



# Rapport de projet

Module : Framework frontend

Enseignant :

M.DIENG Abdoulaye

Membres de groupe :

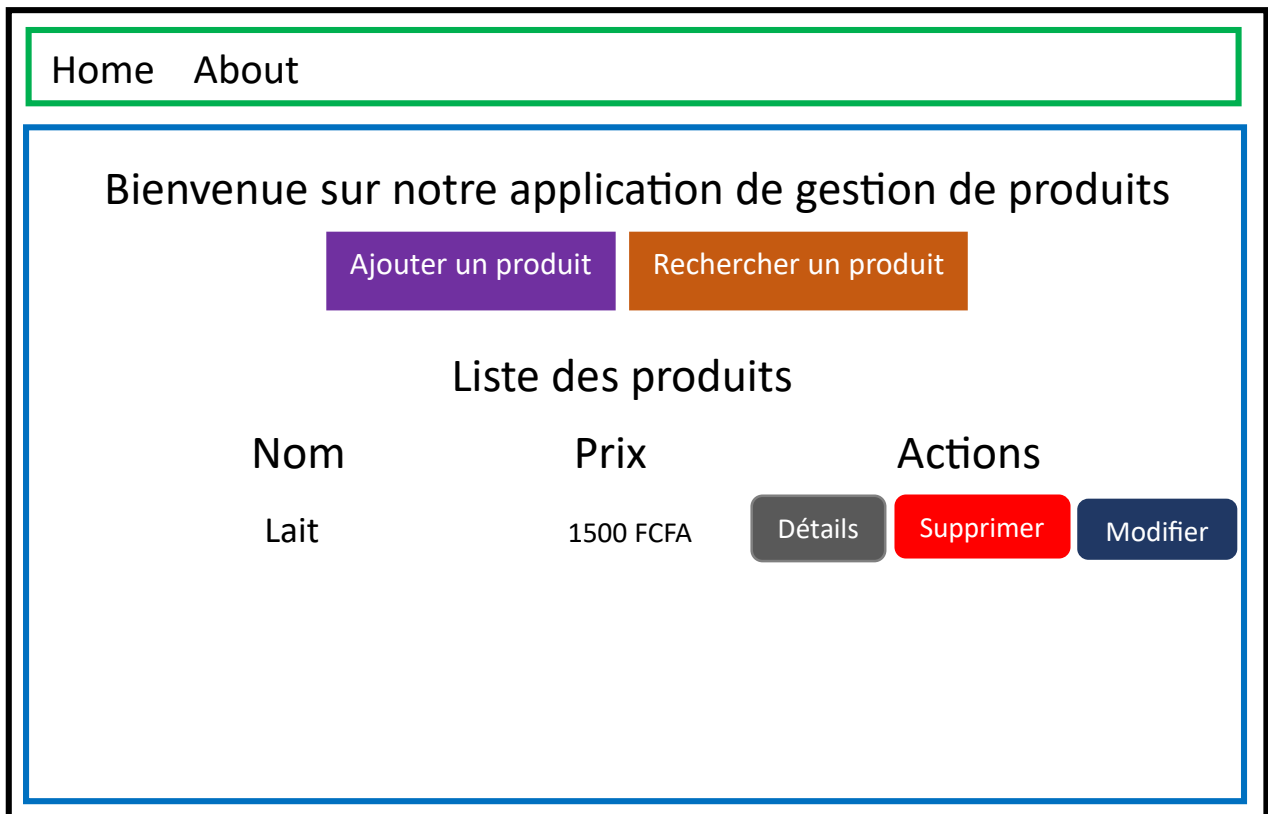
OUEDRAOGO Fabienne

SANOU Sou Roch Landry

## Table des matières

I.	Présentation des composants de notre application .....	3
II.	Présentation des fonctionnalités .....	3
1.	Ajout d'un produit .....	3
2.	Affichage de la liste des produits.....	6
3.	Affichage des détails des produits.....	8
4.	Modification d'un produit .....	10
5.	Suppression d'un produit .....	13
6.	Autres fonctionnalités .....	14
a.	Recherche d'un produit .....	14
b.	Barre navigation et page About.....	17

## I. Présentation des composants de notre application



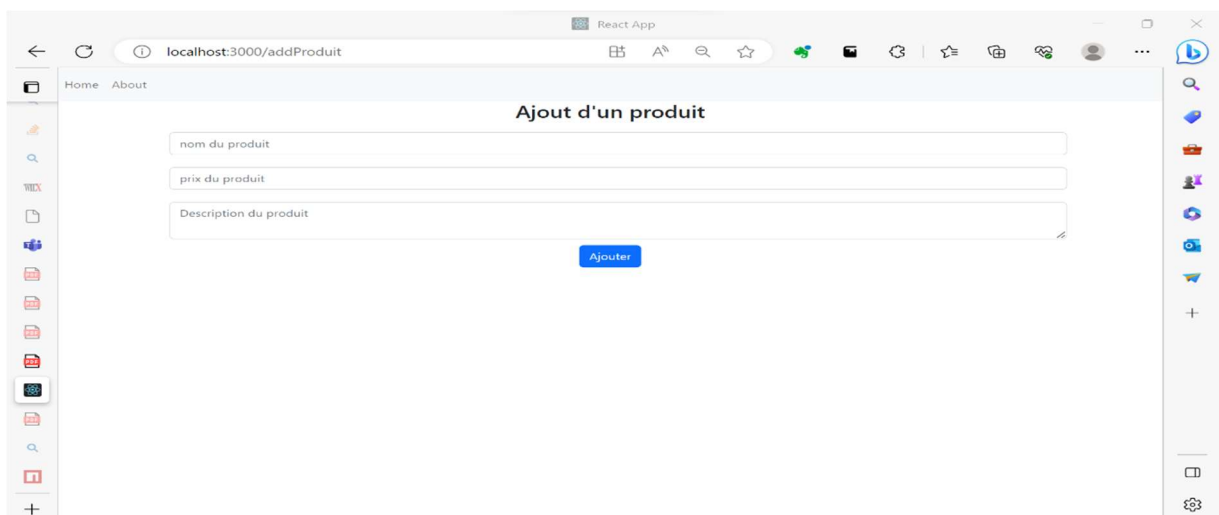
- Composant App
- Composant Details
- Composant AddProduit
- Composant Header
- Composant Magasin
- Composant Search

Les autres éléments en couleur « Supprimer » et « Modifier » ne sont pas de composants.

## II. Présentation des fonctionnalités

### 1. Ajout d'un produit

Pour ajouter un nouveau produit, cliquez sur « Ajouter un produit » dans la page principale et la page suivante s'affichera.



## Les codes sources correspondants

Lien vers la page AddProduit contenu dans le fichier Magasin.js:

```
<button class="btn btn-primary text-center" style={{margin:"5px"}}><a  
class="navbar-brand" href="/addProduit">Ajouter un produit</a></button>
```

Code source correspondant à l'affichage du formulaire d'ajout d'un produit. Ce code se trouve dans le fichier AddProduit.js:

```
import React, {useState } from "react";

function AddProduit({ handleAdd }) {
  const [nom, setNom] = useState('');
  const [prix, setPrix] = useState('');
  const [description, setDescription] = useState('');
  const placeholderName = "nom du produit";
  const placeholderPrix = "prix du produit";
  const placeholderDescription = "Description du produit";

  const handleChangeNom = (event) => {
    setNom(event.target.value);
  }

  const handleChangePrix = (event) => {
    setPrix(event.target.value);
  }

  const handleChangeDescription = (event) => {
    setDescription(event.target.value);
  }

  const handleSubmit = (event) => {
    event.preventDefault();
    handleAdd(nom,prix,description);
    setNom('');
    setPrix('');
    setDescription('');
  }

  return (
    <div class="container-fluid" className="my_media_query">
      <form onSubmit={handleSubmit}>
        <div class="container">
          <h3 class="text-center" id="titre">Ajout d'un produit</h3>
          <div class="form-group row" >
            <div class="col-12" style={{margin:"10px"}}>
```



```
export default AddProduit
```

Code source représentant la fonction d'ajout d'un nouveau produit. Ce code se trouve dans le fichier App.js:

```
const [produits, setProduits] = useState([]);
const handleAdd=(nom,Prix,Description)=> {
  axios
    .post('http://localhost:3333/produits/', { nom, Prix,Description})
    .then((res) => {
      setProduits([...produits, res.data]);
    });
}
```

Route vers la page AddProduit. Ce code se trouve dans le fichier App.js

```
<Route path="/addProduit" element={<AddProduit handleAdd={handleAdd}/>}/>
```

## 2. Affichage de la liste des produits

La liste des produits s'affiche automatiquement sur la page principale. Il n'y pas de bouton à cliquer.

Liste des produits		
Nom	Prix	Actions
Lait	1500 FCFA	<button>Détails</button> <button>Supprimer</button> <button>Modifier</button>
Sardine	600 FCFA	<button>Détails</button> <button>Supprimer</button> <button>Modifier</button>
Préssea orange	1000 FCFA	<button>Détails</button> <button>Supprimer</button> <button>Modifier</button>
Sucre	300 FCFA	<button>Détails</button> <button>Supprimer</button> <button>Modifier</button>

## Les codes sources correspondants

Code source correspondant à l'affichage de la liste des produits se trouve dans le fichier Magasin.js.

```
const [products, setProduits] = useState([]);
const [isLoading,setIsLoading] = useState(true)
const [error,setError] = useState(null);
```

```

useEffect((event) => {
  axios
    .get('http://localhost:3333/produits')
    .then((res) => {
      setProduits(res.data);
      setIsLoading(false);
    })
    .catch((error) => {
      setError(error);
      setIsLoading(false);
    });
}, [error]);

```

```

<div class="col-12" id="main" className='my_media_query'>
  <h3 class="text-center">Liste des produits</h3>
  {isLoading ? (
    <p>Loading...</p>
  ): (
    <div >
      <table class="col-9 offset-3">
        <thead>
          <tr>
            <th>Nom</th>
            <th>Prix</th>
            <th>Actions</th>
          </tr>
        </thead>
        <tbody >

          {products.map((produit) =>(
            <tr key={produit.id} >
              <td> {produit.nom} </td>
              <td> {produit.Prix} FCFA</td>
              <td>

                <button class="btn btn-primary" style={{margin:"2px"}} onClick={() =>
                  handleDetails(produit.id)} >Détails</button>

                <button
                  style={{margin:"2px"}}class="btn btn-danger btn-xs" onClick={() =>
                    handleDelete(produit.id)}>Supprimer</button>

                <button style={{margin:"2px"}}
                  class="btn btn-warning btn-xs" onClick={() =>
                    handleModifier(produit.id,produit.nom,produit.Prix,produit.Description)}>Modif
                  ier</button>

              </td>
            </tr>
          )) }
        </tbody>
      </table>

```

```
</div>
```

```
)}  
</div>
```

### 3. Affichage des détails des produits

Pour visualiser les détails d'un produit, aller au niveau de la liste des produits et cliquer sur le bouton « Détails » et les détails du produit correspondant s'affichera en bas.

Illustration avec le produit « Sucre » :

Liste des produits		
Nom	Prix	Actions
Lait	1500 FCFA	<button>Détails</button> <button>Supprimer</button> <button>Modifier</button>
Sardine	600 FCFA	<button>Détails</button> <button>Supprimer</button> <button>Modifier</button>
Préssea orange	1000 FCFA	<button>Détails</button> <button>Supprimer</button> <button>Modifier</button>
Sucre	300 FCFA	<button>Détails</button> <button>Supprimer</button> <button>Modifier</button>

Détails du produit Sucre	
Identifiant: 11	
Nom: Sucre	
Prix: 300 FCFA	
Description: Sucre en poudre un quart de kilo	

### Les codes sources correspondants

Ce code source se trouve dans le fichier Magasin.js :

```
const [product, setProduit] = useState([]);
```

```
const handleDetails=(id)=> {  
  axios  
    .get('http://localhost:3333/produits/' + id)  
    .then((res) =>  
      setProduit(res.data)  
    )  
    .catch((error) => {  
      setError(error);  
      setIsLoading(false);  
    });  
}
```



```
});
}
```

```
button class="btn btn-primary" style={{margin:"2px"}} onClick={() =>
handleDetails(produit.id)} >Détails</button>
```

```
{
(product.nom && product.id && product.Prix && product.Description &&
!selectedProduit) && (
    <div>

        {isLoading ? (
            <p>Loading...</p>
        ) : (

            <Details
                key={product.id}
                product={product}
            />
        )}

    </div>
)
}
```

Celui se trouve dans le fichier Details.js :

```
import React from 'react'

function Details(props) {
    const {product} = props;
    return (
        <div>
            {
                (product.nom && product.id && product.Prix && product.Description) ?
            (

                <div>
                    <h3 class="text-center">Détails du produit {product.nom}</h3>
                    <ul>
                        <li class="list-unstyled "><p class="font-weight-bold"> Identifiant:
{product.id}</p></li>
                        <li class="list-unstyled font-weight-bold"><p class="font-weight-bold">Nom:
{product.nom}</p></li>
```



## Les codes sources correspondants

Ce code se trouve dans le fichier Magasin.js :

```
const [selectedProduit, setSelectedProduit] = useState(null);
const [modifiedProduitName, setModifiedProduitName] = useState('');
const [modifiedProduitPrix, setModifiedProduitPrix] = useState('');
const [modifiedProduitDescription, setModifiedProduitDescription]
= useState('');
```

```
const handleModifier = (id,nom,prix,description) =>{
  setSelectedProduit(id);
  setModifiedProduitName(nom);
  setModifiedProduitPrix(prix);
  setModifiedProduitDescription(description);
}
```

```
const saveModifiedProduit = () => {
  axios
    .put(`http://localhost:3333/produits/${selectedProduit}`, {
      nom: modifiedProduitName,
      Prix: modifiedProduitPrix,
      Description:modifiedProduitDescription,

    })
    .then(() => {
      setProduits((prevProduits) =>
        prevProduits.map((produit) =>
          produit.id === selectedProduit
            ? { ...produit, nom:
modifiedProduitName,Prix:modifiedProduitPrix, Description:
modifiedProduitDescription }
            : produit
          )
        );
      setSelectedProduit(null);
      setModifiedProduitName('');
      setModifiedProduitPrix('');
      setModifiedProduitDescription('');

    })
    .catch((error) => setError(error));
};
```

```

{selectedProduit && (

    <div class="col-12">
        <div class="form-group row" >
            <div class="col-12" style={{margin:"10px"}}>
                <input
                    type="text"
                    required
                    class="form-control"
                    value={modifiedProduitName}
                    onChange={e =>
setModifiedProduitName(e.target.value)}
                />
            </div>
        </div>
        <div class="form-group row" >
            <div class="col-12" style={{margin:"10px"}}>
                <input
                    type="number"
                    required
                    class="form-control"
                    value={modifiedProduitPrix}
                    onChange={e => setModifiedProduitPrix(e.target.value)}
                />
            </div>
        </div>
        <div class="form-group row" >
            <div class="col-12" style={{margin:"10px"}}>
                <input
                    type="text"
                    required
                    class="form-control"
                    value={modifiedProduitDescription}
                    onChange={e =>
setModifiedProduitDescription(e.target.value)}
                />
            </div>
        </div>
        <button onClick={saveModifiedProduit} class="btn btn-info d-
block mx-auto">Sauvegarder</button>
    </div>
) }

```

```
<button style={{margin:"2px"}} class="btn btn-warning btn-xs" onClick={() =>
handleModifier(produit.id,produit.nom,produit.Prix,produit.Description)}>Modif
ier</button>
```

Lorsque la modification d'un produit est sauvegardée, la liste des produits est automatiquement actualisée.

## 5. Suppression d'un produit

Pour supprimer un produit, aller au niveau de la liste de produits et cliquer sur le bouton « Supprimer » et le produit correspondant sera automatiquement supprimé de la liste des produits.

Illustration avant la suppression du produit « Sucre » :

Liste des produits		
Nom	Prix	Actions
Lait	1500 FCFA	Détails Supprimer Modifier
Sardine	600 FCFA	Détails Supprimer Modifier
Préssea orange	1000 FCFA	Détails Supprimer Modifier
Sucre	300 FCFA	Détails Supprimer Modifier

Illustration après la suppression du produit « Sucre » :

Liste des produits		
Nom	Prix	Actions
Lait	1500 FCFA	Détails Supprimer Modifier
Sardine	600 FCFA	Détails Supprimer Modifier
Préssea orange	1000 FCFA	Détails Supprimer Modifier

Nous voyons que le produit « Sucre » n'existe plus dans la liste des produits.

## Les codes sources correspondants

Ce code se trouve dans le fichier Magasin.js :

```
<button style={{margin:"2px"}}class="btn btn-danger btn-xs" onClick={() =>
handleDelete(produit.id)}>Supprimer</button>
```

```
const handleDelete=(id)=> {
  axios
    .delete('http://localhost:3333/produits/' + id)
    .then((res) =>
      setProduits((prevProduits) =>
        prevProduits.filter((produit) => produit.id !== id)
      )
    )
    .catch((error) => {
      setError(error);
      setIsLoading(false);
    });
}
```

### 6. Autres fonctionnalités

#### a. Recherche d'un produit

L'objectif de la recherche d'un produit est de faciliter la visualisation par exemple du prix d'un produit parmi une liste de plusieurs produits.

Pour rechercher un produit, cliquer sur « Rechercher un produit » au niveau de la page principale et la page suivante s'affichera :

Home	About
Recherche d'un produit	
Rechercher un produit	
Nom	Prix

Ce code se trouve dans le fichier Search.js :

```
import React from 'react'
import produits from '../data/produits.json'
import { useState } from 'react';
import style from 'react';

function Search() {
  const [searchTerm, setSearchTerm] = useState('');
  const [searchResults, setSearchResults] = useState([]);

  const handleSearchSubmit = (event) => {
    event.preventDefault();
    const results = produits.produits.filter((produit) =>
      produit.nom.toLowerCase().includes(searchTerm.toLowerCase())
    );
    setSearchResults(results);
  };
  const handleSearchChange = (event) => {
    setSearchTerm(event.target.value);
  };
  return (
    <div class=" container-fluid row " className="my_media_query">
      <h2 class="text-center" className="my_media_query"
id="titre">Recherche d'un produit</h2>
      <div class="col-12">
        <input
          name="nom"
          type="text"
          value={searchTerm}
          onChange={handleSearchChange}
          placeholder="Rechercher un produit"
          class="form-control"
          onKeyUp={handleSearchSubmit}
        />
      </div>
      <div class="col-6 offset-3">
        <div>
          <table class="table table-striped">
            <thead>
              <tr>
                <th>Nom</th>
                <th>Prix</th>
              </tr>
            </thead>
            <tbody >
              {searchResults.map((produit) =>(
                <tr key={produit.id} >
                  <td> {produit.nom} </td>
```

```

        <td> {produit.Prix}
FCFA</td>
      </tr>
    )) }
  </tbody>
</table>
</div>
</div>
</div>
)
}

export default Search

```

Lien vers la page de recherche se trouve dans le fichier Magasin.js :

```

<button class="btn btn-primary text-center"><a class="navbar-brand"
href="/searchProduit">Rechercher un produit</a></button>

```

Route vers la page de recherche d'un produit se trouve dans le fichier App.js :

```

<Route path="/searchProduit" element={<Search/>}/>

```

Pour rechercher un produit, il suffit d'entrer le nom du produit dans le champ de saisie et le produit s'affiche s'automatiquement grace à un évènement **onKeyUp** sur le champ du formulaire.

Illustration avec la recherche du produit « Lait » :

Home About	
Recherche d'un produit	
Lait	
Nom	Prix
Lait	1500 FCFA



## b. Barre navigation et page About

La barre de navigation sert à passer d'une page à une autre. Dans notre cas, elle permet de passer de la page Home à la page About et vice versa.



## Les codes sources correspondant

Ce code se trouve dans le fichier Header.js

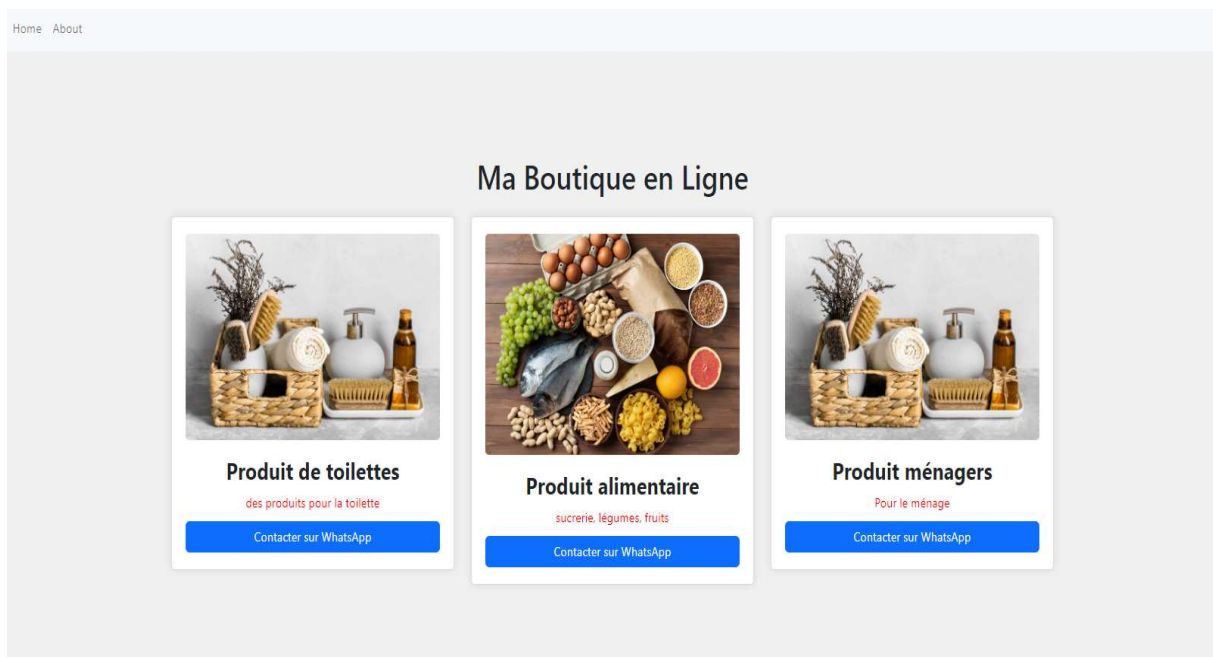
```
import React from 'react'

function Header() {
  return (
    <div class="container-fluid" className="my_media_query">
      <header>
        <nav class="navbar navbar-expand-md navbar-light bg-light fixed-top col-
md-12 col-xs-12 col-sm-12 col-lg-12 col-xl-12">
          <ul class="navbar-nav mr-auto">
            <div><li class="col-6 nav-item active"> <a href="/"
class="nav-link">Home</a></li></div>
            <div><li class="col-6 nav-item"> <a href="/about" class="nav-
link">About</a></li></div>
          </ul>
        </nav>
      </header>
    </div>

  )
}

export default Header
```

La page About est chargé décrire les produits de notre application avec des images à l'appui, et un bouton qui nous dirige sur le contact WhatsApp du propriétaire de notre boutique :



Le code source correspondant :

Ce code se trouve dans le fichier About.js :

```
import React from 'react';
import './About.css';

function About() {
  return (

    <div className="About">
      <div className="container">
        <h1 className="text-center my-4">Ma Boutique en Ligne</h1>
        <div className="row">
          <div className="col-lg-4 col-md-6 mb-4">
            <div className="product-card">
              
              <h3 className="product-title">Produit de toilettes</h3>
              <p className="product-description">des produits pour la toilette</p>
              <a href="https://api.whatsapp.com/send?phone=00221774718835" target="_blank" rel="noopener noreferrer" className="btn btn-primary">
```

```

    Contacter sur WhatsApp
  </a>

    </div>
  </div>

  <div className="col-lg-4 col-md-6 mb-4">
    <div className="product-card">
      
      <h3 className="product-title">Produit alimentaire</h3>
      <p className="product-description">sucrerie, légumes,
fruits</p>
      <a href="https://api.whatsapp.com/send?phone=00221774718835"
target="_blank" rel="noopener noreferrer" className="btn btn-primary">
    Contacter sur WhatsApp
  </a>

    </div>
  </div>

  <div className="col-lg-4 col-md-6 mb-4">
    <div className="product-card">
      
      <h3 className="product-title">Produit ménagers</h3>
      <p className="product-description">Pour le ménage</p>
      <a href="https://api.whatsapp.com/send?phone=00221774718835"
target="_blank" rel="noopener noreferrer" className="btn btn-primary">
    Contacter sur WhatsApp
  </a>

    </div>
  </div>

  { /* Ajoutez plus de colonnes ici pour plus de produits */ }
</div>
</div>
</div>
);
}

export default About;

```

## Code du fichier About.css

```
.About {
  background-color: #f0f0f0; /* Remplacez cette valeur par la couleur de
fond souhaitée */
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
}

.my-4 {
  margin-top: 4rem;
  margin-bottom: 4rem;
}

/* Styles pour les cartes produits */
.product-card {
  background-color: #fff;
  border: 1px solid #ddd;
  border-radius: 5px;
  padding: 20px;
  text-align: center;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.product-card img {
  max-width: 100%;
  height: auto;
  border-radius: 5px;
  margin-bottom: 20px;
}

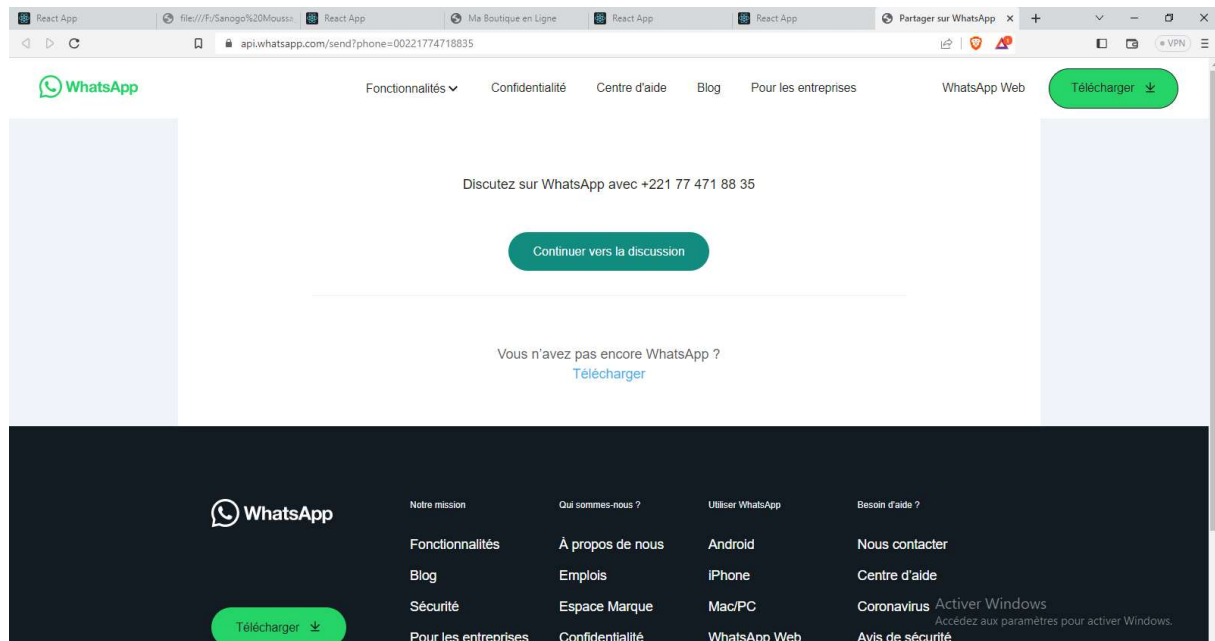
.product-card .product-title {
  font-weight: bold;
  margin-bottom: 10px;
}

.product-card .product-description {
  color: #f00;
  margin-bottom: 10px;
}

.product-card .btn {
```

```
width: 100%;  
}
```

Résultat du clic sur un bouton « Contacter sur WhatsApp » de la page About :



Le code CSS pour gérer la responsivité de notre application

Ce code se trouve dans le fichier App.css :

```
.my_media_query{  
  
  @media (max-width: 576px) {  
    #titre {  
      margin-top: 20%;  
      text-align: center;  
    }  
    .navbar{  
      position: relative;  
    }  
  }  
  @media (min-width: 577px) and (max-width: 992px) {  
    #titre{  
      margin-top: 10%;  
      text-align: center;  
    }  
    .navbar{
```

```
        position: relative;

    }
}

@media (min-width: 993px) {
    #titre {
        margin-top: 5%;
        text-align: center;
    }
    .navbar{
        position: relative;
    }
}
}
```