Update Produit



import Listproducts from '../../../../../components/products/ListCard';

import React from 'react'

async function getProducts(){

    const res= await fetch('https://backend-ecommerce-jwt-2024.vercel.app/api/articles', { cache: 'no-store' })

    const products = await res.json();

    return products;

}

const ListproductCards = async () => {

  const products = await getProducts();

  return (

    <div>

        <Listproducts products={products} />

    </div>

  )

}

export default ListproductCards



'use client';

import Image from "next/image";

import Link from 'next/link'

import React from 'react'

import { useRouter } from 'next/navigation';

const ListCards = (props) => {

    const router = useRouter()

//Pour actualiser la liste

const[products,setProducts]=React.useState(props.products)

const getProducts = async () => {

 const res= await fetch('https://backend-ecommerce-jwt-2024.vercel.app/api/articles')

  const products = await res.json();

  setProducts(products)

 }

React.useEffect(() => {

  getProducts();

 }, [products]);

    //suppression

   const handleDelete=async(\_id)=>{

           if(window.confirm("supprimer la catégorie O/N")) {

                    console.log(\_id)

                    await ( fetch('https://backend-ecommerce-jwt-2024.vercel.app/api/articles/' + \_id, {

                        method: "DELETE"

                    })).then(()=>{

                        router.push("/products/cards")

                    })

                    .catch(error=>{

                        console.log(error)

                        alert("Erreur ! Suppression non effectuée")

                    })

        }

    }

  return (

    <div className="grid grid-cols-2 md:grid-cols-4 gap-4">

     {

      products.map(product =>{

return  <div className="col-sm-3" key={product.\_id}>

<div className="card">

<Image

src={product.imageart}

width={300}

height={200}

alt={product.designation}

priority={true}

/>

<h6>{product.designation}</h6>

<p>{product.marque}</p>

<p>{product.prix} TND</p>

<div className="flex mt-4 space-x-3 md:mt-6">

            <Link href={`/admin/products/update/${product.\_id}`} >Edit</Link>

            <span onClick={()=>handleDelete(product.\_id)} >Delete</span>

</div>

</div>

</div>

      })

     }

    </div>

  )

}

export default ListCards



import React from 'react'

import UpdateProduct from '../../../../../../components/products/UpdateProduct'

async function getScategories(){

  const res= await fetch('https://backend-ecommerce-jwt-2024.vercel.app/api/scategories')

  const scategories = await res.json();

  return scategories;

}

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

  const scategories = await getScategories();

  return (

    <div>

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

    </div>

  )

}

export default updateproductPage



'use client';

import React, { useState } 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/ArticleService';

async function getProductDetails(id){

  try {

 /\*  const res = await fetch (`http://localhost:3001/api/articles/${id}`);

   const product = await res.json();\*/

   const product = await  fetchArticleById(id).then((result) =>{return result.data} )

    return product;

  }

  catch (error) {

      console.log(error);

  }

}

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);

  React.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

      }

      console.log(params.id)

     console.log(article)

   /\* await ( fetch('http://localhost:3001/api/articles/' + params.id, {

        method: 'PUT',

        body: article

    }))\*/

    await editArticle(article)

      .then(res=>{

      console.log("Update OK",res);

      setReference("");

      setDesignation("");

      setPrix("");

      setMarque("");

      setQtestock("");

      setImageart("");

      setScategorieID("");

      setValidated(false);

      setFile("")

      alert("Updated with Success")

      router.push("/admin/products/cardsList")

      })

      .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);

        console.log(url)

       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

