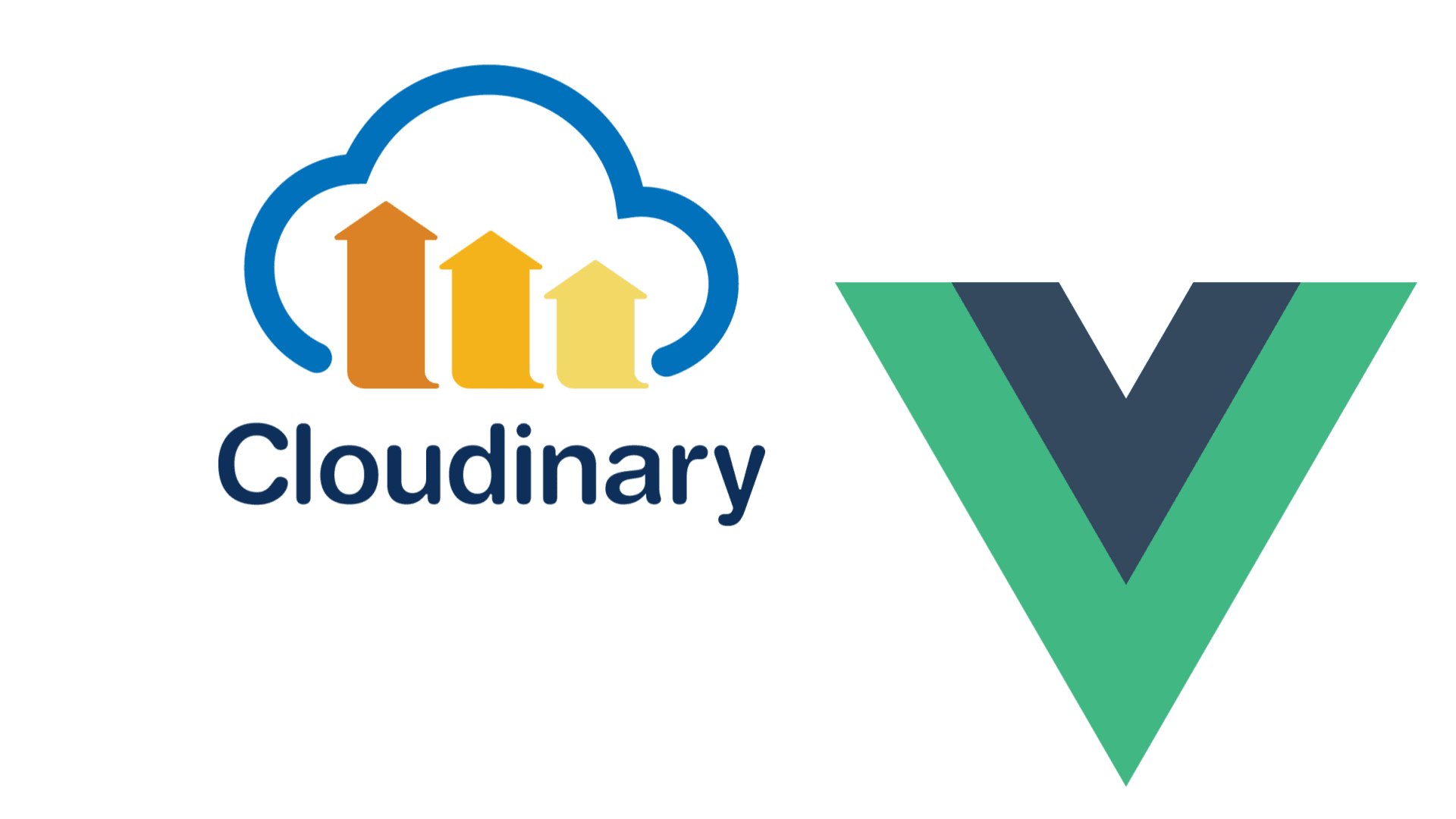
Atelier 4 : CRUD et Upload images



## Liste des articles

Pour l’affichage des données on va utiliser « DataTables for jQuery with styling for DataTables ».

Visiter :

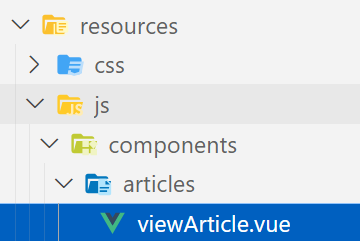
<https://www.npmjs.com/package/datatables.net-dt>

Installer le package :

npm i datatables.net-dt

Pour rappel jquery et bootstrap sont déjà installés.

Créer :



<template>

    <div>

        <h2 class="text-center"> Liste des articles </h2>

        <table class="table" id="example">

            <thead>

            <tr>

          <th>Référence</th>

          <th>Désignation</th>

          <th>Marque</th>

          <th>Prix</th>

          <th>Quantité</th>

          <th>Sous Catégorie</th>

          <th>Image</th>

          <th>Actions</th>

        </tr>

            </thead>

            <tbody>

            <tr v-for="p in Articles" :key="p.id">

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

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

                <td>{{p.marque}}</td>

                <td>{{p.prix}}</td>

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

                <td>{{p.scategories.nomscategorie}}</td>

                <td><img :src="p.imageart" :alt=p.designation width="100" /></td>

                <td>

                    <button @click.prevent="deleteArticle(p.id)" class="btn btn-danger">Supprimer</button></td>

            </tr>

            </tbody>

        </table>

    </div>

</template>

<script setup>

import axios from "../config/axios.js";

//Bootstrap and jQuery libraries

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

import 'jquery/dist/jquery.min.js';

//Datatable Modules

import "datatables.net-dt/js/dataTables.dataTables"

import "datatables.net-dt/css/jquery.dataTables.min.css"

import $ from 'jquery';

import { ref, onMounted } from 'vue';

const Articles = ref([]);

 onMounted(() => {

            getArticles();

            }

);

const getArticles=()=>{

                  axios

                .get('/api/articles/')

                .then(response => {

                                 Articles.value = response.data;

                                 $(function() {$('#example').DataTable();});

                });

            }

const deleteArticle=(id)=> {

                 if (window.confirm("Etes-vous sûr de vouloir supprimer ?")) {

                   axios

                    .delete(`/api/articles/${id}`)

                    .then(() => {

                        let i = Articles.value.map(data => data.id).indexOf(id);

                        Articles.value.splice(i, 1)

                    })

                    .catch(error => {

                                console.log(error)

                            });

                 }

            }

</script>

On remarque l’utilisation de {{p.scategories.nomscategorie}} pour afficher la sous catégorie correspondante .Et c’est grâce à la relation One to Many qui a été mise en œuvre dans Laravel.

## Réglage des routes

Dans routes.js ajouter la route correspondante à la liste des articles.



import ViewArticle from "./components/articles/viewArticle.vue";

 {

        path: '/articles',

        name: 'articles',

        component: ViewArticle

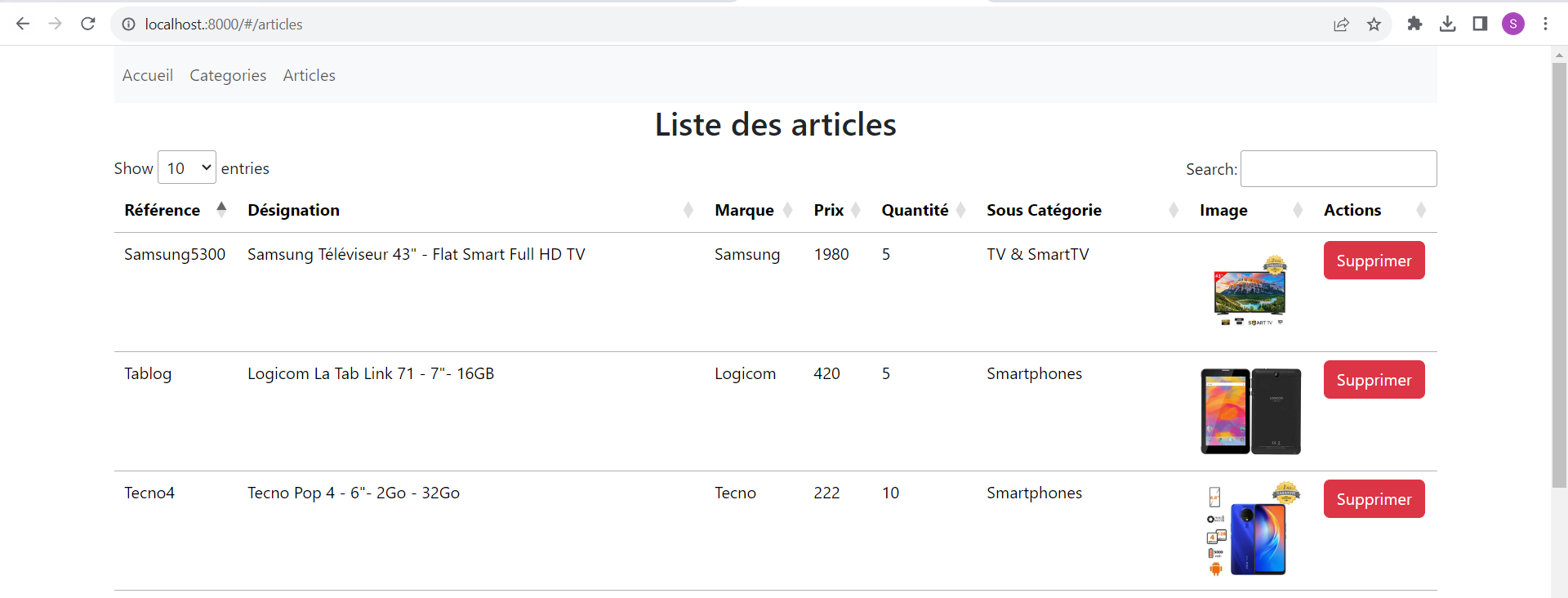
    },

## Mise à jour du menu

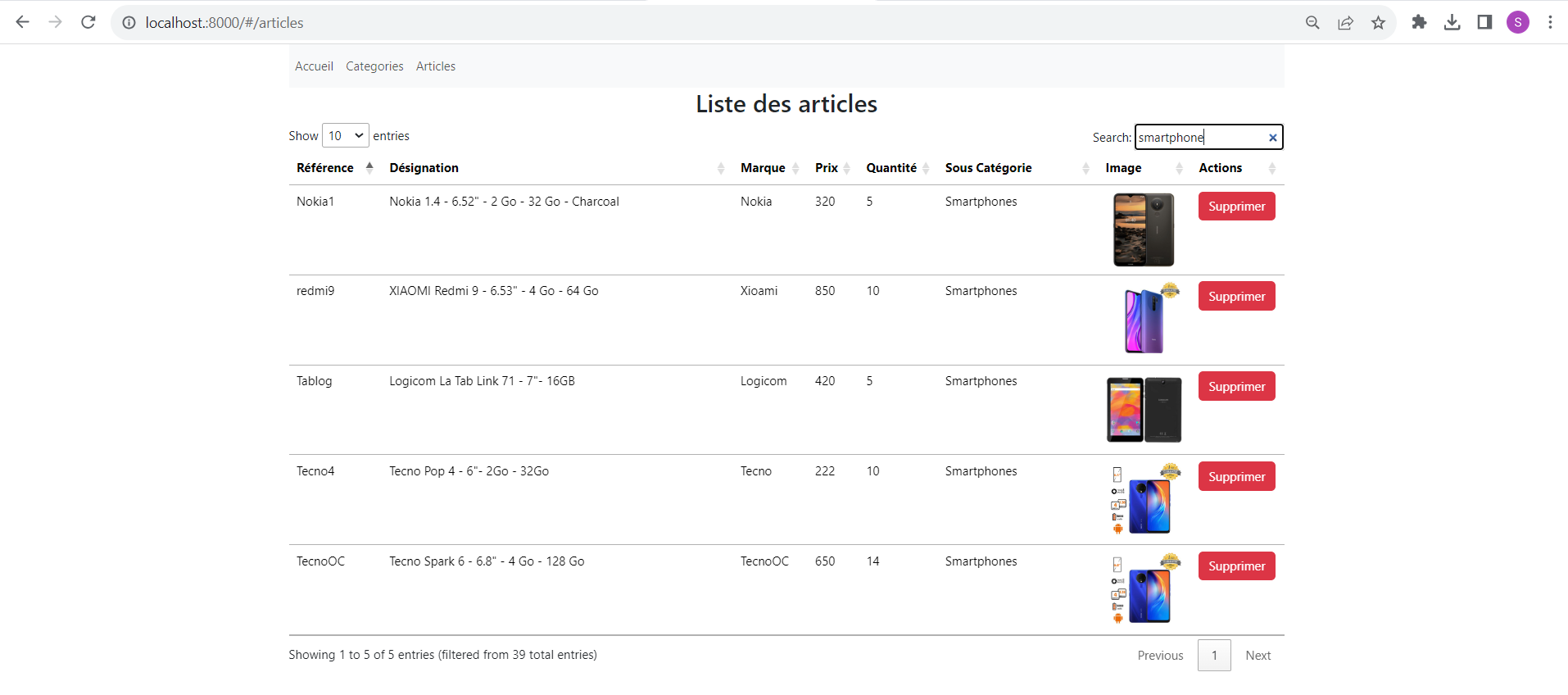
Ajouter le lien vers le composant viewArticle dans App.vue



   <router-link to="/articles" class="nav-item nav-link">  Articles </router-link>







## Création de AddArticle

Maintenant, nous allons créer le formulaire d’ajout pour les articles en faisant appel à la requête axios.post().

On va commencer par créer le fichier addArticle.vue sous resources/js/components/articles.



<template lang="">

    <form @submit.prevent="ajouterproduit">

       <div class="form-group">

       <input type="text" class="form-control" placeholder="reference" v-model="article.reference">

     </div>

     <div class="form-group">

       <input type="text" class="form-control" placeholder="designation" v-model="article.designation">

     </div>

     <div class="form-group">

       <input type="text" class="form-control" placeholder="marque" v-model="article.marque">

     </div>

    <div class="form-group">

       <input type="number" class="form-control" placeholder="prix" v-model="article.prix">

     </div>

    <div class="form-group">

       <input type="number" class="form-control" placeholder="qtestock" v-model="article.qtestock">

     </div>

    <div class="form-group">

       Sous Catégories<select class="form-control" v-model="article.scategorieID">

          <option v-for="sc in Scategories" :key="sc.id" :value=sc.id>{{sc.nomscategorie}}</option>

       </select>

     </div>

     <div class="form-group">

        <input type="text" class="form-control" placeholder="image" v-model="article.imageart">

    </div>

    <button type="submit" class="btn btn-block btn-primary">Ajouter Produit</button>

   </form>

   </template>

   <script setup>

   import axios from "../config/axios.js";

   import { useRouter } from 'vue-router';

   const router = useRouter()

import { ref, onMounted } from 'vue';

const Scategories = ref([]);

 onMounted(() => {

    getscategories();

            }

);

const getscategories=()=>{

    axios.get('/api/scategories').then(res => {

        Scategories.value = res.data;

            }).catch(error => {

                console.log(error)

            });

         }

let  article= {

               reference: "",

               designation: "",

               marque: "",

               prix: "",

               qtestock: "",

               imageart: "",

               scategorie:"",

       }

const  ajouterproduit=()=>{

          axios.post("/api/articles",article)

               .then(() => {

                     router.push('/articles')})

               .catch(error => {

                  console.error("There was an error!", error);})

                  }

   </script>

## Réglage des routes

Dans routes.js ajouter la route correspondante au formulaire d’ajout.



import AddArticle from "./components/articles/addArticle.vue";

  {

        path: '/addArticle',

        name: 'addArticle',

        component: AddArticle

    },

## Mise à jour de la liste



  <router-link :to="{name: 'addArticle'}" class="btn btn-primary">Ajout Article</router-link>



