CRUDs Article en utilisant les hooks et les props

**Pagination Backend**

Ajouter le code suivant dans le fichier articles.route.js du dossier backend

router.get('/art/pagination', async(req, res) => {

    const page = parseInt(req.query.page);

    const pageSize = parseInt(req.query.pageSize);

    // Calculate the start and end indexes for the requested page

    const startIndex = (page - 1) \* pageSize;

    const endIndex = page \* pageSize;

    const articles = await Article.find().populate("scategorieID").exec()

    // Slice the products array based on the indexes

    const paginatedProducts = articles.slice(startIndex, endIndex);

    // Calculate the total number of pages

    const totalPages = Math.ceil(articles.length / pageSize);

    // Send the paginated products and total pages as the API response

    res.json({ products: paginatedProducts, totalPages });

  });

**Tester le service avec thunder client**

**http://localhost:3002/api/articles/art/pagination?page=3&pageSize=5**

**Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement**

**Update git**

git add .

git commit -a -m "version2"

git push --set-upstream origin main

**Adresse du backend vercel est :**

https://ecombackend-rouge.vercel.app/api/articles

Vérifier cors dans le fichier app.js du dossier backend

app.use(cors({

    origin:'\*'

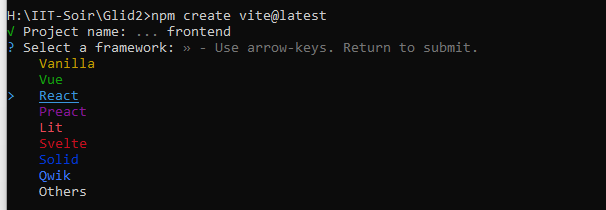
}))

**Partie Frontend**

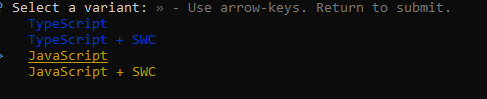
Créer un nouveau projet intitule frontendhooks sous le dossier Ecommerce dont vous créer.

C:\ Ecommerce>npm create vite@latest

Nom du projet : frontendhooks



Choisir **React** et appuyer sur la touche Enter



Choisir **JavaScript**

C:\Ecommerce/>Cd frontendhooks

C:\Ecommerce/frontend/>

npm install

Code .

Terminal🡪new terminal

C:\Ecommerce/frontend>**npm run dev**



Pour changer le port, ouvrir le fichier package.json

"scripts": {

**"dev": "vite --port 3000",**

**npm install axios react-router-dom bootstrap react-bootstrap**

ajouter la ligne suivante dans le fichier app.jsx

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

* **Installer fontawesome**

**Site :** **https://fontawesome.com/search?o=r&m=free**

npm install @fortawesome/fontawesome-free

importer fontawesome dans le fichier app.jsx

import "@fortawesome/fontawesome-free/css/all.css";

Céer le dossier Api sous src

Qjouter le fichier axios.js dans le dossier Api

import axios from 'axios'

export default axios.create({

baseURL:" https://ecombackend-rouge.vercel.app/api/"

})

Créer dossier services sous src

**articleservice.js**

import axios from "../Api/axios";

const ARTICLE\_API="articles"

export const fetcharticles=async()=> {

return await axios.get(ARTICLE\_API);

}

export const fetcharticleById=async(articleId)=> {

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

    }

export const deletearticle=async(articleId) =>{

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

    }

export const addarticle=async(article)=> {

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

    }

export const editarticle=(article) =>{

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

    }

    export const fetcharticlesPagination=async(page,limit)=> {

        return await axios.get(ARTICLE\_API + `/art/pagination?page=${page}&pageSize=${limit}`)

        }

**categorieservice.js**

import axios from "../Api/axios";

const CATEGORIE\_API="/categories"

export const fetchcategories=async()=> {

return await axios.get(CATEGORIE\_API);

}

export const fetchcategorieById=async(categorieId)=> {

return await axios.get(CATEGORIE\_API + '/' + categorieId);

}

export const deletecategorie=async(categorieId) =>{

return await axios.delete(CATEGORIE\_API + '/' + categorieId);

}

export const addcategorie=async(categorie)=> {

return await axios.post(CATEGORIE\_API,categorie);

}

export const editcategorie=(categorie) =>{

return axios.put(CATEGORIE\_API + '/' + categorie.\_id, categorie);

}

**scategorieservice.js**

import axios from "../Api/axios";

const SCATEGORIE\_API="scategories"

export const fetchscategories=async()=> {

return await axios.get(SCATEGORIE\_API);

}

export const fetchscategorieById=async(scategorieId)=> {

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

}

export const deletescategorie=async(scategorieId) =>{

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

}

export const addscategorie=async(scategorie)=> {

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

}

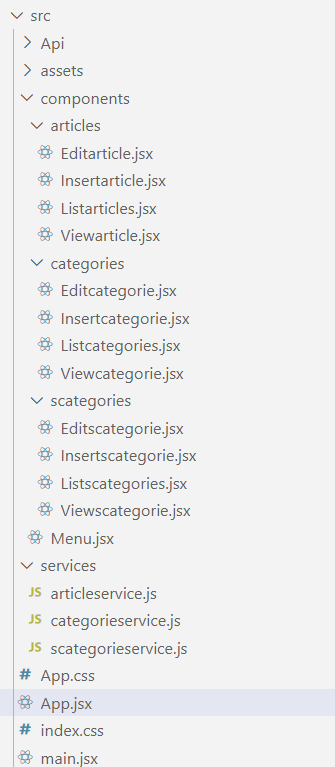
export const editscategorie=(scategorie) =>{

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

}

* Créer l’arborescence du site, comme indiqué ci-dessous

Créer l’arborescence suivante sous le dossier **src** de votre frontend:



Composant Listarticles.jsx (rafce)

Ajouter la route de chaque composant dans le fichier App.jsx

import { BrowserRouter as Router,Routes, Route } from "react-router-dom";

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

import "@fortawesome/fontawesome-free/css/all.css";

import Listarticles from './components/articles/Listarticles';

import Insertarticle from './components/articles/Insertarticle';

import Editarticle from './components/articles/Editarticle';

import Viewarticle from './components/articles/Viewarticle';

import Listcategories from './components/categories/Listcategories';

import Insertcategorie from './components/categories/Insertcategorie';

import Editcategorie from './components/categories/Editcategorie';

import Viewcategorie from './components/categories/Viewcategorie';

import Listscategories from './components/scategories/Listscategories';

import Insertscategorie from './components/scategories/Insertscategorie';

import Editscategorie from './components/scategories/Editscategorie';

import Viewscategorie from './components/scategories/Viewscategorie';

function App() {

 return (

<>

<Router>

<Routes>

<Route path="/articles"  element={<Listarticles/>}/>

<Route path="/articles/add" element={<Insertarticle/>}/>

<Route path="/article/edit/:id" element={<Editarticle/>}/>

<Route path="/article/view/:id" element={<Viewarticle/>}/>

<Route path="/categories" exact element={<Listcategories/>}/>

<Route path="/categories/add" element={<Insertcategorie/>}/>

<Route path="/categories/edit/:id" element={<Editcategorie/>}/>

<Route path="/categories/view/:id" element={<Viewcategorie/>}/>

<Route path="/scategories" element={<Listscategories/>}/>

<Route path="/scategories/add" element={<Insertscategorie/>}/>

<Route path="/scategories/edit/:id" element={<Editscategorie/>}/>

<Route path="/scategories/view/:id" element={<Viewscategorie/>}/>

</Routes>

</Router>

</>

  )

}

export default App

accéder au site [https://react-bootstrap.github.io](https://react-bootstrap.github.io/)

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Fichier Menu.jsx

import React from 'react'

import {Nav, Navbar,Container,Form,FormControl,Button} from 'react-bootstrap';

import {Link } from 'react-router-dom'

const Menu = () => {

  return (

    <Navbar bg="primary" variant="dark">

    <Container>

    <Navbar.Brand >Gestion Commerciale</Navbar.Brand>

    <Nav className="me-auto">

      <Nav.Link as={Link} to="/categories">Catégories</Nav.Link>

      <Nav.Link as={Link} to="/scategories">Sous Catégories</Nav.Link>

      <Nav.Link as={Link} to="/articles">Liste des Articles</Nav.Link>

    </Nav>

    </Container>

    <Form className="d-flex">

        <FormControl

          type="search"

          placeholder="Search"

          className="me-2"

          aria-label="Search"

        />

        <Button variant="success">Chercher</Button>

      </Form>

  </Navbar>

  )

}

export default Menu

1. Afficher la liste des articles

 Dans le dossier articles, créer le composant Listarticles.jsx

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

import {fetcharticlesPagination} from "../../services/articleservice"

import "./article.css"

import Affichearticle from './Affichearticle';

const Listarticles = () => {

  const [currentPage, setCurrentPage] = useState(1);

  const [totalPages, setTotalPages] = useState(0);

  const [articles, setArticles] = useState([]);

  const [limit, setLimit]=useState(5)

  const fetchProducts = async (page,limit) => {

    try {

      const res = await fetcharticlesPagination(page,limit)

      setArticles(res.data.products);

      setTotalPages(res.data.totalPages);

    } catch (error) {

      console.log(error);

    }

  };

 useEffect(() => {

    fetchProducts(currentPage,limit);

  }, [currentPage,limit]);

return (

    <div>

      <Affichearticle articles={articles}/>

    </div>

  )

}

export default Listarticles

Composant Affichearticle.jsx

import React from 'react'

const Affichearticle = ({articles}) => {

  return (

    <div className="table-container">

    <table >

    <thead>

    <tr>

        <th >Image</th>

        <th>Référence</th>

        <th>Désignation</th>

        <th>Marque</th>

        <th>Quanité</th>

        <th>Prix</th>

        <th>Modifier</th>

        <th>Supprimer</th>

    </tr>

    </thead>

    <tbody>

        {

            articles.map((art,index)=>

            <tr key={index}>

                <td><img src ={art.imageart} width={80} height={80} /></td>

                <td>{art.reference}</td>

                <td>{art.designation}</td>

                <td>{art.marque}</td>

                <td>{art.qtestock}</td>

                <td>{art.prix}</td>

                <td><button className='edit'>

                <i className="fa-solid fa-pen-to-square"></i>Update</button></td>

                <td><button className="delete" >

                <i class="fa-solid fa-trash"></i>

                Delete</button></td>

            </tr>

            )}

    </tbody>

    <tfoot>

            </tfoot>

    </table>

    </div>

  )

}

export default Affichearticle

Ajouter la pagination dans le composant Listarticles

Composant Listarticles.jsx

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

import {fetcharticlesPagination} from "../../services/articleservice"

import "./article.css"

import Affichearticle from './Affichearticle';

import Pagination from './Pagination';

const Listarticles = () => {

  const [currentPage, setCurrentPage] = useState(1);

  const [totalPages, setTotalPages] = useState(0);

  const [articles, setArticles] = useState([]);

  const [limit, setLimit]=useState(5)

  const fetchProducts = async (page) => {

    try {

      const res = await fetcharticlesPagination(page,limit)

      setArticles(res.data.products);

      setTotalPages(res.data.totalPages);

      console.log(res.data.products)

    } catch (error) {

      console.log(error);

    }

  };

 useEffect(() => {

    fetchProducts(currentPage);

  }, [currentPage,limit]);

  const handlePrevPage = () => {

    if (currentPage > 1) {

      setCurrentPage(currentPage - 1);

    }

  };

  const handleNextPage = () => {

    if (currentPage < totalPages) {

      setCurrentPage(currentPage + 1);

    }

  };

  const handlePageChange = (page) => {

    setCurrentPage(page);

  };

  return (

    <div>

      <Affichearticle articles={articles}/>

      <Pagination handlePrevPage={handlePrevPage}

      handleNextPage={handleNextPage}

      handlePageChange={handlePageChange}

      totalPages={totalPages}

      currentPage ={currentPage }

      />

    </div>

  )

}

export default Listarticles

Composant Pagination.jsx

import React from 'react'

import "./article.css"

const Pagination = ({handlePrevPage,handleNextPage,handlePageChange,totalPages,currentPage }) => {

  return (

    <div className="pagination">

      {/\* Pagination controls \*/}

      <button onClick={()=>handlePrevPage()} disabled={currentPage === 1}

        >

      Previous

      </button>

        {Array.from({ length: totalPages }, (\_, index) => (

          <button

            key={index}

            onClick={() => handlePageChange(index + 1)}

            disabled={currentPage === index + 1}

            className={currentPage === index + 1 ? 'page-link active' : ''}

          >

            {index + 1}

          </button>

        ))}

      <button onClick={()=>handleNextPage()} disabled={currentPage === totalPages}>

        Next

      </button>

      </div>

  )

}

export default Pagination

Ajouter maintenant le nombre de document par page:

Composant Listarticles.jsx

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

import {fetcharticlesPagination} from "../../services/articleservice"

import "./article.css"

import Affichearticle from './Affichearticle';

import Pagination from './Pagination';

const Listarticles = () => {

  const [currentPage, setCurrentPage] = useState(1);

  const [totalPages, setTotalPages] = useState(0);

  const [articles, setArticles] = useState([]);

  const [limit, setLimit]=useState(5)

  const fetchProducts = async (page,limit) => {

    try {

      const res = await fetcharticlesPagination(page,limit)

      setArticles(res.data.products);

      setTotalPages(res.data.totalPages);

    } catch (error) {

      console.log(error);

    }

  };

 useEffect(() => {

    fetchProducts(currentPage,limit);

  }, [currentPage,limit]);

  const handlePrevPage = () => {

    if (currentPage > 1) {

      setCurrentPage(currentPage - 1);

    }

  };

  const handleNextPage = () => {

    if (currentPage < totalPages) {

      setCurrentPage(currentPage + 1);

    }

  };

  const handlePageChange = (page) => {

    setCurrentPage(page);

  };

  const handleLimitChange = (event) => {

    setLimit(parseInt(event.target.value, 10));

    setCurrentPage(1);

  };

  return (

    <div>

 <Affichearticle articles={articles} handleLimitChange={handleLimitChange} limit={limit}/>

      <Pagination handlePrevPage={handlePrevPage}

      handleNextPage={handleNextPage}

      handlePageChange={handlePageChange}

      totalPages={totalPages}

      currentPage ={currentPage }

      />

    </div>

  )

}

export default Listarticles

Composant Affichearticle.jsx

import React from 'react'

import "./article.css"

const Affichearticle = ({articles,handleLimitChange,limit}) => {

  return (

    <div className="table-container">

    <table >

    <thead>

    <tr>

        <th >Image</th>

        <th>Référence</th>

        <th>Désignation</th>

        <th>Marque</th>

        <th>Quanité</th>

        <th>Prix</th>

        <th>Modifier</th>

        <th>Supprimer</th>

    </tr>

    </thead>

    <tbody>

        {

            articles.map((art,index)=>

            <tr key={index}>

                <td><img src ={art.imageart} width={80} height={80} /></td>

                <td>{art.reference}</td>

                <td>{art.designation}</td>

                <td>{art.marque}</td>

                <td>{art.qtestock}</td>

                <td>{art.prix}</td>

                <td><button className='edit'>

                <i className="fa-solid fa-pen-to-square"></i>Update</button></td>

                <td><button className="delete" >

                <i class="fa-solid fa-trash"></i>

                 Delete</button></td>

            </tr>

            )}

    </tbody>

    <tfoot>

    <tr>

                <td colspan="8">

                <div className="limit-selector-container">

                <label>

                    Afficher &nbsp;

                    <select

                      value={limit}

                      onChange={handleLimitChange}

                    >

                      <option value={5}>5</option>

                      <option value={10}>10</option>

                      <option value={20}>20</option>

                      <option value={100}>100</option>

                    </select>

                  </label>

                  </div>

                  </td>

            </tr>

            </tfoot>

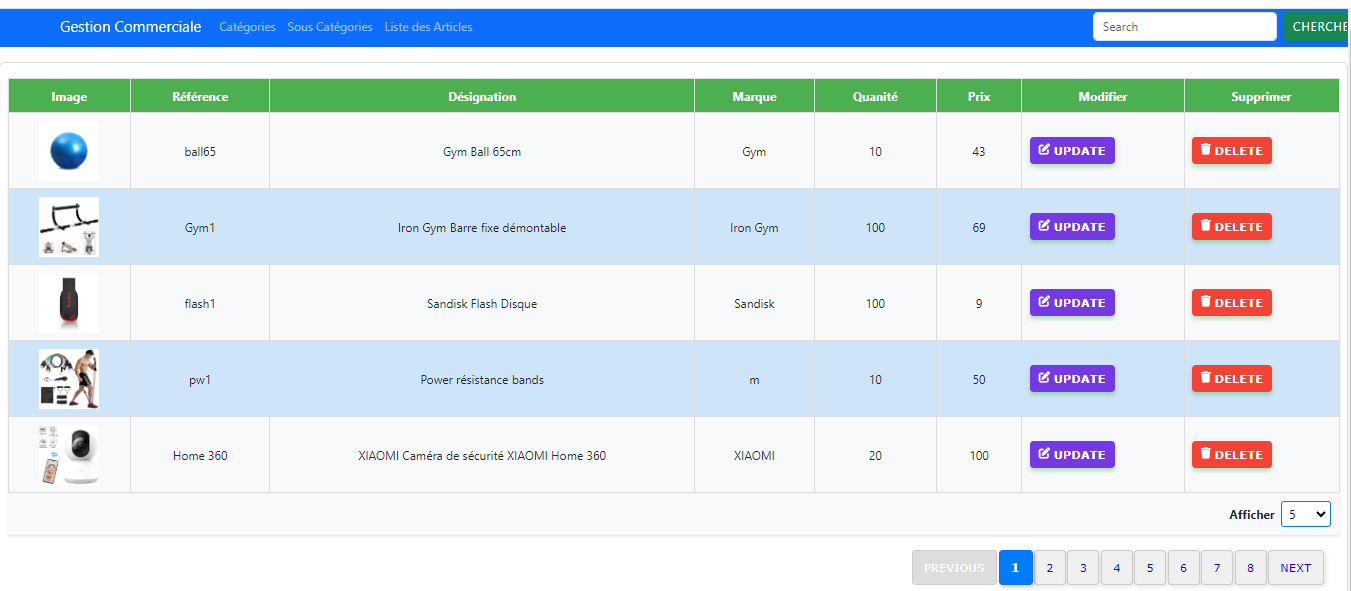
    </table>

    </div>

  )

}

export default Affichearticle



On ajoutera maintenant le bouton Nouveau et la zone de texte de recherche dans le composant Listarticles

On doit commencer par modifier le backend

Fichier article.route.js

router.get('/art/pagination', async(req, res) => {

    const filtre = req.query.filtre || "";

    const page = parseInt(req.query.page);

    const pageSize = parseInt(req.query.pageSize);

    // Calculate the start and end indexes for the requested page

    const startIndex = (page - 1) \* pageSize;

    const endIndex = page \* pageSize;

    const articles = await Article.find({ designation: { $regex: filtre, $options: "i"}}, null, {sort: {'\_id': -1}}).populate("scategorieID").exec()

    // Slice the products array based on the indexes

    const paginatedProducts = articles.slice(startIndex, endIndex);

    // Calculate the total number of pages

    const totalPages = Math.ceil(articles.length / pageSize);

    // Send the paginated products and total pages as the API response

    res.json({ products: paginatedProducts, totalPages });

  });

      Service frontend : articleservice.js

export const fetcharticlesPagination=async(page,limit,searchtext)=> {

        return await axios.get(ARTICLE\_API + `/art/pagination?filtre=${searchtext}&page=${page}&pageSize=${limit}`)

        }

Composant Listarticles.jsx

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

import {fetcharticlesPagination} from "../../services/articleservice"

import "./article.css"

import Affichearticle from './Affichearticle';

import Pagination from './Pagination';

import { Link } from 'react-router-dom';

const Listarticles = () => {

  const [currentPage, setCurrentPage] = useState(1);

  const [totalPages, setTotalPages] = useState(0);

  const [articles, setArticles] = useState([]);

  const [limit, setLimit]=useState(5)

  const [searchText, setSearchText] = useState('');

  const fetchProducts = async (page,limit,searchText) => {

    try {

      const res = await fetcharticlesPagination(page,limit,searchText)

      setArticles(res.data.products);

      setTotalPages(res.data.totalPages);

    } catch (error) {

      console.log(error);

    }

  };

 useEffect(() => {

    fetchProducts(currentPage,limit,searchText);

  }, [currentPage,limit,searchText]);

  const handlePrevPage = () => {

    if (currentPage > 1) {

      setCurrentPage(currentPage - 1);

    }

  };

  const handleNextPage = () => {

    if (currentPage < totalPages) {

      setCurrentPage(currentPage + 1);

    }

  };

  const handlePageChange = (page) => {

    setCurrentPage(page);

  };

  const handleLimitChange = (event) => {

    setLimit(parseInt(event.target.value, 10));

    setCurrentPage(1);

  };

  const handleSearchChange = (event) => {

    setSearchText(event.target.value);

  };

  return (

    <div>

      <div className="table-container-header">

  <button className="new">

    <Link to="/articles/add" style={{ textDecoration: 'none', color: 'inherit' }}>

      <i className="fa-solid fa-plus-square"></i> Nouveau

    </Link>

  </button>

  <div className="search-container">

            <i className="fa-solid fa-search"></i>

            <input

              type="text"

              value={searchText}

              onChange={handleSearchChange}

              placeholder="Rechercher des articles..."

              className="search-input"

            />

          </div>

</div>

      <Affichearticle articles={articles} handleLimitChange={handleLimitChange} limit={limit}/>

      <Pagination handlePrevPage={handlePrevPage}

      handleNextPage={handleNextPage}

      handlePageChange={handlePageChange}

      totalPages={totalPages}

      currentPage ={currentPage }

      />

    </div>

  )

}

export default Listarticles

Créer le composant Headerarticle.jsx

import React from 'react'

import { Link } from 'react-router-dom'

import "./article.css"

const Headerarticle = ({searchText,handleSearchChange}) => {

  return (

    <div className="table-container-header">

  <button className="new">

    <Link to="/articles/add" style={{ textDecoration: 'none', color: 'inherit' }}>

      <i className="fa-solid fa-plus-square"></i> Nouveau

    </Link>

  </button>

  <div className="search-container">

            <i className="fa-solid fa-search"></i>

            <input

              type="text"

              value={searchText}

              onChange={handleSearchChange}

              placeholder="Rechercher des articles..."

              className="search-input"

            />

          </div>

</div>

  )

}

export default Headerarticle

composant Listarticles.jsx

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

import {fetcharticlesPagination} from "../../services/articleservice"

import "./article.css"

import Affichearticle from './Affichearticle';

import Pagination from './Pagination';

import Headerarticle from './Headerarticle';

const Listarticles = () => {

  const [currentPage, setCurrentPage] = useState(1);

  const [totalPages, setTotalPages] = useState(0);

  const [articles, setArticles] = useState([]);

  const [limit, setLimit]=useState(5)

  const [searchText, setSearchText] = useState('');

  const fetchProducts = async (page,limit,searchText) => {

    try {

      const res = await fetcharticlesPagination(page,limit,searchText)

      setArticles(res.data.products);

      setTotalPages(res.data.totalPages);

      console.log(res.data.products)

    } catch (error) {

      console.log(error);

    }

  };

 useEffect(() => {

    fetchProducts(currentPage,limit,searchText);

  }, [currentPage,limit,searchText]);

  const handlePrevPage = () => {

    if (currentPage > 1) {

      setCurrentPage(currentPage - 1);

    }

  };

  const handleNextPage = () => {

    if (currentPage < totalPages) {

      setCurrentPage(currentPage + 1);

    }

  };

  const handlePageChange = (page) => {

    setCurrentPage(page);

  };

  const handleLimitChange = (event) => {

    setLimit(parseInt(event.target.value, 10));

    setCurrentPage(1);

  };

  const handleSearchChange = (event) => {

    setSearchText(event.target.value);

    setCurrentPage(1);

  };

  return (

    <div>

      <Headerarticle searchText={searchText}

      handleSearchChange={handleSearchChange}/>

      <Affichearticle articles={articles} handleLimitChange={handleLimitChange} limit={limit}/>

      <Pagination handlePrevPage={handlePrevPage}

      handleNextPage={handleNextPage}

      handlePageChange={handlePageChange}

      totalPages={totalPages}

      currentPage ={currentPage }

      />

    </div>

  )

}

export default Listarticles

Affichage Card

Créer le dossier client sous le dossier components

Créer le composant Listarticlescard.jsx

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

import {fetcharticlesPagination} from "../../services/articleservice"

import Card from './Card';

import Pagination from '../articles/Pagination';

import "./stylecard.css"

const Listarticlescard = () => {

    const [currentPage, setCurrentPage] = useState(1);

  const [totalPages, setTotalPages] = useState(0);

  const [limit, setLimit]=useState(18)

  const [articles, setArticles] = useState([]);

    const fetchProducts = async (page,limit) => {

        try {

          const res = await fetcharticlesPagination(page,limit,'')

          setArticles(res.data.products);

          setTotalPages(res.data.totalPages);

        } catch (error) {

          console.log(error);

        }

      };

     useEffect(() => {

        fetchProducts(currentPage,limit,'');

      }, [currentPage,limit]);

      const handlePrevPage = () => {

        if (currentPage > 1) {

          setCurrentPage(currentPage - 1);

        }

      };

      const handleNextPage = () => {

        if (currentPage < totalPages) {

          setCurrentPage(currentPage + 1);

        }

      };

      const handlePageChange = (page) => {

        setCurrentPage(page);

      };

  return (

    <>

  <div className="card-container">

      {articles.map((art,index)=>

        <Card  imageart={art.imageart}

        reference={art.reference}

        designation={art.designation}

        prix={art.prix}

        />

      )}

    </div>

    <Pagination handlePrevPage={handlePrevPage}

      handleNextPage={handleNextPage}

      handlePageChange={handlePageChange}

      totalPages={totalPages}

      currentPage ={currentPage }

      />

      </>

  )

}

export default Listarticlescard

Créer le composant Card.jsx

import React from 'react';

import './stylecard.css';

const Card = ({ imageart, reference, designation,prix }) => {

  return (

    <div className="card">

      {imageart && <img src={imageart} alt={reference} />}

      <div className="card-content">

        <h1 className="card-title">{reference}</h1>

        <p className="card-description">{designation.substr(0,20)}</p>

        <h1 className="card-title">Prix : {prix} TND</h1>

        <button className="card-button"><i class="fa-solid fa-basket-shopping"></i>Add to card</button>

      </div>

    </div>

  );

};

export default Card;

Ajouter la route du composant Listarticlescard dans le fichier App.jsx

import { BrowserRouter as Router,Routes, Route } from "react-router-dom";

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

import "@fortawesome/fontawesome-free/css/all.css";

import Listarticles from './components/articles/Listarticles';

import Insertarticle from './components/articles/Insertarticle';

import Editarticle from './components/articles/Editarticle';

import Viewarticle from './components/articles/Viewarticle';

import Listcategories from './components/categories/Listcategories';

import Insertcategorie from './components/categories/Insertcategorie';

import Editcategorie from './components/categories/Editcategorie';

import Viewcategorie from './components/categories/Viewcategorie';

import Listscategories from './components/scategories/Listscategories';

import Insertscategorie from './components/scategories/Insertscategorie';

import Editscategorie from './components/scategories/Editscategorie';

import Viewscategorie from './components/scategories/Viewscategorie';

import Menu from './components/Menu';

import Listarticlescard from "./components/client/Listarticlescard";

function App() {

  return (

    <>

      <Router>

        <Menu/>

<Routes>

<Route path="/articles"  element={<Listarticles/>}/>

<Route path="/articlescard"  element={<Listarticlescard/>}/>

<Route path="/articles/add" element={<Insertarticle/>}/>

<Route path="/article/edit/:id" element={<Editarticle/>}/>

<Route path="/article/view/:id" element={<Viewarticle/>}/>

<Route path="/categories" exact element={<Listcategories/>}/>

<Route path="/categories/add" element={<Insertcategorie/>}/>

<Route path="/categories/edit/:id" element={<Editcategorie/>}/>

<Route path="/categories/view/:id" element={<Viewcategorie/>}/>

<Route path="/scategories" element={<Listscategories/>}/>

<Route path="/scategories/add" element={<Insertscategorie/>}/>

<Route path="/scategories/edit/:id" element={<Editscategorie/>}/>

<Route path="/scategories/view/:id" element={<Viewscategorie/>}/>

</Routes>

</Router>

    </>

  )

}

export default App

Ajouter l’appel du composant Listarticlecard dans le menu Menu.jsx

 <Nav className="me-auto">

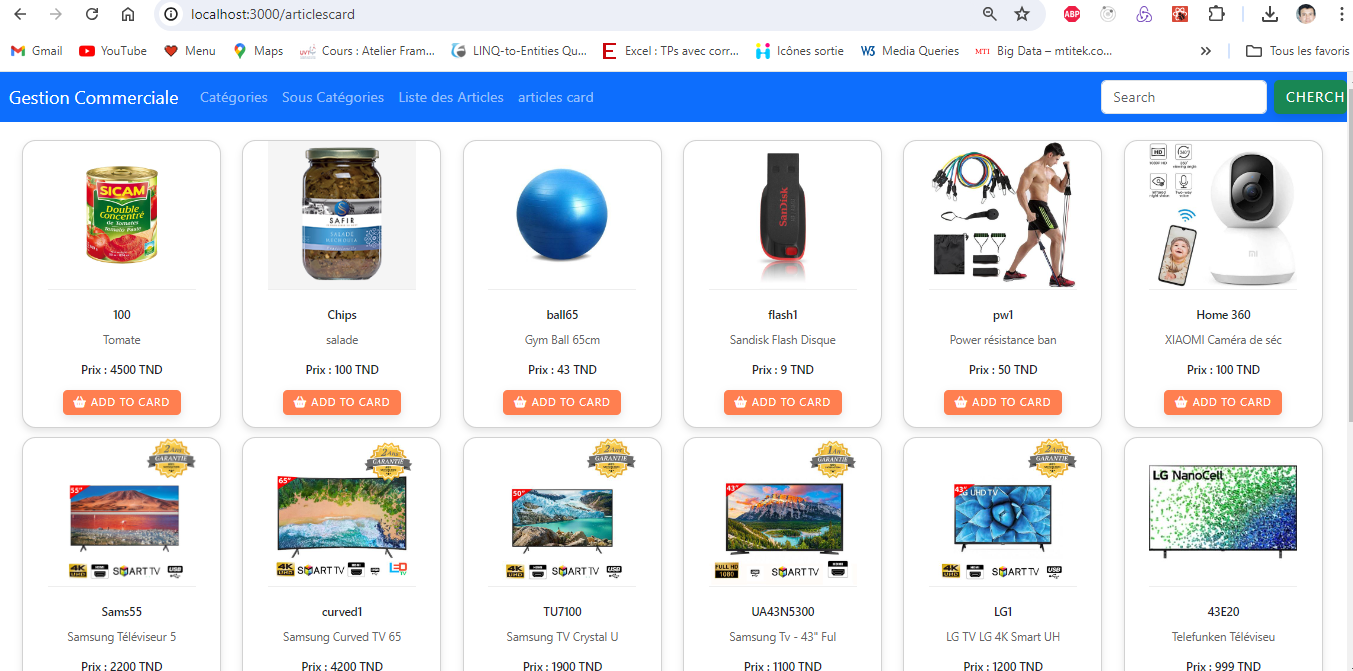
      <Nav.Link as={Link} to="/categories">Catégories</Nav.Link>

      <Nav.Link as={Link} to="/scategories">Sous Catégories</Nav.Link>

      <Nav.Link as={Link} to="/articles">Liste des Articles</Nav.Link>

      <Nav.Link as={Link} to="/articlescard">articles card</Nav.Link>

  </Nav>



Supprimer un article

Composant Listarticles

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

import {fetcharticlesPagination,deletearticle} from "../../services/articleservice"

import "./article.css"

import Affichearticle from './Affichearticle';

import Pagination from './Pagination';

import Headerarticle from './Headerarticle';

const Listarticles = () => {

  const [currentPage, setCurrentPage] = useState(1);

  const [totalPages, setTotalPages] = useState(0);

  const [articles, setArticles] = useState([]);

  const [limit, setLimit]=useState(5)

  const [searchText, setSearchText] = useState('');

  const fetchProducts = async (page,limit,searchText) => {

    try {

      const res = await fetcharticlesPagination(page,limit,searchText)

      setArticles(res.data.products);

      setTotalPages(res.data.totalPages);

    } catch (error) {

      console.log(error);

    }

  };

 useEffect(() => {

    fetchProducts(currentPage,limit,searchText);

  }, [currentPage,limit,searchText]);

  const handlePrevPage = () => {

    if (currentPage > 1) {

      setCurrentPage(currentPage - 1);

    }

  };

  const handleNextPage = () => {

    if (currentPage < totalPages) {

      setCurrentPage(currentPage + 1);

    }

  };

  const handlePageChange = (page) => {

    setCurrentPage(page);

  };

  const handleLimitChange = (event) => {

    setLimit(parseInt(event.target.value, 10));

    setCurrentPage(1);

  };

  const handleSearchChange = (event) => {

    setSearchText(event.target.value);

    setCurrentPage(1);

  };

  const handleDeletearticle = async (id) => {

    try {

     if(window.confirm("confirmer la suppression"))

         {

           await deletearticle(id)

             .then(res=>fetchProducts(currentPage,limit,''))

         }

    } catch (error) {

     alert("impossible de se connecter au serveur")

 }

 }

  return (

    <div>

      <Headerarticle searchText={searchText}

      handleSearchChange={handleSearchChange}/>

      <Affichearticle articles={articles} handleLimitChange={handleLimitChange} limit={limit} handleDeletearticle={handleDeletearticle}/>

      <Pagination handlePrevPage={handlePrevPage}

      handleNextPage={handleNextPage}

      handlePageChange={handlePageChange}

      totalPages={totalPages}

      currentPage ={currentPage }

      />

    </div>

  )

}

export default Listarticles

Composant Affichearticles

import React from 'react'

import "./article.css"

const Affichearticle = ({articles,handleLimitChange,limit,handleDeletearticle}) => {

    console.log("bonjour dans affichearticle")

  return (

    <div className="table-container">

    <table >

    <thead>

    <tr>

        <th >Image</th>

        <th>Référence</th>

        <th>Désignation</th>

        <th>Marque</th>

        <th>Quanité</th>

        <th>Prix</th>

        <th>Modifier</th>

        <th>Supprimer</th>

    </tr>

    </thead>

    <tbody>

        {

            articles.map((art,index)=>

            <tr key={index}>

                <td><img src ={art.imageart} width={80} height={80} /></td>

                <td>{art.reference}</td>

                <td>{art.designation}</td>

                <td>{art.marque}</td>

                <td>{art.qtestock}</td>

                <td>{art.prix}</td>

                <td><center><button className='edit'>

                <i className="fa-solid fa-pen-to-square"></i>Update</button></center></td>

                <td><center><button className="delete" onClick={()=>handleDeletearticle(art.\_id)} >

                <i className="fa-solid fa-trash"></i>

                 Delete</button></center></td>

            </tr>

            )}

    </tbody>

    <tfoot>

    <tr>

                <td colSpan="8">

                <div className="limit-selector-container">

                <label>

                    Afficher &nbsp;

                    <select

                      value={limit}

                      onChange={handleLimitChange}

                    >

                      <option value={5}>5</option>

                      <option value={10}>10</option>

                      <option value={20}>20</option>

                      <option value={100}>100</option>

                    </select>

                  </label>

                  </div>

                  </td>

            </tr>

            </tfoot>

    </table>

    </div>

  )

}

export default Affichearticle

**Installer react-confirm-alert**

**npm install react-confirm-alert**

Dans le composant Listarticles, modifier la méthode handleDeletearticle

On doit Ajouter les imports suivants dans le composant ArticleApp.jsx

import { confirmAlert } from 'react-confirm-alert'; // Import

import 'react-confirm-alert/src/react-confirm-alert.css'; // Import css

const handleDeletearticle = async (id,ref) => {

      confirmAlert({

        title: "Confirm delete...",

        message: " supprimer l' article: " + ref,

        buttons: [

        {

        label: 'Oui',

        onClick: () => deletearticle(id)

.then(res=>fetchProducts(currentPage,limit,''))

        .catch(error=>console.log(error))

        },

        {

        label: 'Non',

        }

        ]

        });

      }

Composant Affichearticles

<td><center><button className='edit'>

 <i className="fa-solid fa-pen-to-square"></i>Update</button></center></td>

  <td><center><button className="delete" onClick={()=>handleDeletearticle(art.\_id,art.reference)} >

<i className="fa-solid fa-trash"></i>Delete</button></center></td>

Ajouter un article

**Composant Insertarticle**

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

import "./insertarticle.css"

import { Modal } from 'react-bootstrap';

import {fetchscategories} from "../../services/scategorieservice"

import {addarticle} from "../../services/articleservice"

const Insertarticle = ({show,handleClose,fetchProducts,limit}) => {

  const[article,setArticle]=useState({})

  const[scategories,setScategories]=useState([])

  const loadscategories=async()=>{

    try {

      const res = await fetchscategories()

      setScategories(res.data);

     console.log(res.data)

    } catch (error) {

      console.log(error);

    }

  }

  useEffect(() => {

    console.log("bonjour")

    loadscategories()

  }, [])

  const handleSubmit = async(event) => {

    event.preventDefault();

    // Logique pour soumettre le formulaire

   await addarticle(article)

   fetchProducts(1,limit,'')

   handleClose()

    // Réinitialiser les champs du formulaire

   setArticle({})

  };

  return (

    <div className="form-container">

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

      <form  className="article-form">

      <Modal.Header closeButton>

 <h2>Ajouter Article</h2>

</Modal.Header>

<Modal.Body>

        <div className="form-grid">

          <div className="form-group">

            <label htmlFor="title">Référence</label>

            <input

              type="text"

              id="reference"

              value={article.reference}

              onChange={(e) => setArticle({...article,reference:e.target.value})}

              className="form-input"

              placeholder="Entrez référence article"

            />

          </div>

          <div className="form-group">

            <label htmlFor="description">Désignation</label>

            <input

            type="text"

              id="designation"

              value={article.designation}

              onChange={(e) => setArticle({...article,designation:e.target.value})}

              className="form-input"

              placeholder="Entrez la désignation article"

            />

          </div>

          <div className="form-group">

            <label htmlFor="marque">Marque</label>

            <input

              type="text"

              id="marque"

              value={article.marque}

              onChange={(e) => setArticle({...article,marque:e.target.value})}

              className="form-input"

              placeholder="Entrez marque"

            />

          </div>

          <div className="form-group">

            <label htmlFor="quantite">Quantité</label>

            <input

              type="number"

              id="qtestock"

              value={article.qtestock}

              onChange={(e) => setArticle({...article,qtestock:e.target.value})}

              className="form-input"

              placeholder="Entrez quantité stock"

            />

          </div>

          <div className="form-group">

            <label htmlFor="prix">Prix</label>

            <input

              type="number"

              required

              id="prix"

              value={article.prix}

              onChange={(e) => setArticle({...article,prix:e.target.value})}

              className="form-input"

              placeholder="Entrez Quantité stock"

            />

          </div>

          <div className="form-group">

            <label htmlFor="prix">Catégorie</label>

            <select

                id="category"

                className="form-control"

                value={article.scategorieID}

                onChange={(e) => setArticle({...article,scategorieID:e.target.value})}

              >

                {scategories.map((scat,index)=>

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

              )}

               </select>

          </div>

          <div className="form-group">

            <label htmlFor="prix">Image</label>

            <input

              type="text"

              required

              id="imageart"

              value={article.imageart}

              onChange={(e) => setArticle({...article,imageart:e.target.value})}

              className="form-input"

              placeholder="Image"

            />

          </div>

        </div>

        </Modal.Body>

        <Modal.Footer>

        <button type="button" className="form-submit-button" onClick={(e)=>handleSubmit(e)}>Enregistrer</button>

        <button type="reset" className="form-reset-button" onClick={()=>handleClose()}>Annuler</button>

        </Modal.Footer>

      </form>

      </Modal>

    </div>

  );

}

export default Insertarticle

**Composant Listarticles.jsx**

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

import {fetcharticlesPagination,deletearticle} from "../../services/articleservice"

import "./article.css"

import Affichearticle from './Affichearticle';

import Pagination from './Pagination';

import Headerarticle from './Headerarticle';

import { confirmAlert } from 'react-confirm-alert'; // Import

import 'react-confirm-alert/src/react-confirm-alert.css'; // Import css

import Insertarticle from './Insertarticle';

const Listarticles = () => {

  const [currentPage, setCurrentPage] = useState(1);

  const [totalPages, setTotalPages] = useState(0);

  const [articles, setArticles] = useState([]);

  const [limit, setLimit]=useState(5)

  const [searchText, setSearchText] = useState('');

  const [show, setShow] = useState(false);

  const handleClose = () => setShow(false);

  const handleShow = () => setShow(true);

  const fetchProducts = async (page,limit,searchText) => {

    try {

      const res = await fetcharticlesPagination(page,limit,searchText)

      setArticles(res.data.products);

      setTotalPages(res.data.totalPages);

    } catch (error) {

      console.log(error);

    }

  };

 useEffect(() => {

    fetchProducts(currentPage,limit,searchText);

  }, [currentPage,limit,searchText]);

  const handlePrevPage = () => {

    if (currentPage > 1) {

      setCurrentPage(currentPage - 1);

    }

  };

  const handleNextPage = () => {

    if (currentPage < totalPages) {

      setCurrentPage(currentPage + 1);

    }

  };

  const handlePageChange = (page) => {

    setCurrentPage(page);

  };

  const handleLimitChange = (event) => {

    setLimit(parseInt(event.target.value, 10));

    setCurrentPage(1);

  };

  const handleSearchChange = (event) => {

    setSearchText(event.target.value);

    setCurrentPage(1);

  };

  const handleDeletearticle = async (id,ref) => {

      confirmAlert({

        title: "Confirm delete...",

        message: " supprimer l' article: " + ref,

        buttons: [

        {

        label: 'Oui',

        onClick: () => deletearticle(id)

        .then(res=>fetchProducts(currentPage,limit,''))

        .catch(error=>console.log(error))

        },

        {

        label: 'Non',

        }

        ]

        });

      }

  return (

    <div>

      <div className="table-container-header">

  <button className="new" onClick={handleShow}>

      <i className="fa-solid fa-plus-square"></i> Nouveau

  </button>

      <Headerarticle searchText={searchText}

      handleSearchChange={handleSearchChange}/>

      <Affichearticle articles={articles} handleLimitChange={handleLimitChange} limit={limit} handleDeletearticle={handleDeletearticle}/>

      </div>

      <Pagination handlePrevPage={handlePrevPage}

      handleNextPage={handleNextPage}

      handlePageChange={handlePageChange}

      totalPages={totalPages}

      currentPage ={currentPage }

      />

{show && <Insertarticle

          show={show}

          handleClose={handleClose}

          fetchProducts={fetchProducts}

          limit={limit}

          />  }

    </div>

  )

}

export default Listarticles

**Ajout article avec filepond et cloudinary**

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

composant Insertarticle

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

import "./insertarticle.css"

import axios from "axios"

import { Modal } from 'react-bootstrap';

import {fetchscategories} from "../../services/scategorieservice"

import {addarticle} from "../../services/articleservice"

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 Insertarticle = ({show,handleClose,fetchProducts,limit}) => {

  const[article,setArticle]=useState({})

  const[scategories,setScategories]=useState([])

  const [files, setFiles] = useState([]);

  const loadscategories=async()=>{

    try {

      const res = await fetchscategories()

      setScategories(res.data);

     console.log(res.data)

    } catch (error) {

      console.log(error);

    }

  }

  useEffect(() => {

    console.log("bonjour")

    loadscategories()

  }, [])

  const handleSubmit = async(event) => {

    event.preventDefault();

    // Logique pour soumettre le formulaire

   await addarticle(article)

   fetchProducts(1,limit,'')

   handleClose()

    // Réinitialiser les champs du formulaire

   setArticle({})

  };

  const serverOptions = () => { console.log('server pond');

    return {

      process: (fieldName, file, metadata, load, error, progress, abort) => {

          console.log(file)

        const data = new FormData();

        data.append('file', file);

        data.append('upload\_preset', 'Ecommerce\_cloudinary');

        data.append('cloud\_name', 'iset-sfax');

        data.append('public\_id', file.name);

        axios.post('https://api.cloudinary.com/v1\_1/iset-sfax/image/upload', data)

          .then((response) => response.data)

          .then((data) => {

            console.log(data);

           setArticle({...article,imageart:data.url}) ;

            load(data);

          })

          .catch((error) => {

            console.error('Error uploading file:', error);

            error('Upload failed');

            abort();

          });

      },

    };

  };

  return (

    <div className="form-container">

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

      <form  className="article-form">

      <Modal.Header closeButton>

 <h2>Ajouter Article</h2>

</Modal.Header>

<Modal.Body>

        <div className="form-grid">

          <div className="form-group">

            <label htmlFor="title">Référence</label>

            <input

              type="text"

              id="reference"

              value={article.reference}

              onChange={(e) => setArticle({...article,reference:e.target.value})}

              className="form-input"

              placeholder="Entrez référence article"

            />

          </div>

          <div className="form-group">

            <label htmlFor="description">Désignation</label>

            <input

            type="text"

              id="designation"

              value={article.designation}

              onChange={(e) => setArticle({...article,designation:e.target.value})}

              className="form-input"

              placeholder="Entrez la désignation article"

            />

          </div>

          <div className="form-group">

            <label htmlFor="marque">Marque</label>

            <input

              type="text"

              id="marque"

              value={article.marque}

              onChange={(e) => setArticle({...article,marque:e.target.value})}

              className="form-input"

              placeholder="Entrez marque"

            />

          </div>

          <div className="form-group">

            <label htmlFor="quantite">Quantité</label>

            <input

              type="number"

              id="qtestock"

              value={article.qtestock}

              onChange={(e) => setArticle({...article,qtestock:e.target.value})}

              className="form-input"

              placeholder="Entrez quantité stock"

            />

          </div>

          <div className="form-group">

            <label htmlFor="prix">Prix</label>

            <input

              type="number"

              required

              id="prix"

              value={article.prix}

              onChange={(e) => setArticle({...article,prix:e.target.value})}

              className="form-input"

              placeholder="Entrez Quantité stock"

            />

          </div>

          <div className="form-group">

            <label htmlFor="prix">Catégorie</label>

            <select

                id="category"

                className="form-control"

                value={article.scategorieID}

                onChange={(e) => setArticle({...article,scategorieID:e.target.value})}

              >

                {scategories.map((scat,index)=>

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

              )}

               </select>

          </div>

          <div className="form-group">

          <label htmlFor="prix">Image</label>

          <div style={{ width: "80%", margin: "auto", padding: "1%" }}>

     <FilePond

                   files={files}

                   acceptedFileTypes="image/\*"

                   onupdatefiles={setFiles}

                   allowMultiple={true}

                   server={serverOptions()}

                   name="file"

          />

    </div>

    </div>

        </div>

        </Modal.Body>

        <Modal.Footer>

        <button type="button" className="form-submit-button" onClick={(e)=>handleSubmit(e)}>Enregistrer</button>

        <button type="reset" className="form-reset-button" onClick={()=>handleClose()}>Annuler</button>

        </Modal.Footer>

      </form>

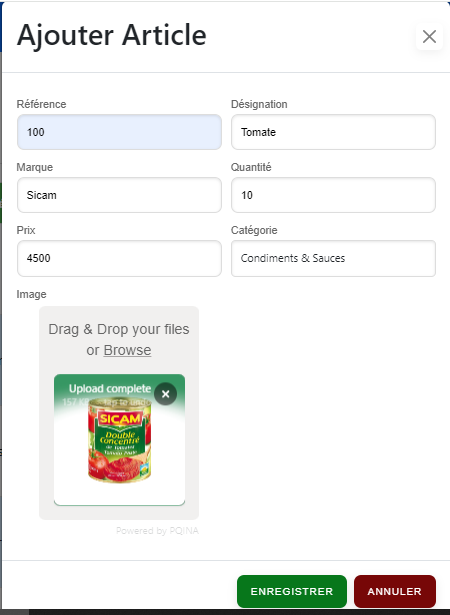
      </Modal>

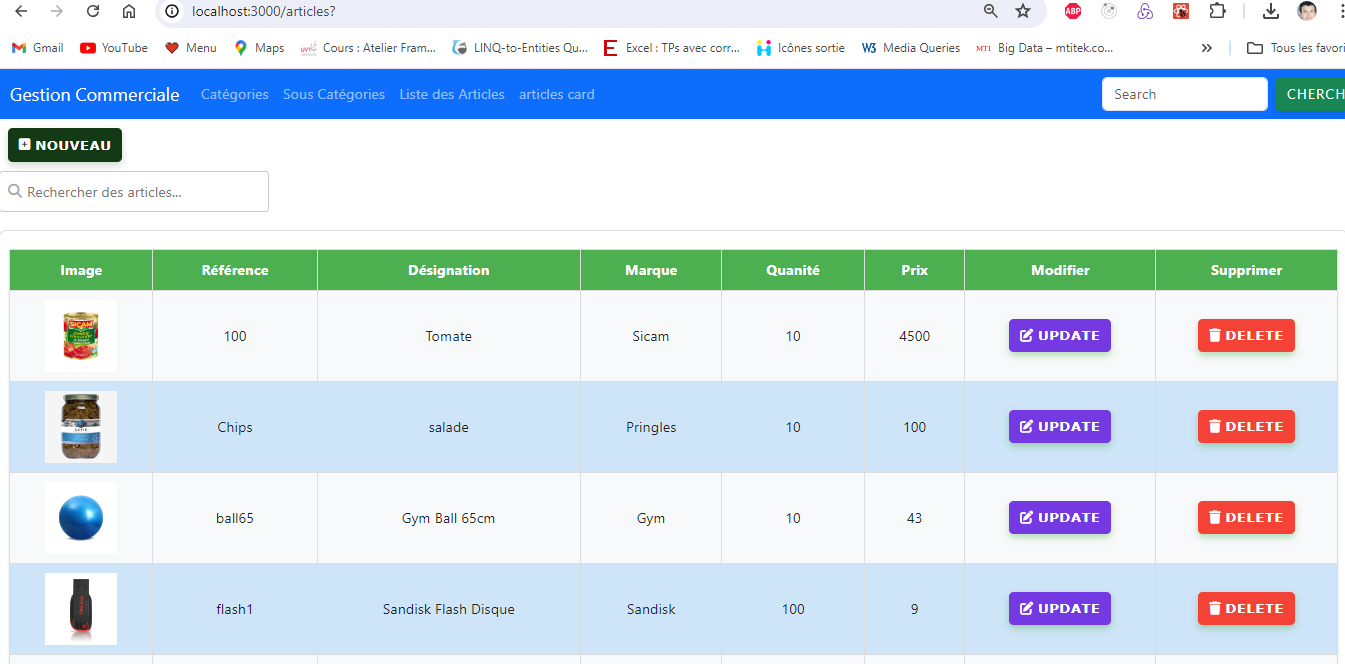
    </div>

  );

}

export default Insertarticle





Git : <https://github.com/hsan-hachicha-isetsf/ecommercehooks.git>

**Modification article**

**composant Affichearticle**

import React from 'react'

import "./article.css"

const Affichearticle = ({articles,handleLimitChange,limit,handleDeletearticle**,handleEditarticle**}) => {

  return (

    <div className="table-container">

    <table >

    <thead>

    <tr>

        <th >Image</th>

        <th>Référence</th>

        <th>Désignation</th>

        <th>Marque</th>

        <th>Quanité</th>

        <th>Prix</th>

        <th>Modifier</th>

        <th>Supprimer</th>

    </tr>

    </thead>

    <tbody>

        {

            articles.map((art,index)=>

            <tr key={index}>

                <td><img src ={art.imageart} width={80} height={80} /></td>

                <td>{art.reference}</td>

                <td>{art.designation}</td>

                <td>{art.marque}</td>

                <td>{art.qtestock}</td>

                <td>{art.prix}</td>

                <td><center><button className='edit' **onClick={() => handleEditarticle(art)}**>

                <i className="fa-solid fa-pen-to-square"></i>Update</button></center></td>

                <td><center><button className="delete" onClick={()=>handleDeletearticle(art.\_id,art.reference)} >

                <i className="fa-solid fa-trash"></i>

                 Delete</button></center></td>

            </tr>

            )}

    </tbody>

    <tfoot>

    <tr>

                <td colSpan="8">

                <div className="limit-selector-container">

                <label>

                    Afficher &nbsp;

                    <select

                      value={limit}

                      onChange={handleLimitChange}

                    >

                      <option value={5}>5</option>

                      <option value={10}>10</option>

                      <option value={20}>20</option>

                      <option value={100}>100</option>

                    </select>

                  </label>

                  </div>

                  </td>

            </tr>

            </tfoot>

    </table>

    </div>

  )

}

export default Affichearticle

**composant Listarticles**

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

import {fetcharticlesPagination,deletearticle} from "../../services/articleservice"

import "./article.css"

import Affichearticle from './Affichearticle';

import Pagination from './Pagination';

import Headerarticle from './Headerarticle';

import { confirmAlert } from 'react-confirm-alert'; // Import

import 'react-confirm-alert/src/react-confirm-alert.css'; // Import css

import Insertarticle from './Insertarticle';

import Editarticle from './Editarticle';

const Listarticles = () => {

  const [currentPage, setCurrentPage] = useState(1);

  const [totalPages, setTotalPages] = useState(0);

  const [articles, setArticles] = useState([]);

  const [limit, setLimit]=useState(5)

  const [searchText, setSearchText] = useState('');

  const [show, setShow] = useState(false);

  const [article, setArticle] = useState(null);

  const [showModalEdit, setShowModalEdit] = useState(false);

 const handleClose = () => {setShow(false); setShowModalEdit(false)}

  const handleShow = () => setShow(true);

  const fetchProducts = async (page,limit,searchText) => {

    try {

      const res = await fetcharticlesPagination(page,limit,searchText)

      setArticles(res.data.products);

      setTotalPages(res.data.totalPages);

    } catch (error) {

      console.log(error);

    }

  };

 useEffect(() => {

    fetchProducts(currentPage,limit,searchText);

  }, [currentPage,limit,searchText]);

  const handlePrevPage = () => {

    if (currentPage > 1) {

      setCurrentPage(currentPage - 1);

    }

  };

  const handleNextPage = () => {

    if (currentPage < totalPages) {

      setCurrentPage(currentPage + 1);

    }

  };

  const handlePageChange = (page) => {

    setCurrentPage(page);

  };

  const handleLimitChange = (event) => {

    setLimit(parseInt(event.target.value, 10));

    setCurrentPage(1);

  };

  const handleSearchChange = (event) => {

    setSearchText(event.target.value);

    setCurrentPage(1);

  };

  const handleDeletearticle = async (id,ref) => {

      confirmAlert({

        title: "Confirm delete...",

        message: " supprimer l' article: " + ref,

        buttons: [

        {

        label: 'Oui',

        onClick: () => deletearticle(id)

        .then(res=>fetchProducts(currentPage,limit,''))

        .catch(error=>console.log(error))

        },

        {

        label: 'Non',

        }

        ]

        });

      }

      const handleEditarticle = (item) => {

        setArticle(item);

        setShowModalEdit(true);

    };

  return (

    <div>

      <div className="table-container-header">

  <button className="new" onClick={handleShow}>

      <i className="fa-solid fa-plus-square"></i> Nouveau

  </button>

      <Headerarticle searchText={searchText}

      handleSearchChange={handleSearchChange}/>

      <Affichearticle

      articles={articles}

      handleLimitChange={handleLimitChange}

      limit={limit} handleDeletearticle={handleDeletearticle}

      handleEditarticle={handleEditarticle}

      />

      </div>

      <Pagination handlePrevPage={handlePrevPage}

      handleNextPage={handleNextPage}

      handlePageChange={handlePageChange}

      totalPages={totalPages}

      currentPage ={currentPage }

      />

{show && <Insertarticle

          show={show}

          handleClose={handleClose}

          fetchProducts={fetchProducts}

          limit={limit}

          />  }

{showModalEdit && (

                <Editarticle

                    show={showModalEdit}

                    handleClose={handleClose}

fetchProducts={fetchProducts}

                    art={article}

limit={limit}

                />

            )}

    </div>

  )

}

export default Listarticles

**composant Editarticle**

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

import "./insertarticle.css"

import axios from "axios"

import { Modal } from 'react-bootstrap';

import {fetchscategories} from "../../services/scategorieservice"

import {editarticle} from "../../services/articleservice"

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 Editarticle = ({show,handleClose,fetchProducts,art,limit}) => {

  const[article,setArticle]=useState([])

  const[scategories,setScategories]=useState([])

  const [category,setCategory]=useState(art.scategorieID.\_id)

  const [files, setFiles] = useState([]);

  const loadscategories=async()=>{

    try {

      const res = await fetchscategories()

      setScategories(res.data);

     console.log(res.data)

    } catch (error) {

      console.log(error);

    }

  }

  useEffect(() => {

    console.log(art.scategorieID.\_id)

    setArticle(art)

    loadscategories()

    setFiles( [

      {

        source: art.imageart,

        options: { type: 'local' }

      }

      ])

  }, [])

  const handleSubmit = async(event) => {

    event.preventDefault();

    // Logique pour soumettre le formulaire

   await editarticle(article)

   fetchProducts(1,limit,'')

   handleClose()

    // Réinitialiser les champs du formulaire

   setArticle({})

   setFiles([])

  };

  const serverOptions = () => { console.log('server pond');

    return {

      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: (fieldName, file, metadata, load, error, progress, abort) => {

          console.log(file)

        const data = new FormData();

        data.append('file', file);

        data.append('upload\_preset', 'Ecommerce\_cloudinary');

        data.append('cloud\_name', 'iset-sfax');

        data.append('public\_id', file.name);

        axios.post('https://api.cloudinary.com/v1\_1/iset-sfax/image/upload', data)

          .then((response) => response.data)

          .then((data) => {

            console.log(data);

           setArticle({...article,imageart:data.url}) ;

            load(data);

          })

          .catch((error) => {

            console.error('Error uploading file:', error);

            error('Upload failed');

            abort();

          });

      },

    };

  };

  return (

    <div className="form-container">

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

      <form  className="article-form">

      <Modal.Header closeButton>

 <h2>Modifier Article </h2>

</Modal.Header>

<Modal.Body>

        <div className="form-grid">

          <div className="form-group">

            <label htmlFor="title">Référence</label>

            <input

              type="text"

              id="reference"

              value={article.reference}

              onChange={(e) => setArticle({...article,reference:e.target.value})}

              className="form-input"

              placeholder="Entrez référence article"

            />

          </div>

          <div className="form-group">

            <label htmlFor="description">Désignation</label>

            <input

            type="text"

              id="designation"

              value={article.designation}

              onChange={(e) => setArticle({...article,designation:e.target.value})}

              className="form-input"

              placeholder="Entrez la désignation article"

            />

          </div>

          <div className="form-group">

            <label htmlFor="marque">Marque</label>

            <input

              type="text"

              id="marque"

              value={article.marque}

              onChange={(e) => setArticle({...article,marque:e.target.value})}

              className="form-input"

              placeholder="Entrez marque"

            />

          </div>

          <div className="form-group">

            <label htmlFor="quantite">Quantité</label>

            <input

              type="number"

              id="qtestock"

              value={article.qtestock}

              onChange={(e) => setArticle({...article,qtestock:e.target.value})}

              className="form-input"

              placeholder="Entrez quantité stock"

            />

          </div>

          <div className="form-group">

            <label htmlFor="prix">Prix</label>

            <input

              type="number"

              required

              id="prix"

              value={article.prix}

              onChange={(e) => setArticle({...article,prix:e.target.value})}

              className="form-input"

              placeholder="Entrez prix"

            />

          </div>

          <div className="form-group">

            <label htmlFor="prix">Catégorie</label>

            <select

                id="category"

                className="form-control"

                value={category}

                onChange={(e) =>{setCategory(e.target.value);setArticle({...article,scategorieID:e.target.value})}}

              >

                {scategories.map((scat,index)=>

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

              )}

               </select>

          </div>

          <div className="form-group">

          <label htmlFor="prix">Image</label>

          <div style={{ width: "80%", margin: "auto", padding: "1%" }}>

     <FilePond

                   files={files}

                   acceptedFileTypes="image/\*"

                   onupdatefiles={setFiles}

                   allowMultiple={true}

                   server={serverOptions()}

                   name="file"

          />

    </div>

    </div>

        </div>

        </Modal.Body>

        <Modal.Footer>

        <button type="button" className="form-submit-button" onClick={(e)=>handleSubmit(e)}>Enregistrer</button>

        <button type="reset" className="form-reset-button" onClick={()=>handleClose()}>Annuler</button>

        </Modal.Footer>

      </form>

      </Modal>

    </div>

  );

}

export default Editarticle

