

---

## Création d'un panier avec ReactJS

---

Créer le composant ListArticleCard

```
import {useState,useEffect} from 'react'
import Card from '@mui/material/Card';
import CardActions from '@mui/material/CardActions';
import CardContent from '@mui/material/CardContent';
import CardMedia from '@mui/material/CardMedia';
import Button from '@mui/material/Button';
import Typography from '@mui/material/Typography';
import Axios from "../../Axios/Api"
const ListArticleCard={()=>{

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

  useEffect(() => {
    GetListArticles();

  });
  const GetListArticles=async()=>{

    await Axios.get("/articles")
      .then((res) => {
        setArticles(res.data);

      });
  }
  return(
    <div className="container">
      <div
style={{display:"flex","flexWrap":"wrap","justifyContent":"center"}}>
        {articles.map((art,ind)=>{
          return <Card sx={{ maxWidth: 'auto',margin: 1 }}>
            <CardMedia
              component="img"
              alt="green iguana"
              height="160"
              image={art.imageartpetitf}
            />
            <CardContent>
              <Typography gutterBottom variant="h5" component="div">
                {art.reference}
              </Typography>
              <Typography variant="body2" color="text.secondary">
                Prix : {art.prixVente} DT
              </Typography>
            </CardContent>
            <CardActions>
```

```

        <Button disabled={art.qtestock<=1}
              variant="contained" color="secondary" size="large"

              >
                {art.qtestock<=1? "OUT OF SOLD": "Add to cart"}
            </Button>

        </CardActions>
    </Card>
  )}</div>
</div>
)

}
export default ListArticleCard

```

Accéder maintenant au site <https://www.npmjs.com/package/react-use-cart>

```
npm i react-use-cart
```

Ajouter au fichier ListArticleCard le code suivant pour le bouton

```

    <Button disabled={art.qtestock<=1}

            variant="contained" color="secondary" size="large"
            onClick={() =>{
              const target = { price: art.prixVente,id:art._id};
              const returnedTarget = Object.assign(target, art);
              addItem(returnedTarget)}
            }

    >

```

Ajouter ensuite en haut les deux lignes suivantes :

```

import { useCart } from "react-use-cart";

const { addItem, totalItems } = useCart();

```

on va créer maintenant un composant intitulé `CartArticle.js`

```
import { useCart } from "react-use-cart";

import Button from '@mui/material/Button';
const CartArticle={()=>{
const {
  isEmpty,
  totalItems,
  cartTotal,
  items,
  updateItemQuantity,
  removeItem,
} = useCart();

if (isEmpty) return <p>Votre panier est vide</p>;

return (
  <>

    <ul>
      {items.map((item) => (
        <div className="row no-gutters py-2">
          <div className="col-sm-2 p-2">
            <img
              alt={item.designation}
              style={{margin: "0 auto", maxHeight: "50px"}}
              src={item.imageartpetitf} className="img-fluid d-block"/>
          </div>
          <div className="col-sm-4 p-2">
            <h5 className="mb-1">{item.designation}</h5>
            <p className="mb-1">Prix: {item.price} </p>

          </div>
          <div className="col-sm-2 p-2 text-center ">
            <p className="mb-0">Quantité: {item.quantity}</p>
          </div>
          <div className="col-sm-4 p-2 text-right">
            <Button variant="contained" color="success" onClick={() => {
              if(item.quantity < item.qtestock )
                updateItemQuantity(item.id, item.quantity + 1);
              else
                alert("Quantité stock indisponible")
            }}>
            +
          </Button>
        </div>
      )})}
    </ul>
  </>
)
```



```

        <h3 className="m-0 txt-right">{cartTotal}
TND</h3>

        <hr className="my-4"/>
        <div className="text-center">
            <Button
variant="contained">CHECKOUT</Button><br/><br/>
            <Button variant="contained"
color="success">VALIDER</Button>
        </div>

    </div>

</div>

</div>

</div>
</div>

</>
);
}
export default CartArticle

```

Remarque : Pour le total des prix on peut utiliser la fonction toFixed pour fixer 3 chiffres après la virgule.

```
{cartTotal.toFixed(3)}
```

## Fichier App.js

```
import LesRoutes from "./Routes/Routes";
import { CartProvider, useCart } from "react-use-cart";
import { BrowserRouter as Router } from "react-router-dom";
import 'bootstrap/dist/css/bootstrap.min.css';
import Navbarre from "../Components/Navbarre"

function App() {
  return (
    <>
    <CartProvider>
    <Router>
    <Navbarre/>
    <LesRoutes/>
  )
}
```

```

    </Router>
  </CartProvider>
  </>
);
}

export default App;

```

fichier route.js

```

<Route path="/ListArticleCard" element={<ListArticleCard/>}/>
<Route path="/Cart" element={<CartArticle/>}/>

```

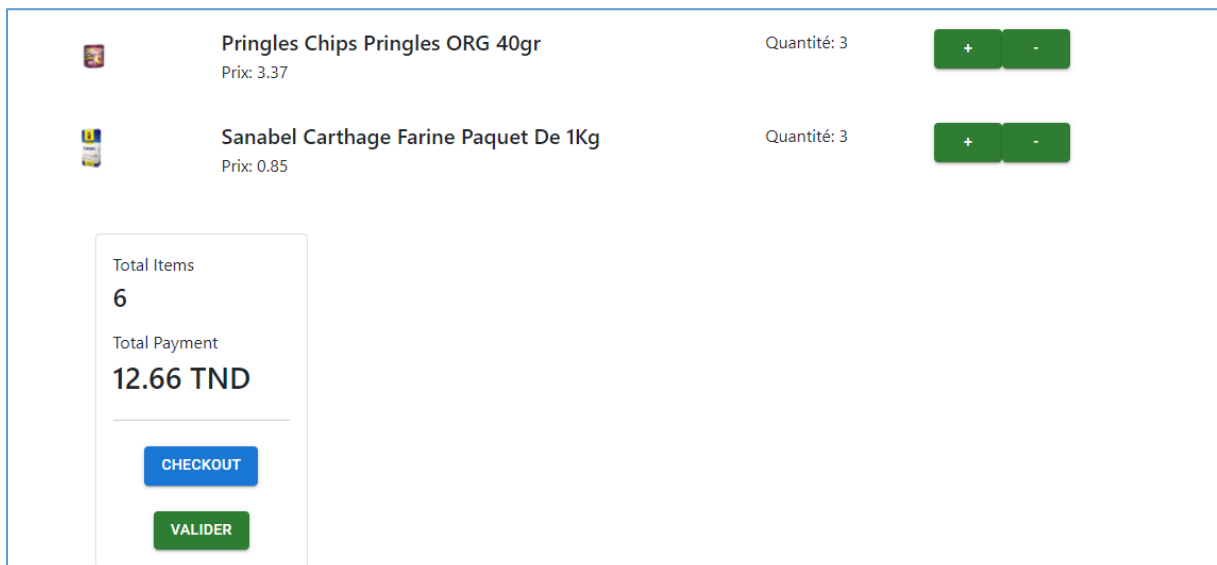
Fichier Navbar.js

```

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

```

A l'adresse <http://localhost:3000/cart> on peut voir notre panier :



Dans ListCardArticles ajouter une barre de menu.

D'abord les imports suivants :

```

import { AppBar } from '@mui/material';
import Toolbar from '@mui/material/Toolbar';

```

Importer aussi Link pour faire le lien vers /Cart

```

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

```

Ensuite l'AppBar pour mettre en place le nombre total des items et le lien vers la page Cart

```
<AppBar position="static">
  <Toolbar>
    <Button color="inherit"><Link to={"/Cart"} style={{ color:
"inherit" }}>{totalItems} Cart</Link></Button>
  </Toolbar>
</AppBar>
```

Le composant ListArticleCard dans sa totalité devient :

```
import {useState,useEffect} from 'react'
import Card from '@mui/material/Card';
import CardActions from '@mui/material/CardActions';
import CardContent from '@mui/material/CardContent';
import CardMedia from '@mui/material/CardMedia';
import Button from '@mui/material/Button';
import Typography from '@mui/material/Typography';
import Axios from "../../Axios/Api";
import { useCart } from "react-use-cart";
import { AppBar } from '@mui/material';
import Toolbar from '@mui/material/Toolbar';
import { Link } from "react-router-dom";

const ListArticleCard=()=>{

  const { addItem, totalItems } = useCart();

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

  useEffect(() => {
    GetListArticles();
  });
  const GetListArticles=async()=>{

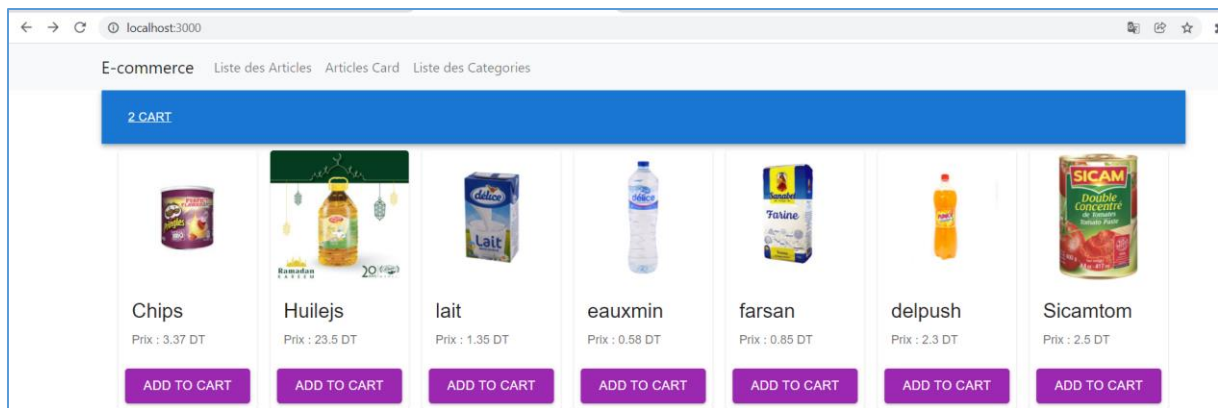
    await Axios.get("/articles")
      .then((res) => {
        setArticles(res.data);
      });
  }
  return(
    <div className="container">
      <AppBar position="static">
```

```

      <Toolbar>
        <Button color="inherit"><Link to={"/Cart"} style={{ color:
"inherit" }}>{totalItems} Cart</Link></Button>
      </Toolbar>
    </AppBar>
    <div
      style={{ "display": "flex", "flexWrap": "wrap", "justifyContent": "center" }}>
      {articles.map((art, ind) => {
        return <Card sx={{ maxWidth: 'auto', margin: 1 }}>
          <CardMedia
            component="img"
            alt="green iguana"
            height="160"
            image={art.imageartpetitf}
          />
          <CardContent>
            <Typography gutterBottom variant="h5" component="div">
              {art.reference}
            </Typography>
            <Typography variant="body2" color="text.secondary">
              Prix : {art.prixVente} DT
            </Typography>
          </CardContent>
          <CardActions>
            <Button disabled={art.qtestock<=1}
              variant="contained" color="secondary" size="large"
              onClick={() => {
                const target = { price: art.prixVente, id: art._id };
                const returnedTarget = Object.assign(target, art);
                addItem(returnedTarget);
              }}
            >
              {art.qtestock<=1? "OUT OF SOLD": "Add to cart"}
            </Button>
          </CardActions>
        </Card>
      })}</div>
    </div>
  )
}
export default ListArticleCard

```





Dans CartArticle ajouter l'import

```
import { Link } from "react-router-dom";
```

Puis modifier le bouton pour faire appel à la page Register

```
<Button variant="contained"><Link to={"/Login"} style={{ color: "inherit"
}}>CHECKOUT</Link></Button><br/><br/>
```

Créer le composant Authentification/Register.js

```
import { useState } from "react";
import { useNavigate, Link } from 'react-router-dom';
import Axios from "../Axios/Api";

function Register() {

  const navigate = useNavigate();
  const [nom, setNom] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleSubmit = async (event) => {
    event.preventDefault();
    const objetuser = {
      nom: nom,
      email: email,
      password: password
    };
    await Axios.post("/users/", objetuser).then((res) => {
      navigate('/Login');
    }).catch(error => {
      console.log(error)
    });
  };
}
```

```

return (
  <>
    <form>
      <div>
        <input
          placeholder="Nom"
          type="text"
          onChange={(event)=>setNom(event.target.value)}
        />
      </div>
      <div>
        <input
          placeholder="Email"
          type="text"
          onChange={(event)=>setEmail(event.target.value)}
        />
      </div>
      <div>
        <input
          placeholder="Password"
          type="password"
          onChange={(event)=>setPassword(event.target.value)}
        />
      </div>
      <div>
        <button onClick={(event)=>handleSubmit(event)}>Valider</button>
      </div>
    </form>
    <div><Link to={"/Login"} style={{ color: "inherit" }}>Login</Link></div>
  </>
);
}

export default Register;

```

Créer le composant Authentification/Login.js

```

import { useState } from "react";
import { useNavigate, Link } from 'react-router-dom';
import Axios from "../Axios/Api";

function Login() {

  const navigate = useNavigate();

  const [email, setEmail]=useState('');

```

```

const [password, setPassword] = useState('');

const handleSubmit = async (event) => {
  event.preventDefault();
  const objetuser = {
    email: email,
    password: password
  };
  await Axios.post("/users/login/", objetuser).then((res) => {
    localStorage.setItem("CC_Token", JSON.stringify(res.data));
    localStorage.setItem("user", JSON.stringify(objetuser));
    navigate('/Cart');
  }).catch(error => {
    console.log(error)
  });
}

return (
  <>
    <form>
      <div>
        <input
          placeholder="Email"
          type="text"
          onChange={(event) => setEmail(event.target.value)}
        />
      </div>
      <div>
        <input
          placeholder="Password"
          type="password"
          onChange={(event) => setPassword(event.target.value)}
        />
      </div>
      <div>
        <button onClick={(event) => handleSubmit(event)}>Valider</button>
      </div>
    </form>
    <div><Link to={"/Register"} style={{ color: "inherit"
}}>Register</Link></div>
  </>

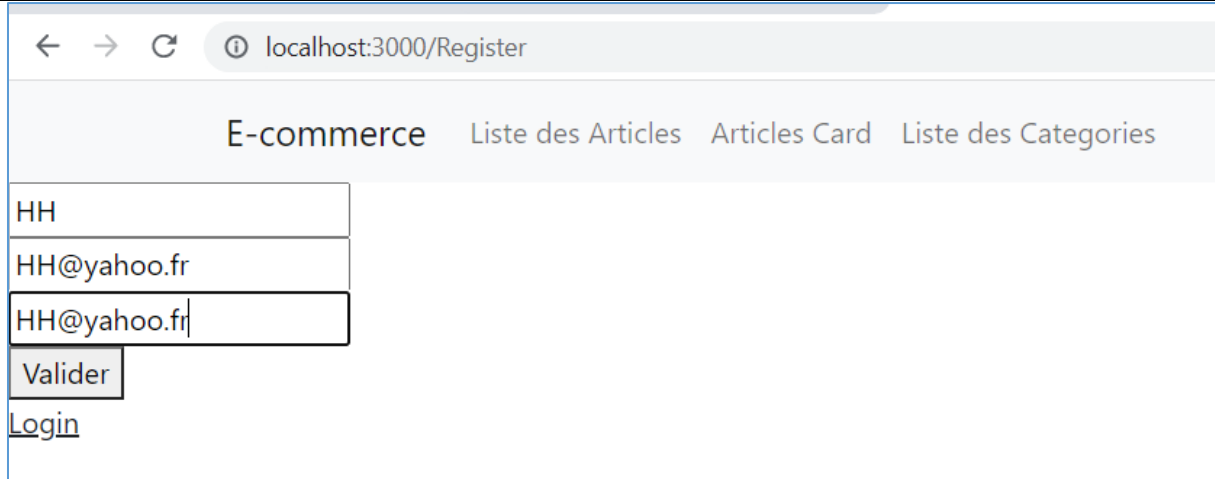
);
}

export default Login;

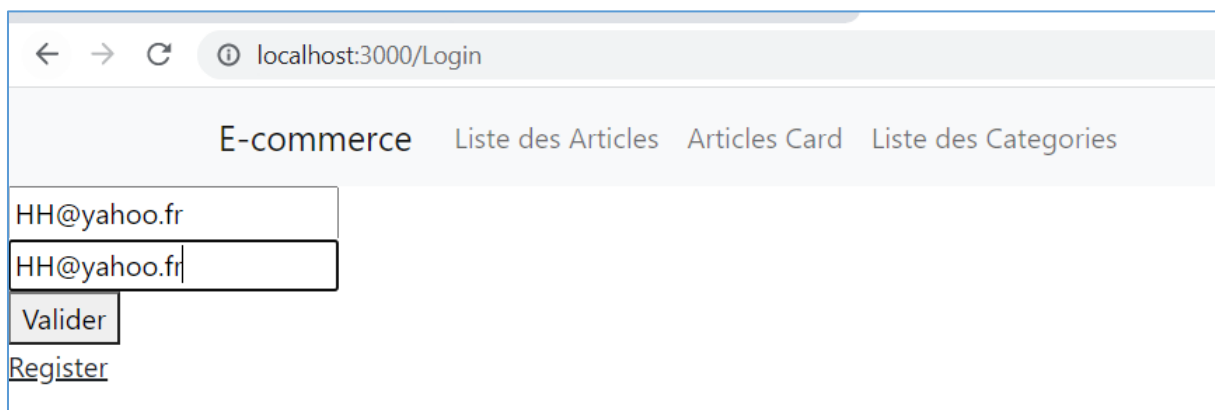
```

Ajouter les routes

```
<Route path="/Login" element={<Login/>}/>
<Route path="/Register" element={<Register/>}/>
```



The screenshot shows a web browser at localhost:3000/Register. The page has a header with 'E-commerce' and navigation links: 'Liste des Articles', 'Articles Card', and 'Liste des Categories'. The main form contains a text input with 'HH', an email input with 'HH@yahoo.fr', another email input with 'HH@yahoo.fr', a 'Valider' button, and a 'Login' link.



The screenshot shows a web browser at localhost:3000/Login. The page has a header with 'E-commerce' and navigation links: 'Liste des Articles', 'Articles Card', and 'Liste des Categories'. The main form contains an email input with 'HH@yahoo.fr', another email input with 'HH@yahoo.fr', a 'Valider' button, and a 'Register' link.

On va modifier CartrArticle.js pour terminer la validation de la commande.

D'abord modifier le bouton VALIDER. Tant que l'utilisateur n'est pas authentifié le bouton doit être caché.

```
{localStorage.getItem("user")?

    <Button
      variant="contained"
      color="success"
      onClick={() => validation()}
    >VALIDER</Button>
    :null
  }
```

Puis ajouter la méthode validation()

```
const validation=async()=>{
```

```

const objetcommande = {
  client: localStorage.getItem("user").email,
  total: cartTotal.toFixed(3),
  articles : items.map((item) => (
    { _id:item._id,
      nom:item.designation,
      qte:item.quantity
    })))
  };
  await Axios.post("/commandes/", objetcommande).then((res)=>{
    localStorage.removeItem("CC-Token");
    localStorage.removeItem("user");
    //Vider le cart
    emptyCart();
    clearCartMetadata();
    navigate('/');
  }).catch(error => {
    console.log(error)
  });
}

```

Ajouter les constantes de useCart qui permettent de vider le panier.

```

const {
  isEmpty,
  totalItems,
  cartTotal,
  items,
  updateItemQuantity,
  removeItem,
  emptyCart,
  clearCartMetadata
} = useCart();

```

Finalement l'import de Axios et de useNavigate

```

import { Link, useNavigate } from "react-router-dom";

import Axios from "../../Axios/Api";

```

Le code complet de CartArticle devient :

```

import { useCart } from "react-use-cart";

import Button from '@mui/material/Button';

```

```

import { Link, useNavigate } from "react-router-dom";

import Axios from "../../Axios/Api";

const CartArticle=()=>{

const navigate = useNavigate();

const {
  isEmpty,
  totalItems,
  cartTotal,
  items,
  updateItemQuantity,
  removeItem,
  emptyCart,
  clearCartMetadata
} = useCart();

if (isEmpty) return <p>Votre panier est vide</p>;

const validation=async()=>{

  const objetcommande = {
    client: localStorage.getItem("user").email,
    total: cartTotal.toFixed(3),
    articles : items.map((item) => (
      { _id:item._id,
        nom:item.designation,
        qte:item.quantity
      })))
  };
  await Axios.post("/commandes/", objetcommande).then((res)=>{
    localStorage.removeItem("CC_Token");
    localStorage.removeItem("user");
    //Vider le cart
    emptyCart();
    clearCartMetadata();
    navigate('/');
  }).catch(error => {
    console.log(error)
  });
}

return (
  <>

```

```

<ul>
  {items.map((item) => (
    <div className="row no-gutters py-2">
      <div className="col-sm-2 p-2">
        <img
          alt={item.designation}
          style={{margin: "0 auto", maxHeight: "50px"}}
          src={item.imageartpetitf} className="img-fluid d-block"/>
        </div>
        <div className="col-sm-4 p-2">
          <h5 className="mb-1">{item.designation}</h5>
          <p className="mb-1">Prix: {item.price} </p>

        </div>
        <div className="col-sm-2 p-2 text-center ">
          <p className="mb-0">Quantité: {item.quantity}</p>
        </div>
        <div className="col-sm-4 p-2 text-right">
          <Button variant="contained" color="success" onClick={() => {
            if(item.quantity < item.qtestock )
              updateItemQuantity(item.id, item.quantity + 1);
            else
              alert("Quantité stock indisponible")
          }}>
+
          </Button>

          {
            item.quantity > 1 &&
            <Button variant="contained" color="success" onClick={() =>

              if(item.quantity < item.qtestock )
                updateItemQuantity(item.id, item.quantity - 1);
              else
                alert("Quantité stock indisponible")
            }}>
-
            </Button>
          }

          {
            item.quantity === 1 &&

            <Button variant="contained" color="error" onClick={() =>
removeItem(item.id)}>
-

```

```

        </Button>

    }

</div>

</div>

    )))}
</ul>

<div className="container">

    <div style={{display:"flex"}}>

        <div className="col-10">
            <div className="col-sm-3 p-3">

                <div className="card card-body">
                    <p className="mb-1">Total Items</p>
                    <h4 className=" mb-3 txt-
right">{totalItems}</h4>

                    <p className="mb-1">Total Payment</p>
                    <h3 className="m-0 txt-
right">{cartTotal.toFixed(3)} TND</h3>
                    <hr className="my-4"/>
                    <div className="text-center">
                        <Button variant="contained"><Link
to={"/Login"} style={{ color: "inherit" }}>CHECKOUT</Link></Button><br/><br/>
                        {localStorage.getItem("user")?
                        <Button
                        variant="contained"
                        color="success"
                        onClick={() => validation()}
                        >VALIDER</Button>
                        :null
                        }
                    </div>

                </div>

            </div>

        </div>

    </div>

</div>

</div>

</div>

```



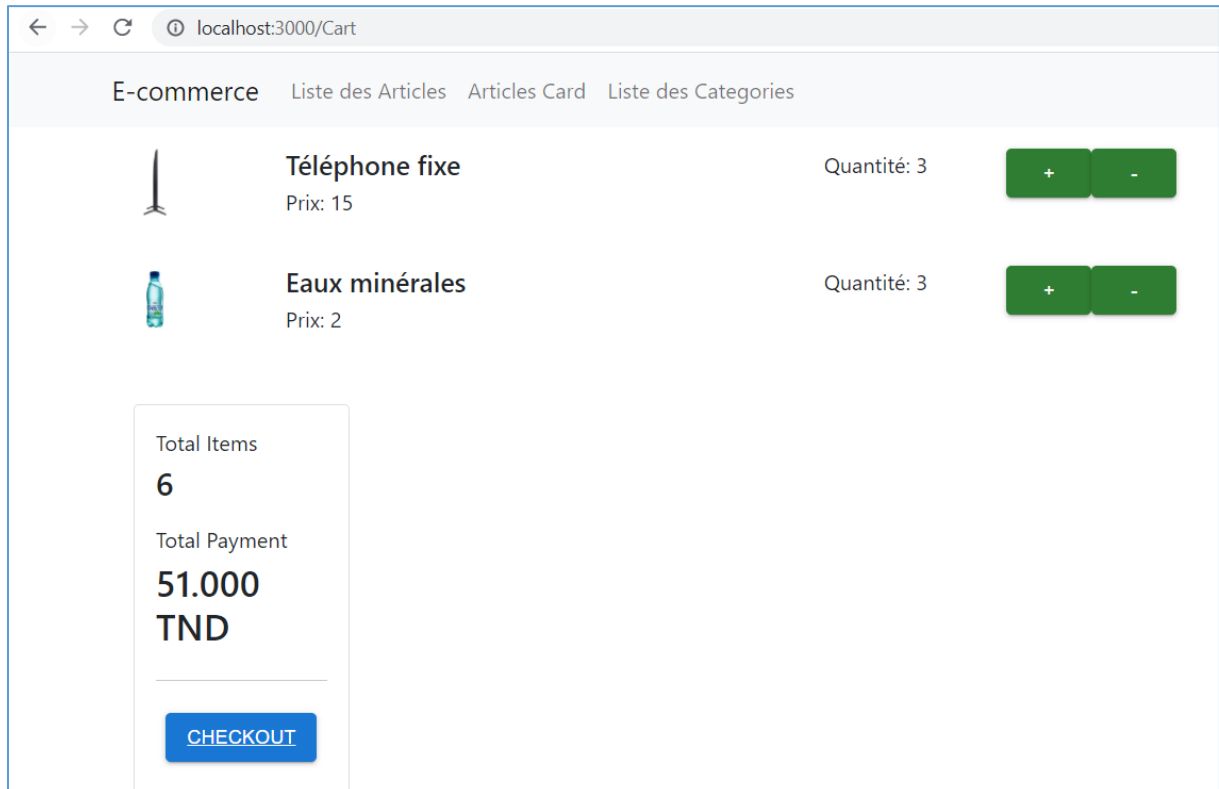
```
</div>
```

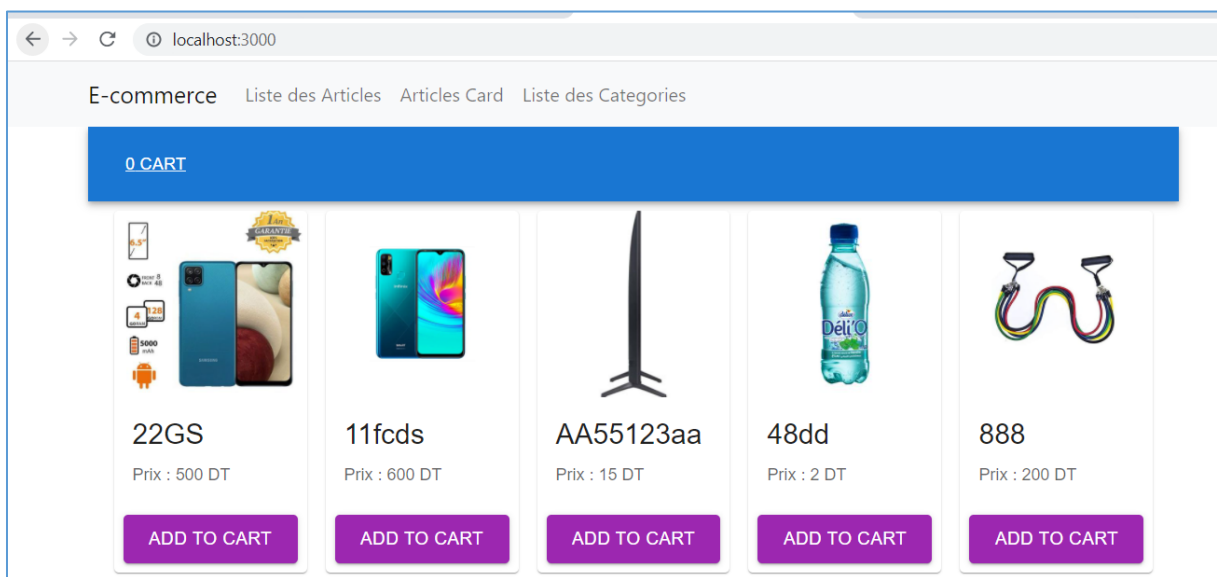
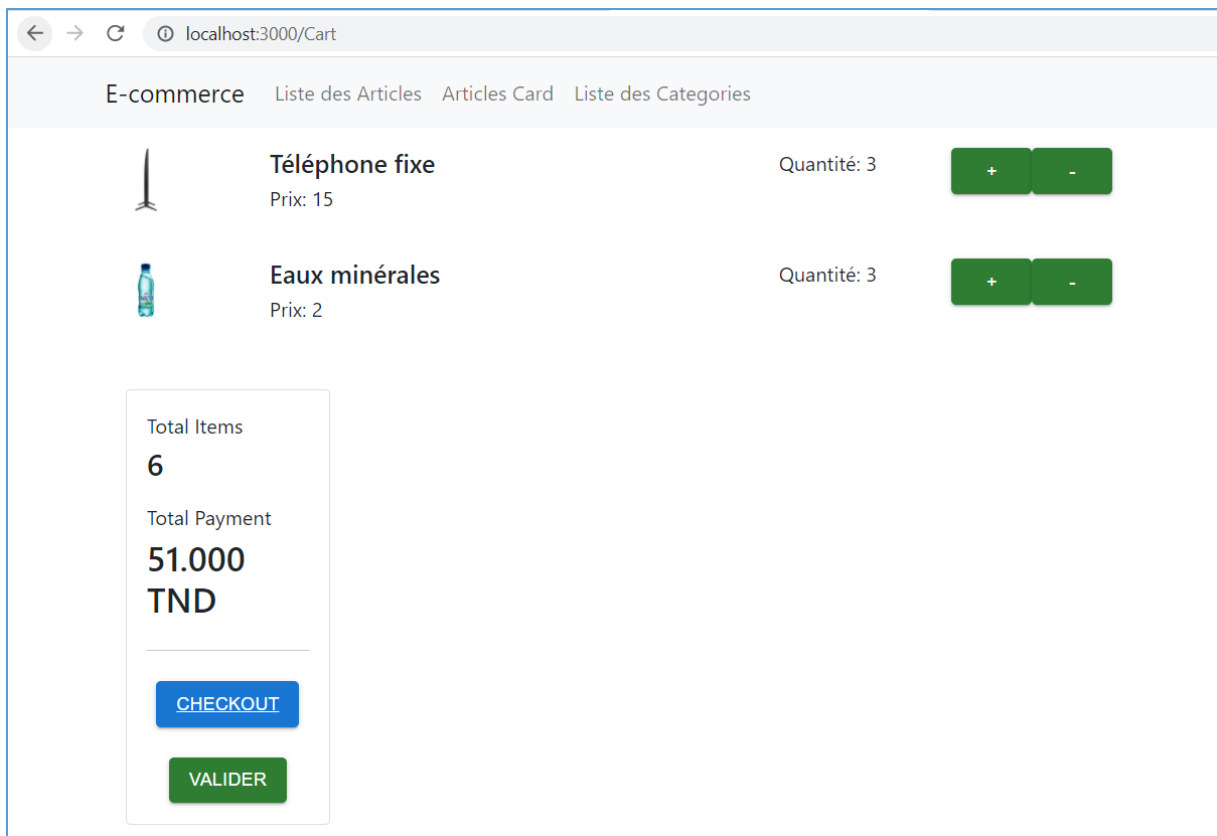
```
</>
```

```
);
```

```
}
```

```
export default CartArticle
```





Maintenant on va modifier la méthode validation du composant cartArticle en soustrayant la quantité commandée de la quantité en stock.

On va ajouter dans la méthode validation le code suivant :

```
items.map((item)=>{
  const objetProduit={
    reference:item.reference,
    designation: item.designation,
```

```

        marque: item.marque,
        prixAchat: item.prixAchat,
        prixVente: item.prixVente,
        prixSolde: item.prixSolde,
        qtestock: item.qtestock-item.quantity,
        caracteristiques: item.caracteristiques,
        imageartpetitf: item.imageartpetitf,
        imageartgrandf: item.imageartgrandf,
        categorieID: item.categorieID,
        scategorieID: item.scategorieID
    }
    Axios.put('/articles/'+item._id,objetProduit ).then((res)=>{
        console.log(res.data)
    });
});
});

```

La méthode validation dans sa totalité deviant :

```

const validation=async()=>{

    //MAj Quantité produit

    items.map((item)=>{
        const objetProduit={
            reference:item.reference,
            designation: item.designation,
            marque: item.marque,
            prixAchat: item.prixAchat,
            prixVente: item.prixVente,
            prixSolde: item.prixSolde,
            qtestock: item.qtestock-item.quantity,
            caracteristiques: item.caracteristiques,
            imageartpetitf: item.imageartpetitf,
            imageartgrandf: item.imageartgrandf,
            categorieID: item.categorieID,
            scategorieID: item.scategorieID
        }
        Axios.put('/articles/'+item._id,objetProduit ).then((res)=>{
            console.log(res.data)
        });
    });

    //Enregistrer la commande


    const objetcommande = {
        client: localStorage.getItem("user").email,
        total: cartTotal.toFixed(3),
        articles : items.map((item) => (

```


```

    { _id:item._id,
      nom:item.designation,
      qte:item.quantity
    })
  });
  await Axios.post("/commandes/", objetcommande).then((res)=>{
    localStorage.removeItem("CC_Token");
    localStorage.removeItem("user");
    //Vider le cart
    emptyCart();
    clearCartMetadata();
    navigate('/');
  }).catch(error => {
    console.log(error)
  });
}

```

Liste des articles													
<input type="checkbox"/>	Reference	Désignation	Prix Achat	Prix Vente	Prix Solde	Marque	Quatité Stock	Image	Catégorie	S/Catégorie	Modification	Suppression	
<input type="checkbox"/>	22GS	GSM	450	500	3	Samsung	18		Electronique	Téléphones	<button>MODIFIER</button>	<button>SUPPRIMER</button>	

[E-commerce](#)
[Liste des Articles](#)
[Articles Card](#)
[Liste des Categories](#)



**GSM**  
 Prix: 500

Quantité: 1
 

+


-

Total Items  
**1**

Total Payment  
**500.000 TND**

CHECKOUT

VALIDER

Liste des articles													
<input type="checkbox"/>	Reference	Désignation	Prix Achat	Prix Vente	Prix Solde	Marque	Quatité Stock	Image	Catégorie	S/Catégorie	Modification	Suppression	
<input type="checkbox"/>	22GS	GSM	450	500	3	Samsung	17		Electronique	Téléphones	<button>MODIFIER</button>	<button>SUPPRIMER</button>	

Par la suite on va créer le composant Components/Commandes/ListCommandes.js qui affichera la totalité des commandes.

```

import {useState,useEffect} from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';

import { ThemeProvider } from "@mui/styles";
import { createTheme } from "@mui/material/styles";
import MUIDataTable from "mui-datatables";
import Axios from "../../Axios/Api";

function ListCommandes() {
  const [listComm,setListComm]=useState([]);
  useEffect(()=>{
    Axios.get("/commandes/").then((res)=>{
      setListComm(res.data)
    })
  });
  const columns = [
    {
      label: "Client",
      name: "client"
    },
    {
      label: "Total",
      name: "total"
    },
    {
      label: "Articles",
      name:"articles",
      options: {
        customBodyRender : (articles) => (
          articles.map((lig,ind)=>{
            return <div key={ind}>
              <div>{lig.nom} - Quantité : - {lig.qte}</div>
            </div>
          })
        )
      }
    },
    {
      label: "Date",
      name: "createdAt"
    }
  ]

  return (
    <>

```

```

<div>

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

    />
  </ThemeProvider>

</div>

</>
);
}
export default ListCommandes;

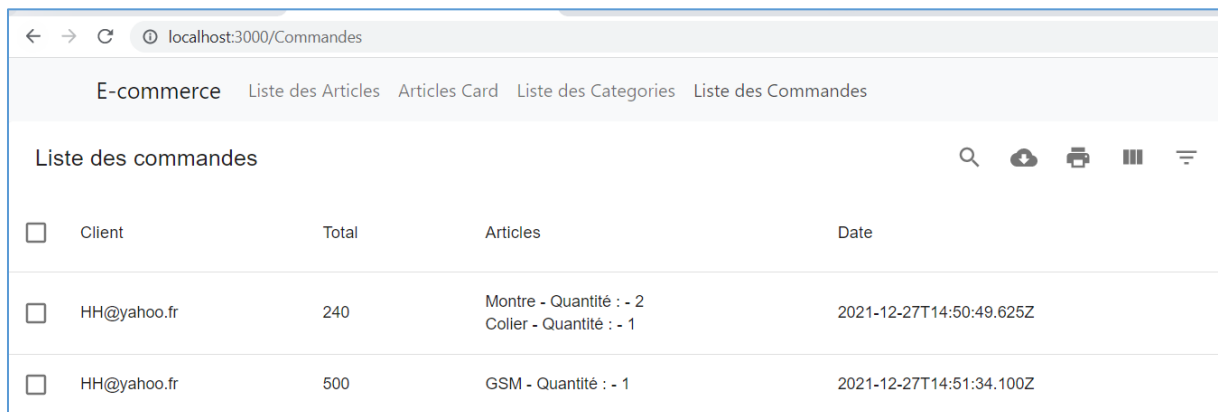
```

On ajoute la route correspondante dans routes.index.js

```
<Route path="/Commandes" element={<Commandes/>}/>
```

Puis on l'ajoute dans le menu de NavBar

```
<Nav.Link as={Link} to="/Commandes">Liste des Commandes</Nav.Link>
```



<input type="checkbox"/>	Client	Total	Articles	Date
<input type="checkbox"/>	HH@yahoo.fr	240	Montre - Quantité : - 2 Colier - Quantité : - 1	2021-12-27T14:50:49.625Z
<input type="checkbox"/>	HH@yahoo.fr	500	GSM - Quantité : - 1	2021-12-27T14:51:34.100Z

### Remarque

Si on veut ajouter le contrôle d'accès à travers le token avant l'enregistrement de la commande on doit envoyer le token dans le headers de la requête pos de commandes.

```
let token=JSON.parse(localStorage.getItem("CC_Token"));
```

```
        await Axios.post("/commandes/", objetcommande, { headers:
{"Authorization" : `Bearer ${token.accessToken}` } }).then((res)=>{
    localStorage.removeItem("CC_Token");
    localStorage.removeItem("user");
    //Vider le cart
    emptyCart();
    clearCartMetadata();
    navigate('/');
}).catch(error => {
    console.log(error)
});
```