

---

## Atelier CRUD REACT HOOKS Partie II

---

### Afficher les données dans MUI DataTable

#### ➤ Composant Listarticles.js

Accéder au site

<https://github.com/gregnb/mui-datatables>

On a déjà installer material ui:

npm install @mui/material @emotion/react @emotion/styled

on va maintenant installer MUI-DATATABLE

pour choisir une icône : <https://mui.com/components/material-icons/>

- Installer muidatatable

```
npm install mui-datatables
```

```
npm i @mui/styles
```

Dans index.js ajouter cet import car Material UI se base sur cette police roboto.

```
npm i typeface-roboto
```

```
import 'typeface-roboto';
```

```
import {ArticleService} from '../Services/Article-Service';
import { useState,useEffect } from 'react';
//npm install @mui/material @emotion/react @emotion/styled

//npm install mui-datatables
import MUIDataTable from "mui-datatables";
//npm i @mui/styles
import { ThemeProvider } from "@mui/styles";
import { createTheme } from "@mui/material/styles";

import { IconButton,Button } from '@mui/material';
import AddCircleIcon from '@mui/icons-material/AddCircle';
import EditIcon from '@mui/icons-material/Edit';
```

```

import DeleteIcon from '@mui/icons-material/Delete';
import { pink } from '@mui/material/colors';
import { Link } from "react-router-dom";

const ListArticlesDatatable=()=>{

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

  useEffect(() => {
    GetListArticles();
    console.log(articles)

  },[]);

  const GetListArticles=async()=>{

    await ArticleService.fetchArticles()
      .then((res) => {
        setArticles(res.data);

      });
  }
  const delArticle= async (_id) => {

    await ArticleService.deleteArticle(_id)

    var newarticles=articles.filter((item)=>{
      console.log(item)
      return item._id!==_id
    })
    setArticles(newarticles);
  }

  const columns = [
    {
      label: "Référence",
      name: "reference"
    },
    {
      label: "Désignation",
      name: "designation"
    },
    {
      label: "Prix Achat",
      name: "prixAchat"
    },
    {
      label: "Prix Vente",

```

```

    name: "prixVente"
  },

  {
    label: "Marque",
    name: "marque"
  },

  {
    label: "Quatité Stock",
    name: "qtestock"
  },

  {
    name: "imageartpetitf",
    label: "Image",
    options: {
      customBodyRender : (rowData) => (
        <img
          style={{ height: 60, borderRadius: '50%' }}
          src={rowData}
          alt=""
        />
      )
    }
  },

  {
    name: "categorieID",
    label: "Catégorie",
    options: {
      customBodyRender : (categ) => (
        categ? categ.nomcategorie : null
      )
    }
  },

  {
    name: "scategorieID",
    label: "S/Catégorie",
    options: {
      customBodyRender : (scateg) => (
        scateg? scateg.nomscategorie : null
      )
    }
  },

  {
    name: "_id",
    label: "Actions",
    options: {
      customBodyRender: (value) => (
        <div>

```

```

        <IconButton >
        { <Link to={"/Article/edit/" + value} >
          <EditIcon color='secondary' />
        </Link>
        }

        </IconButton>
        <IconButton onClick={()=>{delArticle(value)}}>
        <DeleteIcon sx={{ color: pink[500] }} />

        </IconButton>

      </div>
    )
  }
},
];

return(
  <div>
    <div style={{padding:5,margin:5}}>
      <Button
        color="success"

        startIcon={<AddCircleIcon />}
        variant="contained"

        >
          { <Link to={"/Articles/add"} style={{textDecoration:
"none",color:"white"}}>
            Ajouter
          </Link>
        }

      </Button>
    </div>
    {articles.length>0?

      <ThemeProvider theme={createTheme()}>
        <MUIDataTable
          title="Liste des articles"
          data={articles}

```

```

        columns={columns}

      />
    </ThemeProvider>
  :null
}

</div>
)
}
export default ListArticlesDatatable

```

### Composant Insertarticle

1. Dans le dossier « Components/Articles », créer le composant intitulé « Insertarticle.js », il contiendra un formulaire d'ajout de données. On va utiliser textfield de Material pour les champs. Ainsi que la bibliothèque Filepond pour le téléchargement des images.
2. Installer la bibliothèque react-toastify(pour affichage) et Filepond (pour les images)
  - a. Installer « react-toastify ». Il permet d'ajouter facilement des notifications à notre application.

Pour plus d'informations visiter : <https://www.npmjs.com/package/react-toastify>

```
npm i react-toastify
```

- b. Installer « FilePond ». React FilePond est un composant wrapper pratique pour FilePond, 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 formidable, accessible et fluide.

Pour plus d'informations visiter : <https://codesandbox.io/s/rylk9z5lzm?file=/src/index.js:954-962>

```

npm i filepond
npm i react-filepond
npm i filepond-plugin-image-preview
npm i filepond-plugin-image-exif-orientation

```

```

import React, { useState,useEffect } from "react";
import Axios from "../../Axios/Api";
import { useNavigate } from "react-router-dom";
import { Button,TextField,FormControl,MenuItem} from '@mui/material';
import { makeStyles } from '@material-ui/core/styles';
import { ToastContainer, toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
import { FilePond,registerPlugin } from 'react-filepond'
import SaveIcon from '@mui/icons-material/Save';
import CancelIcon from '@mui/icons-material/Cancel';
import { Link } from "react-router-dom";
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 useStyles = makeStyles((theme) => ({
  formControl: {
    margin: theme.spacing(1),

    minWidth: 120,

  }
})));

const Insertarticle =()=> {

  let navigate=useNavigate();

  const classes = useStyles();

  const [reference, setReference] = useState("");
  const [designation, setDesignation] = useState("");
  const [categorieID, setCatID] = useState("");
  const [categories, setCategories] = useState([]);

const [prixAchat, setPrixAchat] = useState("");
  const [prixVente, setPrixVente] = useState("");
  const [prixSolde, setPrixSolde] = useState("");
  const [marque, setMarque] = useState("");
  const [scategories, setscategories] = useState([]);
  const [scategorieID, setSCatID] = useState("");
  const [qtestock, setQtestock] = useState("");
  const [caracteristiques, setCaracteristiques] = useState("");
  const [files, setFiles] = useState("")
  const [filesm, setFilesm] = useState([])

```

```

useEffect(() => {

    GetListCategories();

},[]);

const GetListCategories=async()=>{

    await Axios.get("/categories")
        .then((res) => {
            setCategories(res.data);
        });

}

const GetListSCategories=async(idcat)=>{

    await Axios.get("/scategories/cat/"+idcat)
        .then((res) => {
            setscategories(res.data);
        });

}

const handleSubmit = async (event) => {
    event.preventDefault();
    var ig=[];
    filesm.forEach(element => {
ig.push("images/"+element.file.name);

    });

const objetarticle = {
    reference: reference,
    designation :designation,
    prixAchat :prixAchat,
    prixVente :prixVente,
    prixSolde :prixSolde,
    marque :marque,
    qtestock:qtestock,
    caracteristiques:caracteristiques,
    imageartpetitf : files?"images/"+files[0].file.name:null,
    imageartgrandf:ig,
    categorieID :categorieID,
    scategorieID:scategorieID
    };

    await Axios.post("/articles", objetarticle).then((res)=>{

```

```

        toast("Article ajouté", {
          position: "top-right",
          autoClose: 5000,
          hideProgressBar: false,
          closeOnClick: true,
          pauseOnHover: true,
          draggable: true,
          progress: undefined,
        });
        navigate("/ArticlesTable")
      }).catch(error => {
        toast("Erreur Article non ajouté", {
          position: "top-right",
          autoClose: 5000,
          hideProgressBar: false,
          closeOnClick: true,
          pauseOnHover: true,
          draggable: true,
          progress: undefined,
        });
      });
    });
  }

  return (
    <>
      <ToastContainer />
      <div className="container">

        <form >
          <div>

            <Button style={{padding:15,margin:20,width:150}}
              color="secondary"

              startIcon={<SaveIcon />}
              variant="contained"
              onClick={(event)=>handleSubmit(event)}

            >

              Save
            </Button>

            <Button style={{padding:15,margin:2,width:150}}
              color="primary"

```



```

startIcon={<CancelIcon />}
variant="contained"

>
  { <Link to={"/ArticlesTable"} style={{textDecoration:
"none",color:"white"}}>
    Annuler
  </Link>
}

</Button>

</div>

<FormControl className={classes.formControl} >

  <TextField

    variant="outlined"
    style={{ marginLeft: 8,marginTop:20,width:400}}
    label="Désignation"
    value={designation}
    onChange={e => setDesignation(e.target.value)}
    required />

  </FormControl>

  <FormControl className={classes.formControl} >
  <TextField
    variant="outlined"
    style={{ marginLeft: 20,marginTop:20,width:400}}
    label="Référence"
    value={reference}
    onChange={e => setReference(e.target.value)}
    required />

  </FormControl> <br/>

  <FormControl className={classes.formControl}>
  <TextField
    variant="outlined"
    style={{ marginLeft: 8,marginTop:20,width:400}}
    label="Prix Achat"

```

```

        type="number"
        value={prixAchat}
        onChange={e => setPrixAchat(e.target.value)}
    />
</FormControl>

<FormControl className={classes.formControl}>
<TextField
    variant="outlined"
    style={{ marginLeft: 20,marginTop:20,width:400}}
    label="Prix Vente"
    type="number"
    value={prixVente}
    onChange={e => setPrixVente(e.target.value)}
    />
</FormControl> <br/>

<FormControl className={classes.formControl}>
<TextField
    variant="outlined"
    style={{ marginLeft: 8,marginTop:20,width:400}}
    label="Prix Solde"
    type="number"
    value={prixSolde}
    onChange={e => setPrixSolde(e.target.value)}
    />
</FormControl>

<FormControl className={classes.formControl}>
<TextField
    variant="outlined"
    style={{ marginLeft: 20,marginTop:20,width:400}}
    label="Quantité Stock"
    type="number"
    value={qtestock}
    onChange={e => setQtestock(e.target.value)}
    />
</FormControl> <br/>

<FormControl className={classes.formControl}>
<TextField
    fullWidth
    style={{ marginLeft: 8,marginTop:20,width:400}}
    variant="outlined"
    label="Marque"
    value={marque}
    onChange={e => setMarque(e.target.value)}
    />
</FormControl>

```

```

<FormControl className={classes.formControl}>
  <TextField
    fullWidth
    style={{ marginLeft: 20,marginTop:20,width:400}}
    variant="outlined"
    label="Caractéristiques"
    value={caracteristiques}
    onChange={e => setCaracteristiques(e.target.value)}
  />
</FormControl> <br/>
<FormControl style={{width:350}}>
  <TextField
    fullWidth
    select

    label="Catégories"
    variant="outlined"

    value={categorieID}
    style={{ marginLeft: 8,marginTop:20,width:820}}
    onChange={(event)=>{setCatID(event.target.value); GetListSCateg
ories(event.target.value) }}
    helperText="Sélectionner une catégorie"

    >
    {
      categories ?
      categories.map(cat=>
        <MenuItem key={cat._id}
value={cat._id}>{cat.nomcategorie}
        </MenuItem>
      )
      :null
    }
  </TextField>
</FormControl><br/>
<FormControl style={{width:350}}>
  <TextField
    select
    label="S/Catégorie"
    style={{ marginLeft: 8,marginTop:10,width:820}}
    value={scategorieID}
    variant="outlined"
    helperText="Sélectionner une s/catégorie"
    onChange={e => setSCatID(e.target.value)}
    >
    {
      scategories ?
      scategories.map((scat)=>

```

```

        <MenuItem key={scat._id}
value={scat._id}>{scat.nomscategorie}</MenuItem>
      )
      :null
    }
  </TextField>
</FormControl>

  <h4>Sélectionner une image</h4>
  <FormControl className={classes.formControl}>
    <div style={{width:400, height:50}}>
    <FilePond
files={files}
allowMultiple={false}
onupdatefiles={setFiles}
labelIdle='<span class="filepond--label-action">Browse One</span>'
  />
    </div>
  </FormControl>

  <FormControl className={classes.formControl}>
    <div style={{width:400, height:40}}>
    <FilePond
files={filesm}

allowMultiple={true}
onupdatefiles={setFilesm}
labelIdle='<span class="filepond--label-action">Browse Many</span>'
  />

    </div>
  </FormControl>
</form>

</div>
</>

);
}
export default Insertarticle;

```

Pour que react-toastify fonctionne correctement (après redirection) il faut modifier le fichier App.js comme suit :

```
import Navbarre from './Components/Navbarre';
import 'bootstrap/dist/css/bootstrap.min.css';
import ListRoutes from './Routes/ListRoutes';
import {BrowserRouter as Router} from 'react-router-dom';
import { ToastContainer} from 'react-toastify';
function App() {
  return (
    <>
      <Router>

      <Navbarre/>

      <ListRoutes/>
      <ToastContainer />
    </Router>
    </>
  );
}

export default App;
```

1. Dans le dossier « Components/Articles », créer le composant intitulé « Editarticle.js », qui permettra de mettre à jour une ligne sélectionnée.

```
import React, { useState,useEffect } from "react";
import Axios from "../../Axios/Api"
import { useNavigate, useParams } from "react-router-dom";
import { ArticleService } from "../../Services/Article-Service";
import SaveIcon from '@mui/icons-material/Save';
import CancelIcon from '@mui/icons-material/Cancel';
import { Link } from "react-router-dom";
import { Button } from '@mui/material';
import MenuItem from '@material-ui/core/MenuItem';
import FormControl from '@material-ui/core/FormControl';
import TextField from '@material-ui/core/TextField';
import { toast } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
// Import React FilePond
import { FilePond, File,registerPlugin } from 'react-filepond'
// Import FilePond styles
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 ModifArticle =()=> {

  let navigate=useNavigate();
  const {id} = useParams();
  const [reference, setReference] = useState("");
  const [designation, setDesignation] = useState("");
  const [categories, setCategories] = useState("");
  const [categorieID, setcategorieID] = useState("");
  const [scategories, setScategories] = useState("");
  const [scategorieID, setscategorieID] = useState("");
  const [prixAchat, setPrixAchat] = useState("");
  const [prixVente, setPrixVente] = useState("");
  const [prixSolde, setPrixSolde] = useState("");
  const [marque, setMarque] = useState("");
  const [qtestock, setQtestock] = useState("");
  const [caracteristiques, setCaracteristiques] = useState("");
  const [files, setFiles] = useState([])
  const [filesm, setFilesm] = useState([])

  useEffect(() => {
    GetListCategories();
    ArticleService.fetchArticleById(id)
      .then(res => {
        setReference(res.data.reference);
        setDesignation(res.data.designation);
        setscategorieID(res.data.scategorieID)
        setcategorieID(res.data.categorieID);
        setPrixAchat(res.data.prixAchat);
        setPrixVente(res.data.prixVente);
        setPrixSolde(res.data.prixSolde);
        setMarque(res.data.marque);
        setQtestock(res.data.qtestock);
        setCaracteristiques(res.data.caracteristiques);

        setFiles("/"+ res.data.imageartpetitf);
        setFilesm(res.data.imageartgrandf);

      })
      .catch((error) => {
        console.log(error);
        toast("Une erreur est parvenue", {
          position: "top-right",
          autoClose: 5000,
          hideProgressBar: false,
          closeOnClick: true,
          pauseOnHover: true,

```

```

        draggable: true,
        progress: undefined,
    });
})
},[]);

const GetListCategories=async()=>{

    await Axios.get("/categories")
        .then((res) => {
            setCategories(res.data);
        });
}

const GetListSCategories=async(idcat)=>{

    await Axios.get('/scategories/cat/' + idcat)
        .then((res) => {
            setScategories(res.data);
        });
}

const handleSubmit = async (event) => {

    event.preventDefault();

    var ig=[];

    filesm.forEach(element => {
        ig.push("/images/"+element.file.name);
    });

    const objetarticle = {
        _id : id,
        reference: reference,
        designation :designation,
        prixAchat :prixAchat,
        prixVente :prixVente,
        prixSolde :prixSolde,
        marque :marque,
        qtstock :qtstock,
        caracteristiques:caracteristiques,
        categorieID:categorieID,
        imageartpetitf : files?"images/"+files[0].file.name:null,
        imageartgrandf:ig,
        scategorieID :scategorieID
    };

```

```

        console.log(objetarticle)

        await ArticleService.editArticle(objetarticle).then((res)=>{

            toast("Article modifié", {
                position: "top-right",
                autoClose: 5000,
                hideProgressBar: false,
                closeOnClick: true,
                pauseOnHover: true,
                draggable: true,
                progress: undefined,
            });

            navigate("/ArticlesTable")
        }).catch(error => {
            toast("Erreur Article non modifié", {
                position: "top-right",
                autoClose: 5000,
                hideProgressBar: false,
                closeOnClick: true,
                pauseOnHover: true,
                draggable: true,
                progress: undefined,
            });
        });
    }

    return (
        <>

        <form onSubmit={handleSubmit} >
            <h2>Edit Article </h2>
            <div>

                <Button style={{padding:15,margin:20,width:150}}
                    color="secondary"
                    startIcon={<SaveIcon />}
                    variant="contained"
                    onClick={(event)=>handleSubmit(event)}

                >

                Update
            </Button>

```



```

<Button style={{padding:15,margin:2,width:150}}
  color="primary"

  startIcon={<CancelIcon />}
  variant="contained"

  >
    { <Link to={"/ArticlesTable"} style={{textDecoration:
"none",color:"white"}}>
      Annuler
    </Link>
    }

</Button>
</div>
<FormControl >
  <TextField
    variant="outlined"
    label="Désignation"
    style={{ marginLeft: 8,marginTop:20,width:400}}
    value={designation}
    onChange={e => setDesignation(e.target.value)}
    required />
  </FormControl>
  <FormControl >
    <TextField
      variant="outlined"
      label="Référence"
      value={reference}
      style={{ marginLeft: 8,marginTop:20,width:400}}
      onChange={e => setReference(e.target.value)}
      required />
    </FormControl> <br/>

    <FormControl >
      <TextField
        variant="outlined"
        label="Prix Achat"
        type="number"
        style={{ marginLeft: 8,marginTop:20,width:400}}
        value={prixAchat}
        onChange={e => setPrixAchat(e.target.value)}
        />
      </FormControl>

    <FormControl >

```

```

<TextField
  variant="outlined"
  label="Prix Vente"
  type="number"
  style={{ marginLeft: 8,marginTop:20,width:400}}
  value={prixVente}
  onChange={e => setPrixVente(e.target.value)}
/>
</FormControl> <br/>

<FormControl >
<TextField
  variant="outlined"
  label="Prix Solde"
  type="number"
  style={{ marginLeft: 8,marginTop:20,width:400}}
  value={prixSolde}
  onChange={e => setPrixSolde(e.target.value)}
/>
</FormControl>

<FormControl >

<TextField
  variant="outlined"
  label="Quantité Stock"
  type="number"
  style={{ marginLeft: 8,marginTop:20,width:400}}
  value={qtestock}
  onChange={e => setQtestock(e.target.value)}
/>
</FormControl><br/>
<FormControl>

<TextField
  style={{ marginLeft: 8,marginTop:20,width:400}}
  variant="outlined"
  label="Marque"
  value={marque}
  onChange={e => setMarque(e.target.value)}
/>
</FormControl>
<FormControl >
<TextField
  style={{ marginLeft: 8,marginTop:20,width:400}}
  margin="normal"
  variant="outlined"
  multiline
  rows={2}

```

```

        label="Caractéristiques"
        type="textarea"
        value={caracteristiques}
        onChange={e => setCaracteristiques(e.target.value)}
      />
    </FormControl> <br/>
    <FormControl >
      <TextField
        select
        label="Categories"
        variant="outlined"
        value={categorieID}
        style={{ marginLeft: 8,marginTop:20,width:800}}
        onChange={e=>{setcategorieID(e.target.value);
GetListSCategories(e.target.value)} }
      >
      {
        categories ?
        categories.map(f=>
          <MenuItem value={f._id}>{f.nomcategorie}

          </MenuItem>

        )
        :null
      }
    </TextField>
    </FormControl><br/>
    <FormControl >
      <TextField
        select
        label="Sous Catégorie"
        variant="outlined"
        value={scategorieID}
        style={{ marginLeft: 8,marginTop:20,width:800}}
        onChange={e => setscategorieID(e.target.value)}
      >
      {
        scategories ?
        scategories.map(f=>
          <MenuItem value={f._id}>{f.nomscategorie}

          </MenuItem>

        )
        :<MenuItem value={scategorieID._id}>{scategorieID.nomscategorie}
    </MenuItem>

      }
    </TextField>

```

```

        </FormControl>
    </form>
    <br/>
    <h4>Upload Images</h4>
    <FormControl >
        <div style={{width:400, height:50}}>
        <FilePond
            files={files}
            allowMultiple={false}
            onupdatefiles={setFiles}
            labelIdle='<span class="filepond--label-action">Browse One</span>'

        />
        </div>
    </FormControl>
    <FormControl>
        <div style={{width:400, height:40}}>
        <FilePond
            files={filesm}

            allowMultiple={true}
            onupdatefiles={setFilesm}
            labelIdle='<span class="filepond--label-action">Browse Many</span>'
        />
        </div>
    </FormControl>

    </>

    );
}

export default ModifArticle;

```

2. Créer un composant « Routes/Routes.js » qui contiendra les chemins de notre application.

```

import { Routes, Route } from "react-router-dom";
import ListArticles from "../Components/Articles/ListArticles"
import AjoutArticle from "../Components/Articles/AjoutArticle";

```

```

import ModifArticle from "../Components/Articles/ModifArticle";

function LesRoutes() {
  return (

    <Routes>
    <Route path="/Articles" exact element={<ListArticles/>}/>
    <Route path="/Articles/add" element={<AjoutArticle/>}/>
    <Route path="/Article/edit/:id" element={<ModifArticle/>}/>
    </Routes>

  );
}

export default LesRoutes;

```

3. Créer le composant « Routes/ListRoutes.js » contenant la route de chaque composant.

```

import { Routes, Route } from "react-router-dom";
import Editarticle from "../Components/Articles/Editarticle";
import Insertarticle from "../Components/Articles/Insertarticle";
import ListArticleCard from "../Components/Articles/ListArticleCard";
import Listarticles from "../Components/Articles/Listarticles";
import ListArticlesDatatable from "../Components/Articles/ListArticlesDatatable";
import Editcategorie from "../Components/Categories/Editcategorie";
import Insertcategorie from "../Components/Categories/Insertcategorie";
import Listcategories from "../Components/Categories/Listcategories";
import EditScategorie from "../Components/SouCategories/EditScategorie";
import InsertScategorie from "../Components/SouCategories/InsertScategorie";
import ListScategorie from "../Components/SouCategories/ListScategorie";
const ListRoutes=() =>{
  return (
    <Routes>
    <Route path="/Articles" exact element={<Listarticles/>}/>
    <Route path="/ArticlesCard" exact element={<ListArticleCard/>}/>
    <Route path="/ArticlesTable" exact element={<ListArticlesDatatable/>}/>
    <Route path="/Articles/add" element={<Insertarticle/>}/>
    <Route path="/Article/edit/:id" element={<Editarticle/>}/>
    <Route path="/Categories" exact element={<Listcategories/>}/>
    <Route path="/Categories/add" element={<Insertcategorie/>}/>
    <Route path="/Categories/edit/:id" element={<Editcategorie/>}/>
    <Route path="/Scategories" exact element={<ListScategorie/>}/>
    <Route path="/Scategories/add" element={<InsertScategorie/>}/>
    <Route path="/Scategories/edit/:id" element={<EditScategorie/>}/>
    </Routes>

  );
}

```

```
export default ListRoutes;
```

#### 4. Créer le fichier Components/Navbarre.js

```
import React from 'react'

import {Nav, Navbar, Container, Form, FormControl, Button} from 'react-bootstrap';
import {Link} from 'react-router-dom'
const Navbarre=()=>{

  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.Link as={Link} to="/ArticlesCard">Articles Card</Nav.Link>
          <Nav.Link as={Link} to="/ArticlesTable">ArticlesTable</Nav.Link>

        </Nav>
        <Form className="d-flex">
          <FormControl
            type="search"
            placeholder="Search"
            className="me-2"
            aria-label="Search"
          />
          <Button variant="success">Chercher</Button>
        </Form>
      </Container>

    </Navbar>
  )
}
export default Navbarre
```

#### 5. Modifier « App.js » qui appelle « Routes » et « NavBar » de la façon suivante :

```
import Navbarre from './Components/Navbarre';
import 'bootstrap/dist/css/bootstrap.min.css';
import ListRoutes from './Routes/ListRoutes';
```

```
import {BrowserRouter as Router} from 'react-router-dom';

function App() {
  return (
    <>
      <Router>

      <Navbarre/>

      <ListRoutes/>

    </Router>
  </>
);
}

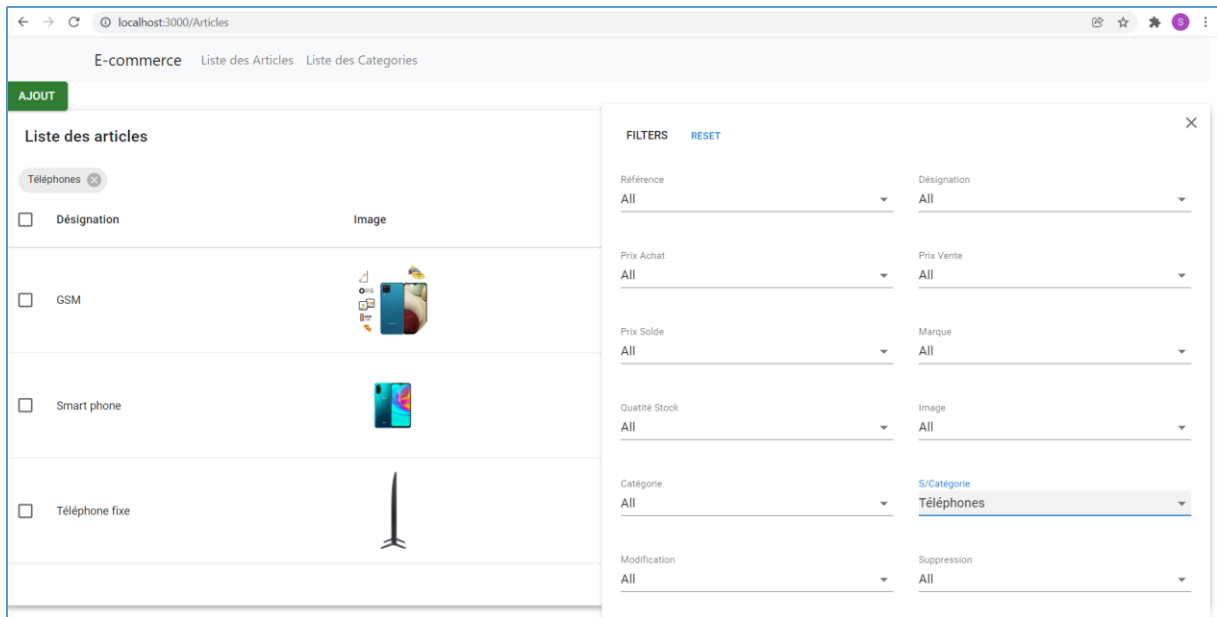
export default App;
```

## Résultat final

1. La liste :
2. Opérations sur les filtres
  - a. Colonnes

<input type="checkbox"/>	RÉFÉRENCE	DÉSIGNATION	PRIX ACHAT	PRIX VENTE	MARQUE	QUATITÉ STOCK	IMAGE	CATÉGORIE	S/CATÉGORIE	ACTIONS
<input type="checkbox"/>	Chips	Pringles Chips Pringles ORG 40gr	2.8	3.37	Pringles	100		Superette	Aliments	
<input type="checkbox"/>	Huilejs	Al Jazira Huile de grignons d'olive	20.3	23.5	Al Jazira	120		Superette	Aliments	
<input type="checkbox"/>	lait	Delice Lait UHT demi-écrémé - 1L	1.2	1.35	Delice	1000		Superette	Aliments	
<input type="checkbox"/>	eauxmin	Delice EAU MINÉRALE 1.5 L	0.39	0.58	Delice	1000		Superette	Aliments	
<input type="checkbox"/>	farsan	Sanabel Carthage Farine Paquet De 1Kg	0.7	0.85	Sanabel Carthage	220		Superette	Aliments	
<input type="checkbox"/>	delpush	Delice Punch Orange 1,5L	1.6	2.3	Delice	85		Superette	Aliments	


- b. Lignes



### 3. Ajout d'un article

The screenshot shows the 'Articles/add' page. It features a form with the following fields: Désignation (Conserve Tomates), Référence (TS123), Prix Achat (2), Prix Vente (3), Prix Solde (2.4), Quantité Stock (20), Marque (SICAM), Caractéristiques (Tomates fraîches), and Catégories (Superette). Below the form, there's a section for 'Upload Images' showing a preview of a tomato paste can and a 'Browse Many' button. The page also has a green 'AJOUT' button at the top left.

L'article est ajouté dans la liste :

<input type="checkbox"/>	TS123	Conserve Tomates	2	3	2.4	SICAM	20		Superette	Aliments	<a href="#">MODIFIER</a>	<a href="#">SUPPRIMER</a>
--------------------------	-------	------------------	---	---	-----	-------	----	---	-----------	----------	--------------------------	---------------------------

Rows per page: 10 1-7 of 7

### 4. Modifier un article



← → ↻ localhost:3000/Article/edit/61c5db4f832557168e4a29c

E-commerce Liste des Articles Liste des Categories

## Edit Article Edit

### Données Article

Designation*	Reference*	Prix Achat	Prix Vente	Prix Solde	Quantité Stock
Tomates	TS123	2.5	3.5	2,4	20

Marque  
SICAM

Caractéristiques

Tomates fraîches

Categories  
Superette

Sous Catégorie

Sélectionner une catégorie

### Upload Images


scamtomate.jpg 1 KB

Browse Many

scamtomate1.jpg 56 KB

Powered by PONA

## Article modifié

<input type="checkbox"/>	TS123	Tomates	2.5	3.5	2.4	SICAM	20		Superette	Aliments	<span>MODIFIER</span>	<span>SUPPRIMER</span>
--------------------------	-------	---------	-----	-----	-----	-------	----	---	-----------	----------	-----------------------	------------------------

Rows per page: 10 1-7 of 7 < >

## 5. En cliquant sur « supprimer » l'article est supprimé dans la base de données et dans la liste







← → ↻ localhost:3000/ArticlesTable

Gestion Commerciale Catégories Sous Catégories Liste des Articles Articles Card ArticlesTable

Search Chercher

AJOUTER

### Liste des articles

	RÉFÉRENCE	DÉSIGNATION	PRIX ACHAT	PRIX VENTE	MARQUE	QUATITÉ STOCK	IMAGE	CATÉGORIE	S/CATÉGORIE	ACTIONS
<input type="checkbox"/>	Chips	Pringles Chips Pringles ORG 40gr	2.8	3.37	Pringles	100		Superette	Aliments	<span></span> <span></span>
<input type="checkbox"/>	Huilejs	Al Jazira Huile de grignons d'olive	20.3	23.5	Al Jazira	120		Superette	Aliments	<span></span> <span></span>
<input type="checkbox"/>	lait	Delice Lait UHT demi-écrémé - 1L	1.2	1.35	Delice	1000		Superette	Aliments	<span></span> <span></span>
<input type="checkbox"/>	eauxmin	Delice EAU MINÉRALE 1.5 L	0.39	0.58	Delice	1000		Superette	Aliments	<span></span> <span></span>
<input type="checkbox"/>	farsan	Sanabel Carthage Farine Paquet De 1Kg	0.7	0.85	Sanabel Carthage	220		Superette	Aliments	<span></span> <span></span>
<input type="checkbox"/>	delpush	Delice Punch Orange 1,5L	1.6	2.3	Delice	85		Superette	Aliments	<span></span> <span></span>