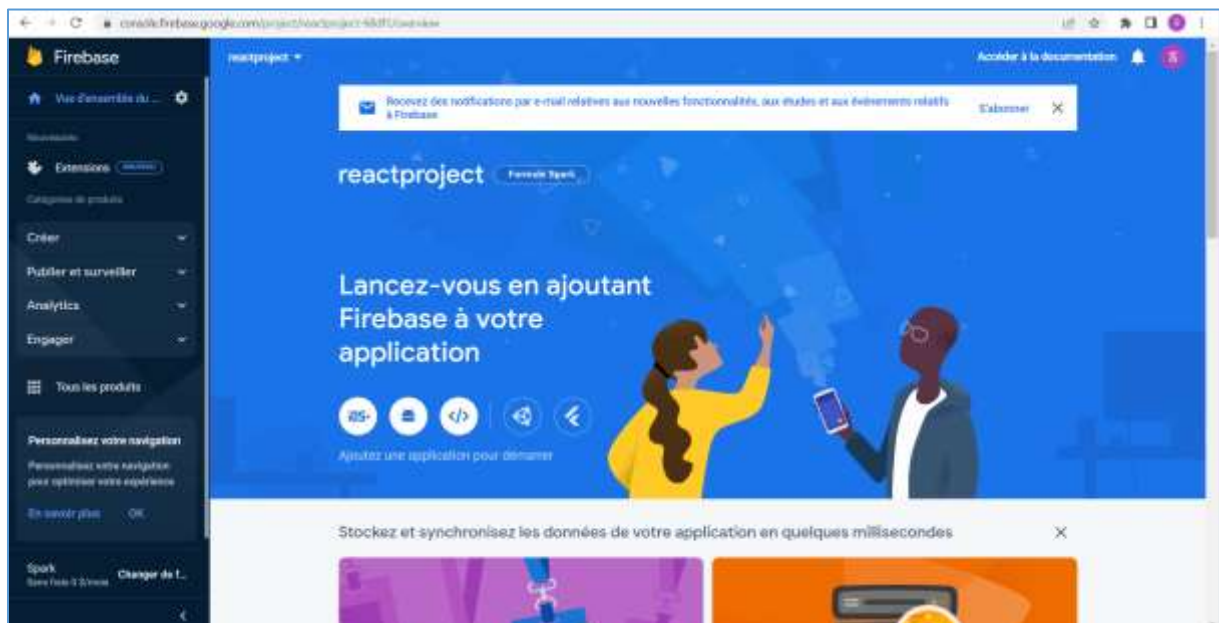
Recommandation

Précédent

Créer un projet



Cliquer sur continuer

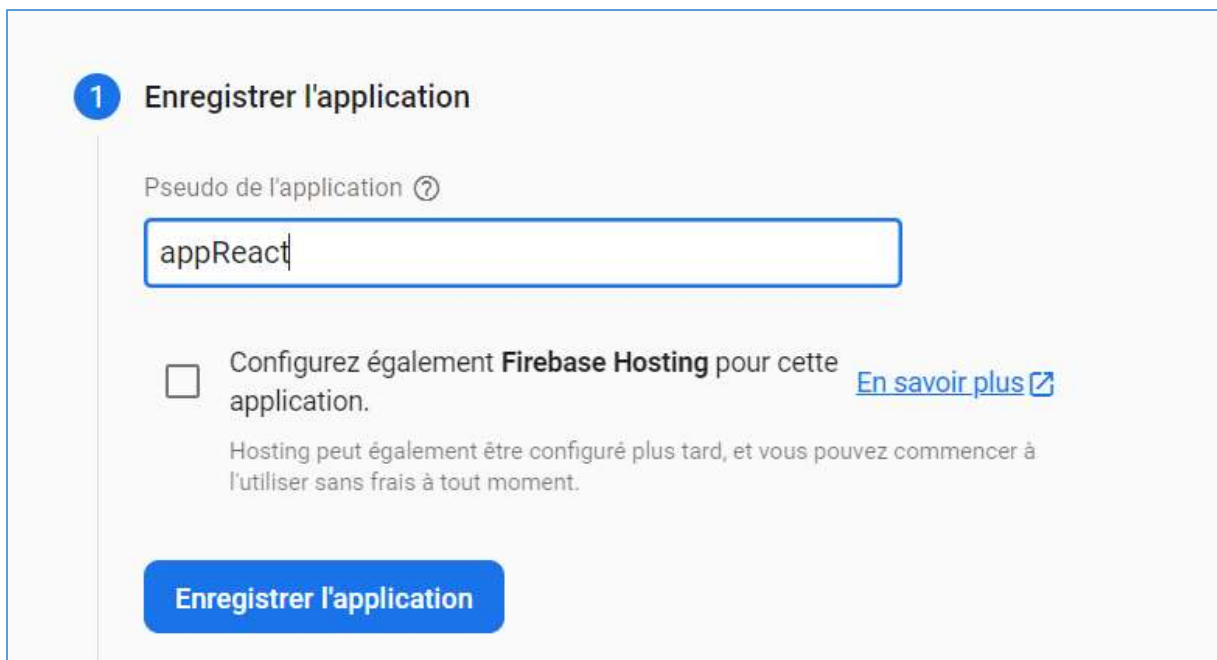
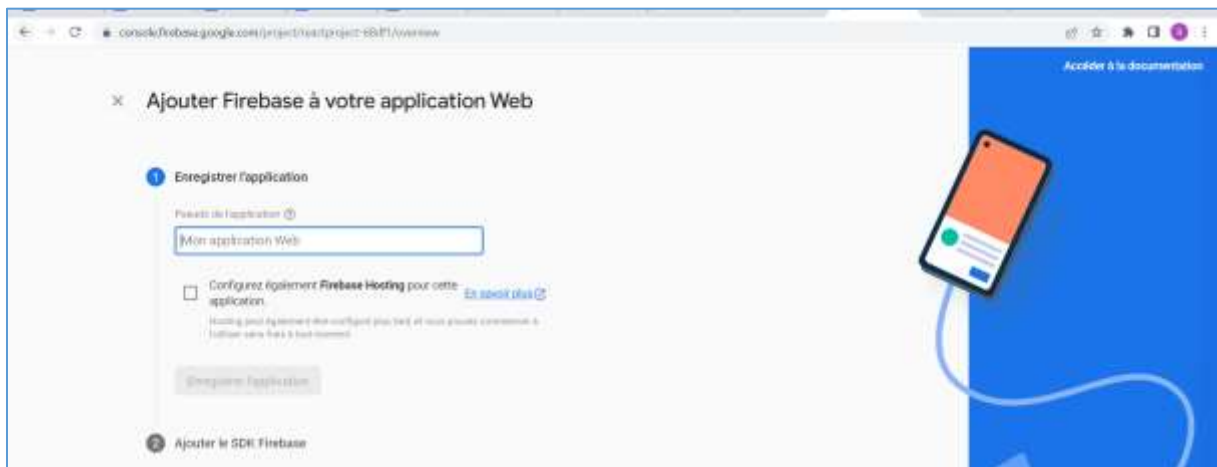


Cliquer sur l'icône Web




Donnez un surnom à votre application et cliquez sur S'inscrire.

Copiez l'objet de configuration fourni. Vous en aurez besoin pour connecter votre application à Firebase.



Copiez l'objet de configuration fourni. Vous en aurez besoin pour connecter votre application à Firebase.

2 Ajouter le SDK Firebase

☒ Utiliser npm  ☐ Utiliser une balise <script> 

Si vous utilisez déjà [npm](#) et un bundler de module comme [webpack](#) ou [Rollup](#), vous pouvez exécuter la commande suivante pour installer la dernière version du SDK :

```
$ npm install firebase
```



Ensuite, initialisez Firebase et commencez à exploiter les SDK pour les produits que vous souhaitez utiliser.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyBaxrbWZQSNFI230jsJHSeRB3RyLaJ1TRXY",
  authDomain: "reactproject-68df1.firebaseio.com",
  projectId: "reactproject-68df1",
  storageBucket: "reactproject-68df1.appspot.com",
  messagingSenderId: "373746469631",
  appId: "1:373746469631:web:416496270e3d4643a45aca"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

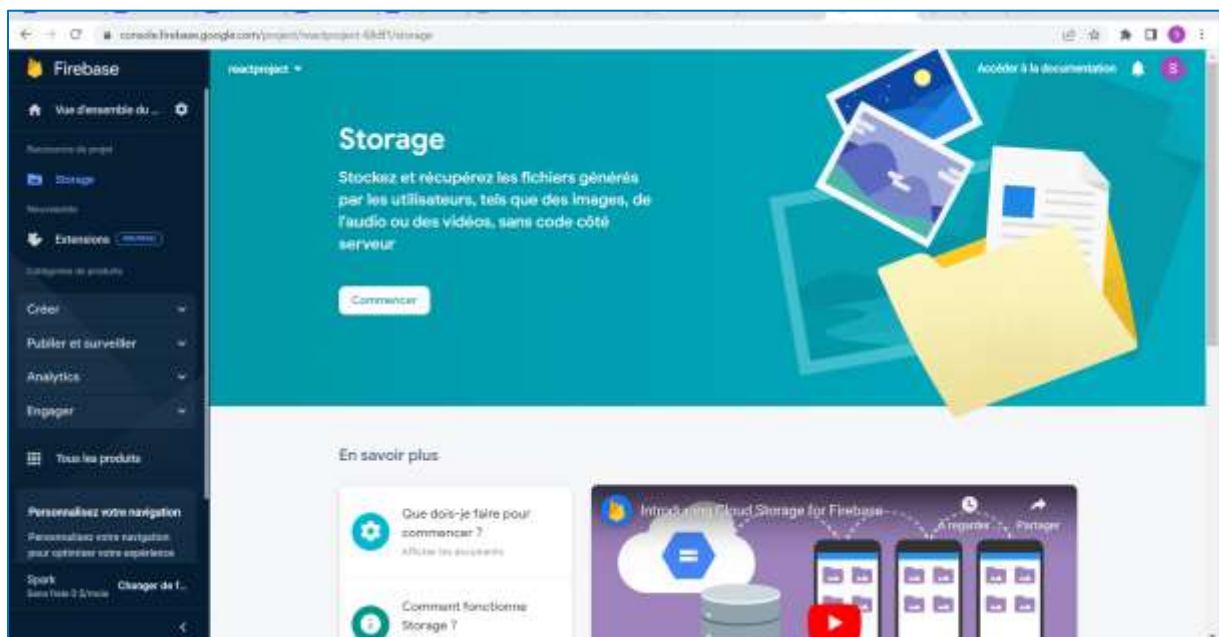
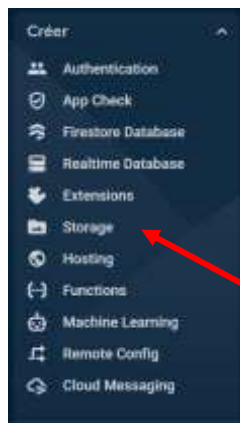


Cliquer sur

Accéder à la console

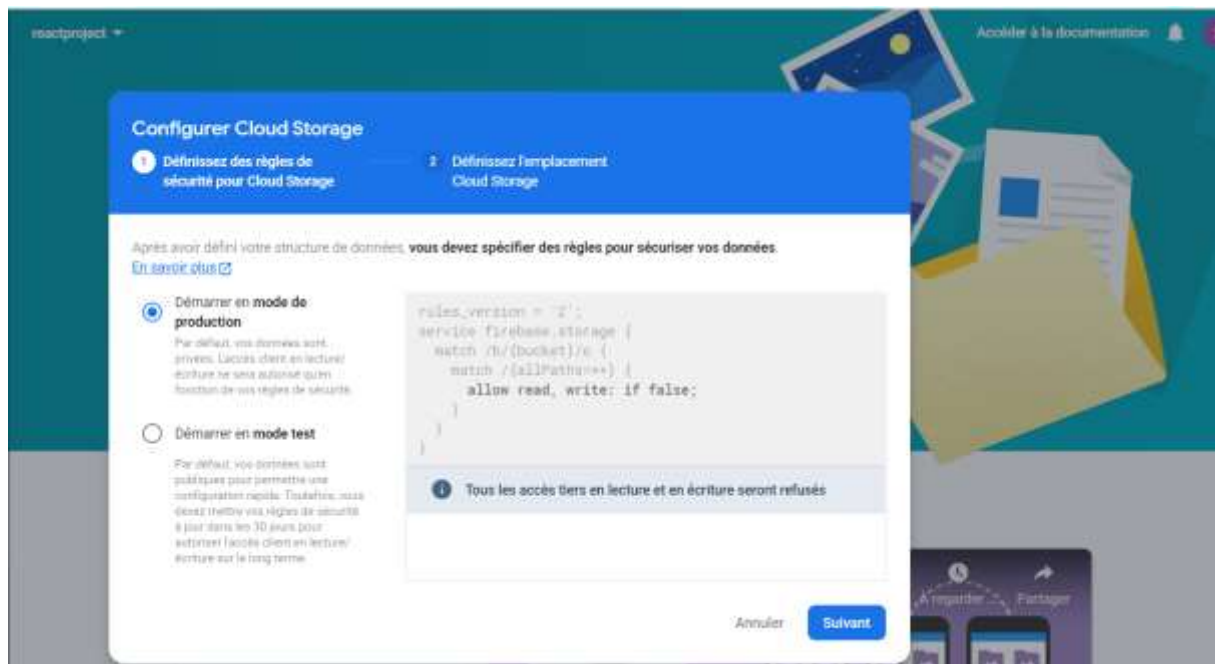
Firebase stocke les fichiers dans un bucket de stockage cloud. Suivez les étapes suivantes pour le créer :

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

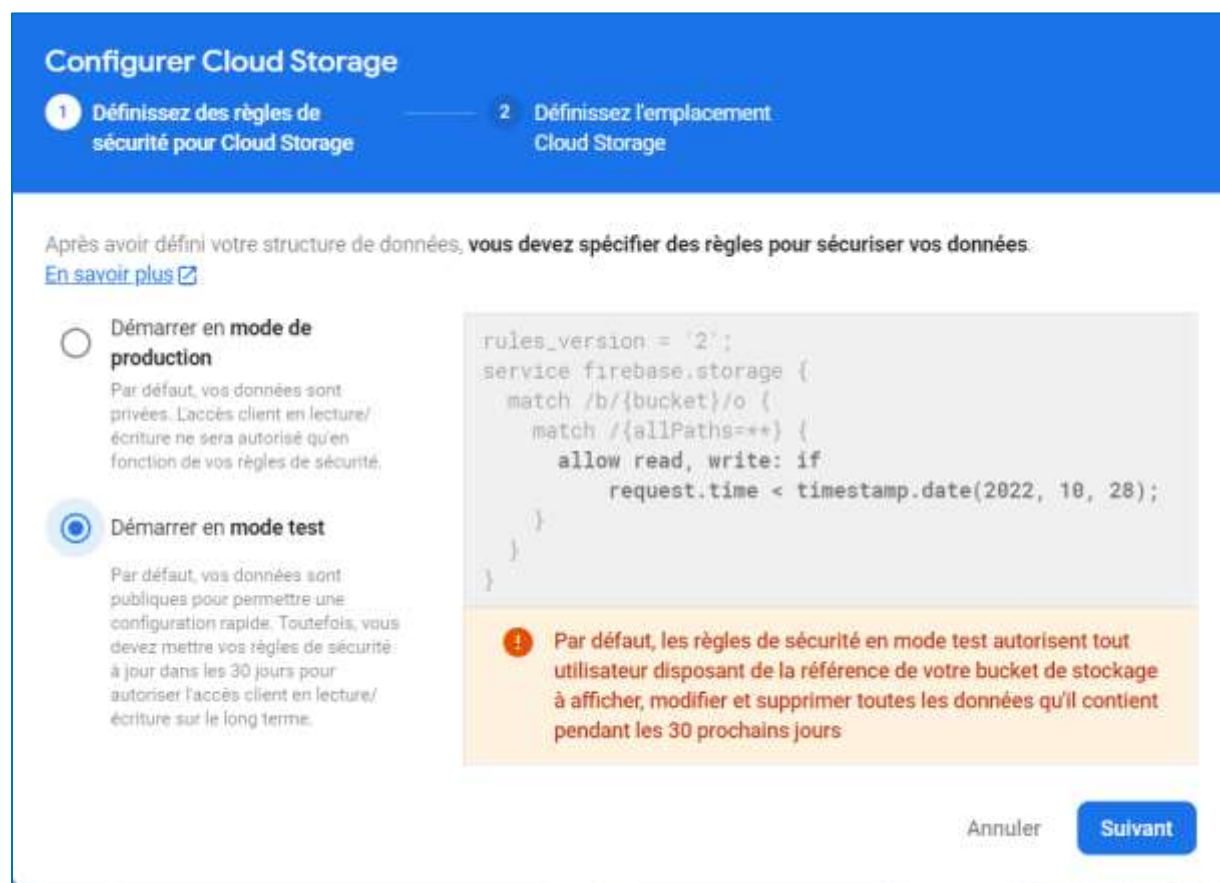


Cliquer sur





Choisir démarrer en mode test puis cliquer sur suivant



Sélectionnez l'emplacement du compartiment de stockage par défaut et cliquez sur OK

Configurer Cloud Storage



Définissez des règles de sécurité pour Cloud Storage



Définissez l'emplacement Cloud Storage

Le paramètre d'emplacement définit l'endroit où sont stockés votre bucket Cloud Storage par défaut et ses données.



Une fois défini, cet emplacement ne peut plus être modifié. De plus, ce paramètre d'emplacement s'appliquera par défaut à Cloud Firestore.

[En savoir plus](#)

Emplacement Cloud Storage

nam5 (us-central)

Les clients ayant souscrit une formule Blaze peuvent choisir d'autres emplacements pour des buckets supplémentaires

Annuler

OK

Configurer Cloud Storage



Définissez des règles de sécurité pour Cloud Storage



Définissez l'emplacement Cloud Storage

Le paramètre d'emplacement définit l'endroit où sont stockés votre bucket Cloud Storage par défaut et ses données.



Une fois défini, cet emplacement ne peut plus être modifié. De plus, ce paramètre d'emplacement s'appliquera par défaut à Cloud Firestore.

[En savoir plus](#)

Création du bucket par défaut...

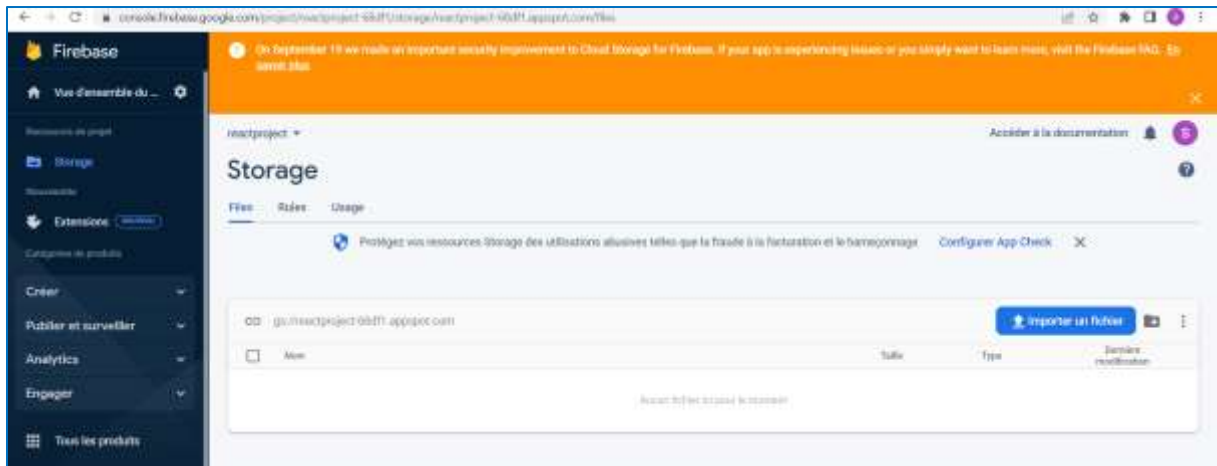
Emplacement Cloud Storage

nam5 (us-central)

Les clients ayant souscrit une formule Blaze peuvent choisir d'autres emplacements pour des buckets supplémentaires

Annuler

OK



Vous êtes maintenant prêt à commencer à télécharger des fichiers sur le stockage Firebase.

Remarque : si vous choisissez le mode **production**, veuillez à changer les **rules**, en ajoutant « *if true* », puis publier.

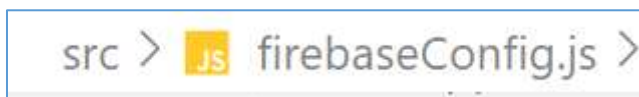


Ajouter Firebase dans React JS

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

```
npm install firebase
```

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'appel de `getStorage` car on va en avoir besoin pour le stockage des images.

```
import { initializeApp } from "firebase/app";
import { getStorage } from "firebase/storage";

// Initialize Firebase
const app = initializeApp ({
```

```

    apiKey: <apiKey>,
    authDomain: <authDomain>,
    projectId: <projectId>,
    storageBucket: <storageBucket>,
    messagingSenderId: <messagingSenderId>,
    appId: <appId>,
    measurementId: <measurementId>,
  });

// Firebase storage reference
const storage = getStorage(app);
export default storage;

```

Ce qui donne ce code (Modifier les paramètres de votre compte) :

```

import { initializeApp } from "firebase/app";
import { getStorage } from "firebase/storage";

// Initialize Firebase
const app = initializeApp ({
  apiKey: "AIzaSyBaxrbWZQSNFI230jsJHSeRB3RyLaJ1RXY",
  authDomain: "reactproject-68df1.firebaseio.com",
  projectId: "reactproject-68df1",
  storageBucket: "reactproject-68df1.appspot.com",
  messagingSenderId: "373746469631",
  appId: "1:373746469631:web:416496270e3d4643a45aca"
});

// 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).

src > utils >  UploadFirebase.js >

```

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 avec Filepond et Upload Firebase

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 react-filepond filepond-plugin-image-exif-orientation filepond-plugin-image-preview
```

Les images seront sélectionnées puis seront uploadées dans Firebase. Ce dernier nous retournera l'url correspondantes.

src > components >  CreateProduct.js >

```
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 axios from "axios";

import Modal from 'react-bootstrap/Modal';

import Grid from '@mui/material/Grid';
import Select from '@mui/material/Select';
import MenuItem from '@mui/material/MenuItem';
import Box from '@mui/material/Box';

import AddCircleIcon from '@mui/icons-material/AddCircle';

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 CreateProduct = ({ addProduct, categories }) => {

  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 [show, setShow] = useState(false);
  const handleClose = () => setShow(false);
  const handleShow = () => setShow(true);

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

```
const URL = "http://localhost:3001/api/"
```

```
const handleSubmit = (url) => {
```

```
    const newProduct = {  
        reference,  
        designation,  
        prix,  
        marque,  
        qtestock,  
        imageart:url,  
        scategorieID  
    };
```

```
    //faire le add dans la BD
```

```
    axios.post(URL+"articles",newProduct)  
    .then(res => {  
        const response = res.data;
```

```
        // faire le add dans le tableau affiché
```

```
        addProduct(response);
```

```
        //vider le form
```

```
        setReference('');
```

```
        setDesignation('');
```

```
        setPrix('');
```

```
        setMarque('');
```

```
        setQtestock('');
```

```
        setImageart('');
```

```
        setScategorieID('');
```

```
        setValidated(false);
```

```
        setFile("")
```

```
        handleClose()
```

```
    })
```

```
    .catch(error=>{  
        console.log(error)  
        alert("Erreur ! Insertion non effectuée")  
    })
```

```
}
```

```
const handleUpload = (event) => {  
    event.preventDefault();  
    const form = event.currentTarget;  
    if (form.checkValidity() === true) {  
        if (!file[0].file) {
```

```

        alert("Please upload an image first!");
    }
    else {
        console.log(file[0].file)
        resultHandleUpload(file[0].file,event);
    }
    if (!file[0].file) {
        alert("Please upload an image first!");
    }
}
setValidated(true);
};

```

```

const resultHandleUpload = async(file) => {

    try {

        const url = await UploadFirebase(file);
        console.log(url);

        handleSubmit(url)
    } catch (error) {
        console.log(error);
    }

}

```

```

const handleReset={()=>{
    setReference('');
    setDesignation('');
    setPrix('');
    setMarque('');
    setQttestock('');
    setImageart('');
    setScategorieID('');
    setValidated(false);
    setFile('')

```

```

    handleClose()

}

return (
    <div>

        <Button className="btn btn-primary" style={{'margin':10,'left':10}}
onClick={handleShow}>
<AddCircleIcon /> Nouveau
</Button>

```



```
<Modal show={show} onHide={handleClose}>
```

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

```
    <Modal.Header closeButton>
```

```
      <h2>Create Product</h2>
```

```
    </Modal.Header>
```

```
    <Modal.Body>
```

```
      <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>'
/>

```

```

</Form.Group>
<Form.Group as={Col} md="12">
<Form.Label>S/Catégorie</Form.Label>

<Box sx={{ minWidth: 400 }}>
<Select sx={{ width: 400 }}

```

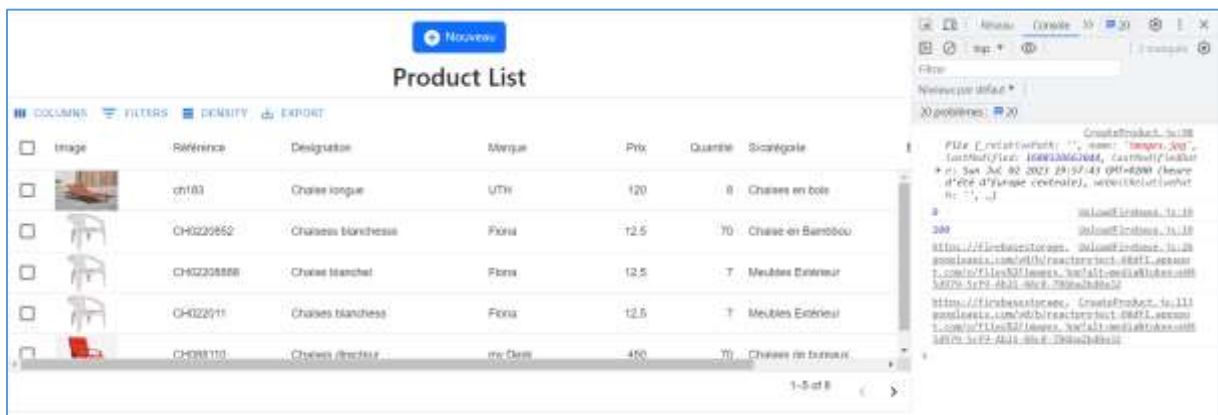
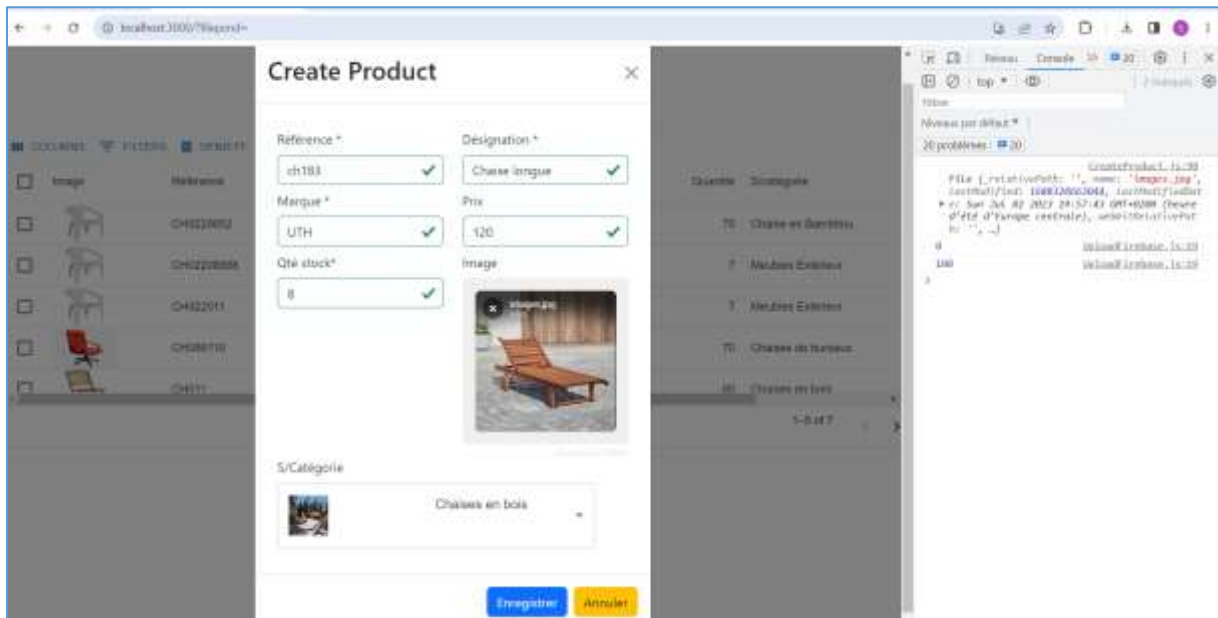
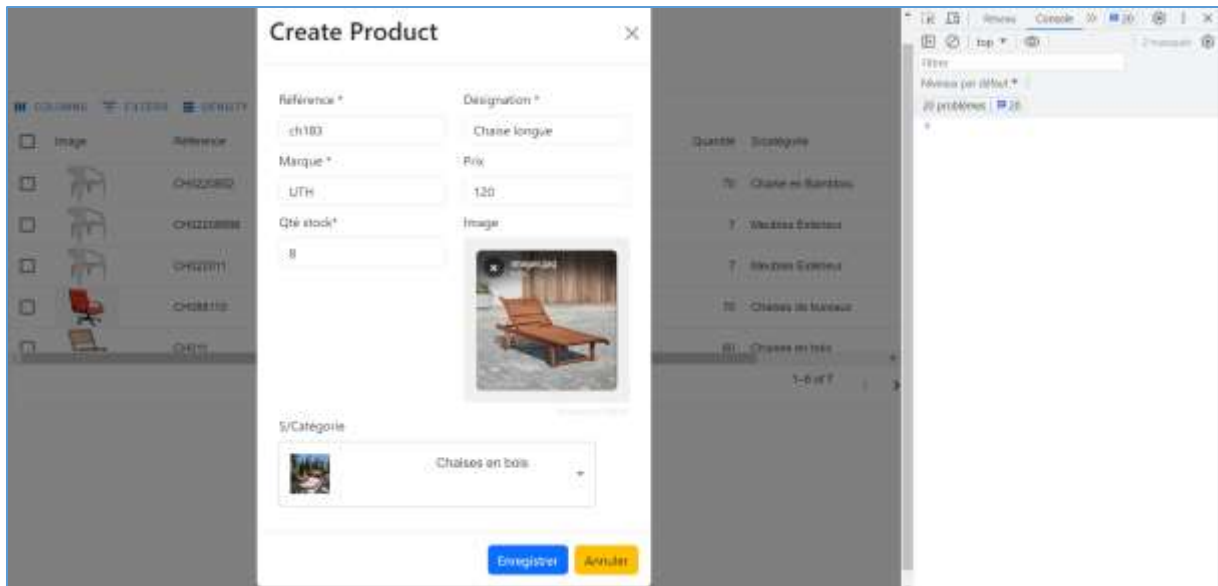
```

        label="S/Catégories"
        value={scategorieID}
        onChange={(e)=>setScategorieID(e.target.value)}
    >
<MenuItem></MenuItem>
{categories.map((scat)=><MenuItem key={scat._id}
value={scat._id}>
    <Grid container spacing={2}>
        <Grid item xs={6}>
            <img src= {scat.imagescat} alt="" width="50" height="50" />
        </Grid>
        <Grid item xs={6}>
            {scat.nomscategorie}
        </Grid>
    </Grid>
</MenuItem>
)}

</Select>
</Box>
</Form.Group>
</Row>
</div>
</div>
</div>
</Modal.Body>
<Modal.Footer>
<Button type="submit">Enregistrer</Button>
<Button type="button" className="btn btn-warning"
onClick={()=>handleReset()}>Annuler</Button>
</Modal.Footer>
</Form>
</Modal>
    </div>
);
};

export default CreateProduct;

```



src > components > JS EditProduct.js >

```
import React, { useState, useEffect, useCallback } 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 axios from "axios";

import Modal from 'react-bootstrap/Modal';

import Grid from '@mui/material/Grid';
import Select from '@mui/material/Select';
import MenuItem from '@mui/material/MenuItem';
import Box from '@mui/material/Box';

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 EditProduct = ({ product, updateProduct ,scategories}) => {

  const [_id,setId] = useState();
  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 fetchEditArticle = useCallback(async () => {
    setId(product._id)
    setReference(product.reference);
    setDesignation(product.designation);
```

```

    setPrix(product.prix);
    setMarque(product.marque);
    setQtestock(product.qtestock);
    setImageart(product.imageart);
    setScategorieID(product.scategorieID._id);
    setShow(true)
  }, [product]);

```

```

useEffect(() => {
  fetchEditArticle();
}, [fetchEditArticle]);

```

```

const [show, setShow] = useState(true);
const handleClose = () => setShow(false);
const handleShow = () => setShow(true);

```

```

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

```

```

const URL = "http://localhost:3001/api/"

```

```

const handleSubmit = (url) => {

```

```

  setImageart(url);

```

```

  const updatedProduct = {
    ...product,
    _id,
    reference,
    designation,
    prix,
    marque,
    qtestock,
    imageart:url,
    scategorieID
  };

```

```

  //update dans la BD

```

```

  axios.put(URL + 'articles/' + product._id, updatedProduct)
    .then(res => {

```

```

    //update dans le tableau affiché

```

```

    updateProduct(res.data);

```

```

    //vider le form

```

```

    setReference('');
    setDesignation('');
    setPrix('');
    setMarque('');
    setQtestock('');
    setImageart('');

```

```

    setScategorieID('');
    setValidated(false);
    setFile("")
  }) .catch(error=>{
    console.log(error)
    alert("Erreur ! Modification non effectuée")
  })

```

```

    handleClose()

```

```

  };

```

```

const handleUpload = (event) => {
  event.preventDefault();
  const form = event.currentTarget;
  if (form.checkValidity() === true) {
    if (!file) {
      const url = imageart;
      handleSubmit(url);
    }
    else {
      console.log(file[0].file)
      resultHandleUpload(file[0].file);
    }
  }
  setValidated(true);
};

const resultHandleUpload = async(file) => {

  try {

    const url = await UploadFirebase(file);
    console.log(url);

    handleSubmit(url)
  } catch (error) {
    console.log(error);
  }
}

const handleReset={()=>{
  setReference('');
  setDesignation('');
  setPrix('');
  setMarque('');

```



```

        setQttestock('');
        setImageart('');
        setScategorieID('');
        setValidated(false);
        setFile("")
        handleClose()
    }

```

```

return (
    <div>
        <Button className="btn btn-primary" style={{'margin':10,'left':10}}
onClick={handleShow}>
Nouveau
    </Button>
    <Modal show={show} onHide={handleClose}>

```

```

        <Form noValidate validated={validated} onSubmit={handleUpload}>
    <Modal.Header closeButton>
    <h2>Edit Product</h2>
    </Modal.Header>
    <Modal.Body>
    <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>

```

```

{!file?<img src={imageart} style={{width:50, height:50}}/> :null}
<FilePond
    files={file}
    allowMultiple={false}
    onupdatefiles={setFile}
    labelIdle='<span class="filepond--label-action">Browse
One</span>'

    />
</Form.Group>
<Form.Group as={Col} md="12">
<Form.Label>S/Catégorie</Form.Label>
<Box sx={{ minWidth: 400 }}>
<Select sx={{ width: 400 }}
    label="S/Catégories"
    value={scategorieID}
    onChange={(e)=>setScategorieID(e.target.value)}
    >
<MenuItem></MenuItem>
{scategories.map((scat)=><MenuItem key={scat._id}
value={scat._id}>
    <Grid container spacing={2}>
    <Grid item xs={6}>
        <img src= {scat.imagescat} alt="" width="50" height="50" />
    </Grid>
    <Grid item xs={6}>
        {scat.nomscategorie}
    </Grid>
    </Grid>
</MenuItem>
)}}
</Select>
</Box>
</Form.Group>
</Row>
</div>
</div>
</div>
</Modal.Body>
<Modal.Footer>
<Button type="submit">Enregistrer</Button>
<Button type="button" className="btn btn-warning"
onClick={()=>handleReset()}>Annuler</Button>
</Modal.Footer>
</Form>
</Modal>
    </div>
);

```

```
};
```

```
export default EditProduct;
```

Edit Product

Référence *

ch183

Désignation *

Chaise longue

Marque *

UTH


Prix

120

Qté stock*


8

Image



Browse One

S/Catégorie

Chaises en bois

Enregistrer

Annuler

Edit Product



Référence *

CH2555

Désignation *

Chaise longue

Marque *

UTH

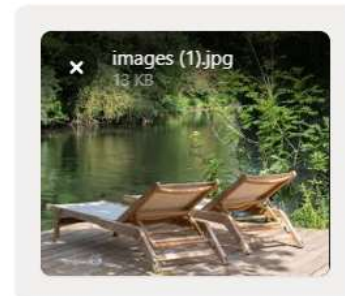
Prix

120

Qté stock*

8

Image



Powered by PQINA

S/Catégorie



Chaises en bois

Enregistrer

Annuler

Product List

[Columns](#) [Filters](#) [Density](#) [Export](#)

<input type="checkbox"/>	Image	Reference	Désignation	Marque	Prix	Quantité	S/Catégorie	<input type="checkbox"/>
<input type="checkbox"/>		CH2555	Chaise longue	UTH	120	8	Chaises en bois	<input checked="" type="checkbox"/>
<input type="checkbox"/>		CH223052	Chaises blanches	Pons	12,5	70	Chaise en Bambou	<input checked="" type="checkbox"/>
<input type="checkbox"/>		CH223058	Chaise blanche	Pons	12,5	7	Mobilier Extérieur	<input checked="" type="checkbox"/>
<input type="checkbox"/>		CH223011	Chaises blanches	Pons	12,5	7	Mobilier Extérieur	<input checked="" type="checkbox"/>
<input type="checkbox"/>		CH494110	Chaises directes	no Desk	400	70	Chaises de bureau	<input checked="" type="checkbox"/>

1-5 of 8

```
editProduct.js:114
File (relativePath: "...", name: "images (1).jpg", lastModified: 158422771348, lastModifiedStr: "Sun Jul 07 2023 20:35:13 GMT+0200 (Heure d'été d'Europe centrale)", lastModifiedStrUTC: "...")
0
@firebase/firestore.js:116
100
@firebase/firestore.js:116
https://firebase.googleapis.com/v1/projects/celestia-90418.firebaseio.com/documents/CH2555?fields=firestore.googleapis.com/documents/CH2555
https://firebase.googleapis.com/v1/projects/celestia-90418.firebaseio.com/documents/CH2555?fields=firestore.googleapis.com/documents/CH2555
```

Edit Product



Référence *

CH088110

Désignation *

Chaises directeur

Marque *

my Desk

Prix

450

Qté stock*

70

Image



[Browse One](#)

Powered by PQINA

S/Catégorie



Chaises de bureaux ▼

Enregistrer

Annuler

Edit Product



Référence *

CH2023

Désignation *

Chaises directeur

Marque *

my Desk

Prix

450

Qté stock*

70

Image



[Browse One](#)

Powered by PQINA

S/Catégorie



Chaises de bureaux

Enregistrer

Annuler

Product List

DOCUMENTS FILTERS DEBUTIFY EXPORT									
<input type="checkbox"/> Image	Reference	Désignation	Marque	Prix	Quantité	S/Catégorie	Edit	Delete	
<input type="checkbox"/>	CH0555	Chaise longue	UTM	120	8	Chaises en bois			
<input type="checkbox"/>	CH0230552	Chaises blanches	Pfoni	12,5	70	Chaise en Bambou			
<input type="checkbox"/>	CH0230556	Chaise blanche	Pfoni	12,5	7	Mobilier Extérieur			
<input type="checkbox"/>	CH023011	Chaises blanches	Pfoni	12,5	7	Mobilier Extérieur			
<input type="checkbox"/>	CH0033	Chaises directeur	my Desk	450	70	Chaises de bureaux			