Atelier 10 : Création et consommation d'une API REST avec Laravel & React

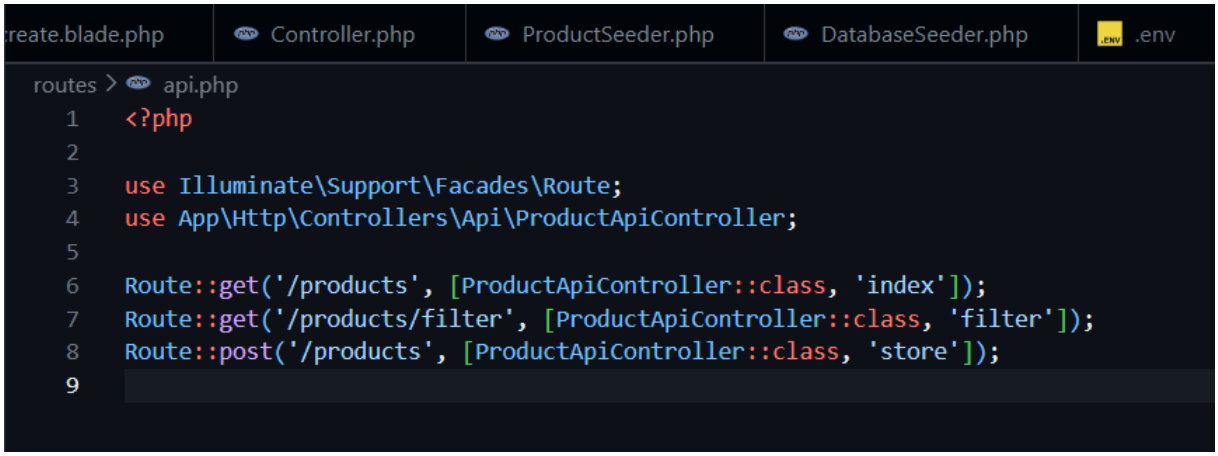
Objectif de l'Atelier

L'objectif de cet atelier est de :

- Créer une **API REST** avec Laravel
- Exposer les données des produits sous forme **JSON**
- Consommer cette API depuis une **application React (Vite)**
- Mettre en place un **filtrage dynamique** des produits
- Héberger l'API Laravel et l'application React sur **Vercel**

Backend Laravel

routes/api.php



app/Http/Controllers/API/ProductApiController.php

reate.blade.php | Controller.php | ProductSeeder.php | DatabaseSeeder.php | .env | ProductApiController.php

app > Http > Controllers > Api > ProductApiController.php > ProductApiController > index

```php
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Models\Product;
use Illuminate\Http\Request;
use Cloudinary\Cloudinary;

class ProductApiController extends Controller
{
    public function index(Request $request)
    {
        $query = Product::query();

        // Filter by category
        if ($request->filled('categorie')) {
            $query->where('categorie', $request->categorie);
        }

        // Filter by name (typing letters)
        if ($request->filled('search')) {
            $query->where('nom', 'LIKE', '%' . $request->search . '%');
        }

        return response()->json(
            $query->select('id', 'nom', 'prix', 'categorie', 'image')
                  ->paginate(100)
        );
    }

    public function store(Request $request)
    {
        $request->validate([
            'nom' => 'required|string|max:255',
            'prix' => 'required|numeric',
            'categorie' => 'required|string|max:255',
            'image' => 'nullable|image|mimes:jpg,jpeg,png,webp|max:2048',
        ]);

        $imagePath = null;

        try {
            if ($request->hasFile('image')) {
                $cloudinary = new Cloudinary([
                    'cloud' => [
                        'cloud_name' => env('CLOUDINARY_CLOUD_NAME'),
                        'api_key'    => env('CLOUDINARY_API_KEY'),
                        'api_secret' => env('CLOUDINARY_API_SECRET'),
                    ],
                ]);

                $uploadResult = $cloudinary->uploadApi()->upload(
                    $request->file('image')->getRealPath(),
                    ['folder' => 'products']
                );

                $imagePath = $uploadResult['secure_url'];
            }
        } catch (\Exception $e) {
            return response()->json([
                'error' => 'Failed to upload image: ' . $e->getMessage(),
            ], 400);
        }
```

```php
        try {
            $product = Product::create([
                'nom' => $request->nom,
                'prix' => $request->prix,
                'categorie' => $request->categorie,
                'image' => $imagePath,
            ]);

            return response()->json([
                'message' => 'Product created successfully',
                'product' => $product,
            ], 201);
        } catch (\Exception $e) {
            return response()->json([
                'error' => 'Failed to create product: ' . $e->getMessage(),
            ], 400);
        }
    }
}
```

app/Models/Product.php

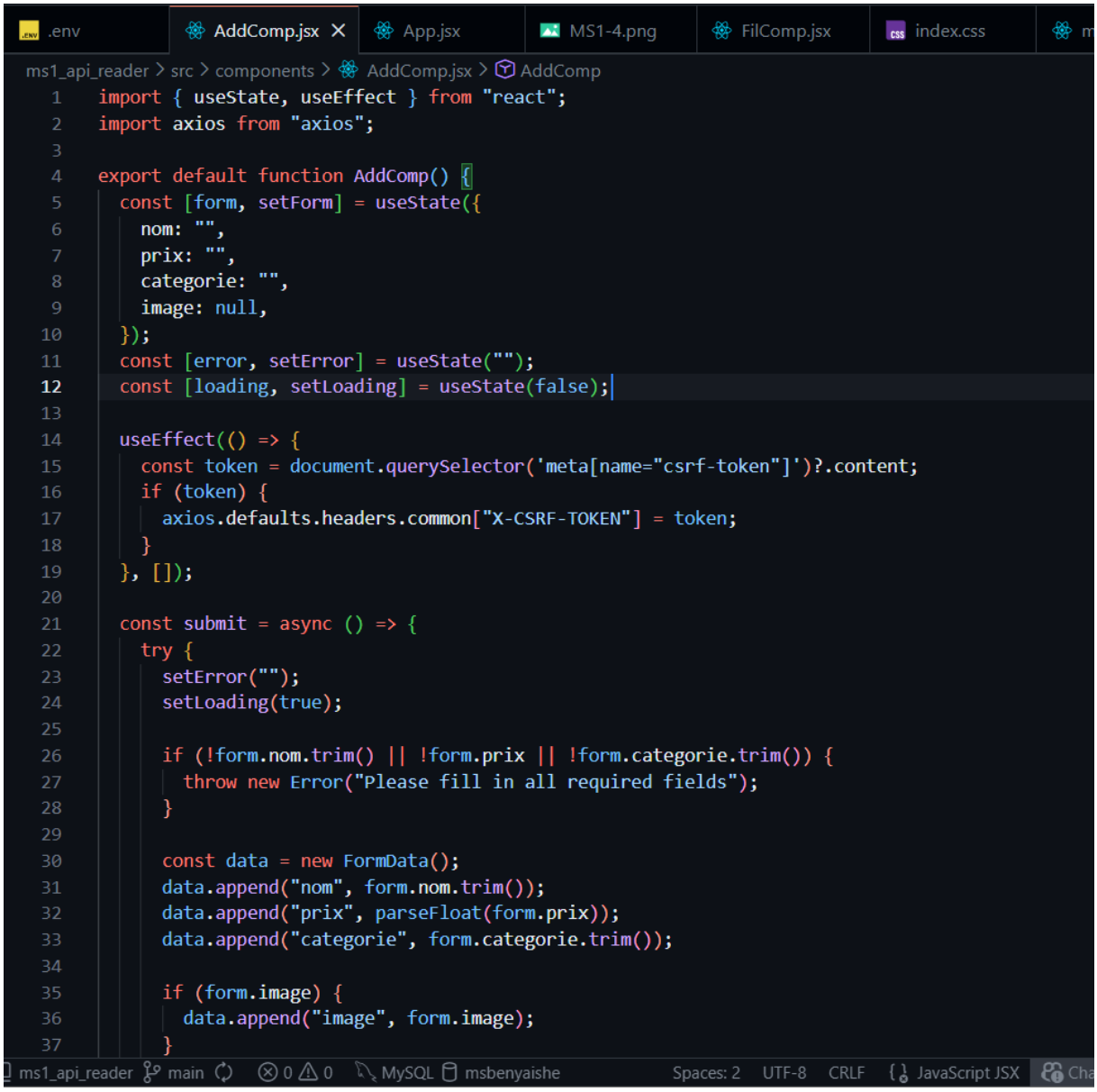
Product.php | flash.blade.php | create.blade.php | Controller.php | ProductSeeder.php

app > Models > Product.php > ...

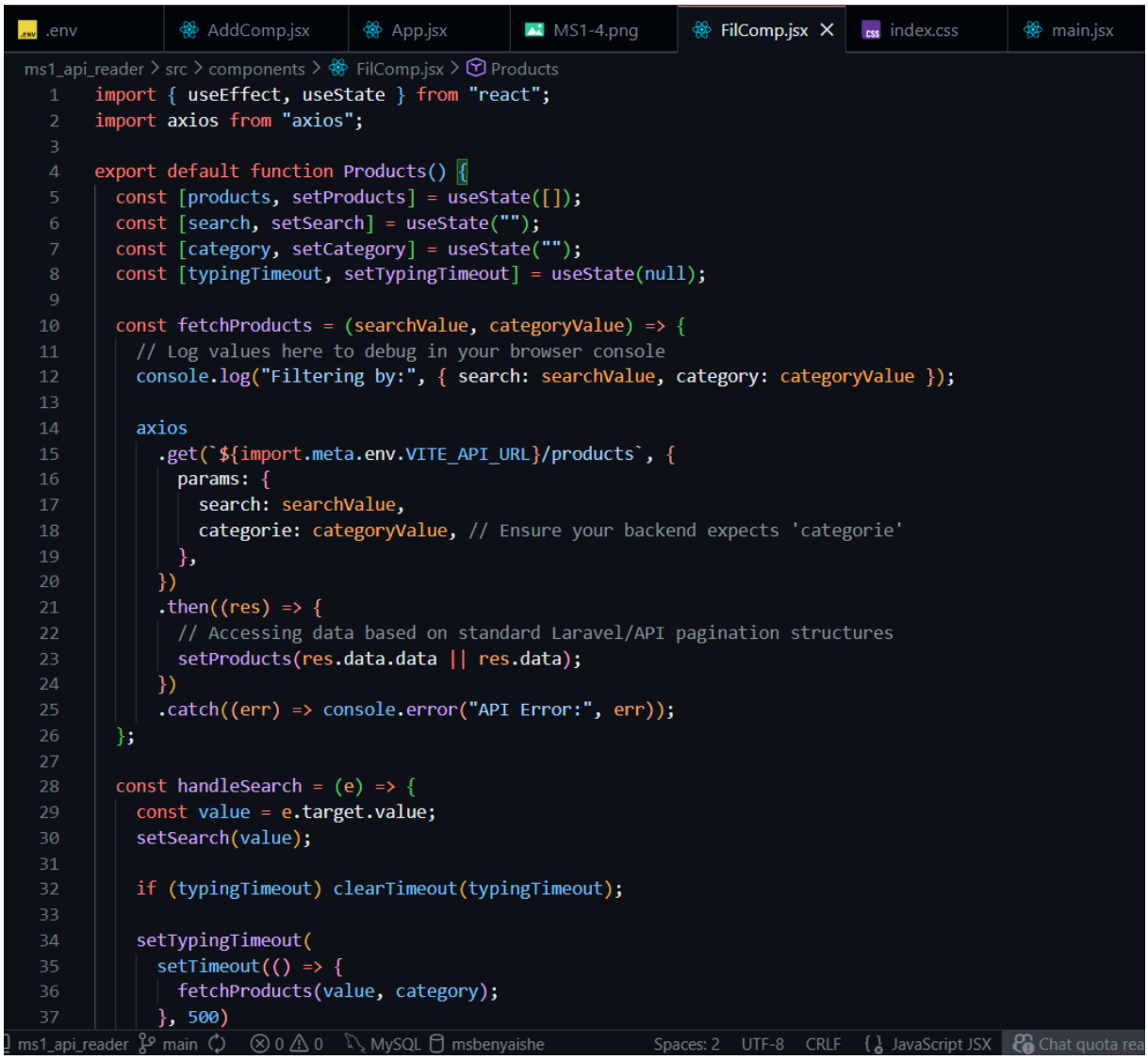
```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    protected $fillable = [
        'nom',
        'prix',
        'categorie',
        'image',
        'solde',
    ];
}

```

Frontend React

src/components/AddComp.jsx



src/components/FilComp.jsx


.env
AddComp.jsx
App.jsx
MS1-4.png
FilComp.jsx
index.css
main.jsx
ms1_api_reader > src > components > FilComp.jsx > Products
import { useEffect, useState } from "react";
import axios from "axios";
export default function Products() {
const [products, setProducts] = useState([]);
const [search, setSearch] = useState("");
const [category, setCategory] = useState("");
const [typingTimeout, setTypingTimeout] = useState(null);
const fetchProducts = (searchValue, categoryValue) => {
// Log values here to debug in your browser console
console.log("Filtering by:", { search: searchValue, category: categoryValue });
axios
.get(`${import.meta.env.VITE_API_URL}/products`, {
params: {
search: searchValue,
categorie: categoryValue, // Ensure your backend expects 'categorie'
},
})
.then((res) => {
// Accessing data based on standard Laravel/API pagination structures
setProducts(res.data.data || res.data);
})
.catch((err) => console.error("API Error:", err));
};
const handleSearch = (e) => {
const value = e.target.value;
setSearch(value);
if (typingTimeout) clearTimeout(typingTimeout);
setTypingTimeout(
setTimeout(() => {
fetchProducts(value, category);
}, 500)
ms1_api_reader
main
0 0
MySQL
msbenyaishe
Spaces: 2
UTF-8
CRLF
JavaScript JSX
Chat quota rea