import { useState } from "react";

import { Card, CardContent } from "@/components/ui/card";

import { Button } from "@/components/ui/button";

import { Input } from "@/components/ui/input";

import { PlusCircle, Search, Trash2, Edit, Globe } from "lucide-react";

import { Select, SelectItem } from "@/components/ui/select";

const initialApps = [

{ name: "ChatGPT", description: "Asistente conversacional avanzado", category: "Chatbots" },

{ name: "DALL·E", description: "Generador de imágenes con IA", category: "Generación de Imágenes" },

{ name: "Whisper", description: "Reconocimiento de voz avanzado", category: "Audio" }

];

const categories = ["Todos", "Chatbots", "Generación de Imágenes", "Audio", "Otros"];

const languages = {

es: "Español", en: "Inglés", fr: "Francés", de: "Alemán", it: "Italiano", pt: "Portugués",

ru: "Ruso", zh: "Chino", ja: "Japonés", ar: "Árabe", ko: "Coreano", nl: "Neerlandés",

sv: "Sueco", pl: "Polaco", tr: "Turco", hi: "Hindi", he: "Hebreo", vi: "Vietnamita"

};

export default function AIHub() {

const [apps, setApps] = useState(initialApps);

const [newApp, setNewApp] = useState({ name: "", description: "", category: "Otros" });

const [search, setSearch] = useState("");

const [filter, setFilter] = useState("Todos");

const [editingIndex, setEditingIndex] = useState(null);

const [language, setLanguage] = useState("es");

const addOrEditApp = () => {

if (newApp.name && newApp.description) {

if (editingIndex !== null) {

const updatedApps = [...apps];

updatedApps[editingIndex] = newApp;

setApps(updatedApps);

setEditingIndex(null);

} else {

setApps([...apps, newApp]);

}

setNewApp({ name: "", description: "", category: "Otros" });

}

};

const deleteApp = (index) => {

setApps(apps.filter((\_, i) => i !== index));

};

const editApp = (index) => {

setNewApp(apps[index]);

setEditingIndex(index);

};

const translateText = async (text) => {

const response = await fetch(`https://api.mymemory.translated.net/get?q=${text}&langpair=es|${language}`);

const data = await response.json();

return data.responseData.translatedText;

};

const translateApps = async () => {

const translatedApps = await Promise.all(

apps.map(async (app) => ({

name: await translateText(app.name),

description: await translateText(app.description),

category: app.category

}))

);

setApps(translatedApps);

};

const filteredApps = apps.filter(

(app) =>

(filter === "Todos" || app.category === filter) &&

app.name.toLowerCase().includes(search.toLowerCase())

);

return (

<div className="p-6 space-y-4">

<h1 className="text-xl font-bold">Centro de Aplicaciones de IA (Código Abierto)</h1>

<p className="text-sm text-gray-600">Este proyecto es de código abierto bajo la licencia MIT.</p>

<div className="flex gap-2">

<Input

placeholder="Buscar aplicación..."

value={search}

onChange={(e) => setSearch(e.target.value)}

/>

<Select value={filter} onChange={(e) => setFilter(e.target.value)}>

{categories.map((cat) => (

<SelectItem key={cat} value={cat}>{cat}</SelectItem>

))}

</Select>

<Select value={language} onChange={(e) => setLanguage(e.target.value)}>

{Object.entries(languages).map(([key, label]) => (

<SelectItem key={key} value={key}>{label}</SelectItem>

))}

</Select>

<Button onClick={translateApps}>

<Globe className="w-5 h-5" />

</Button>

</div>

<div className="grid grid-cols-1 md:grid-cols-3 gap-4">

{filteredApps.map((app, index) => (

<Card key={index} className="p-4 relative">

<CardContent>

<h2 className="text-lg font-semibold">{app.name}</h2>

<p className="text-sm text-gray-600">{app.description}</p>

<p className="text-xs text-gray-500">Categoría: {app.category}</p>

<div className="absolute top-2 right-2 flex gap-2">

<Button variant="outline" size="icon" onClick={() => editApp(index)}>

<Edit className="w-4 h-4 text-blue-500" />

</Button>

<Button variant="outline" size="icon" onClick={() => deleteApp(index)}>

<Trash2 className="w-4 h-4 text-red-500" />

</Button>

</div>

</CardContent>

</Card>

))}

</div>

<div className="flex gap-2">

<Input

placeholder="Nombre de la aplicación"

value={newApp.name}

onChange={(e) => setNewApp({ ...newApp, name: e.target.value })}

/>

<Input

placeholder="Descripción"

value={newApp.description}

onChange={(e) => setNewApp({ ...newApp, description: e.target.value })}

/>

<Select value={newApp.category} onChange={(e) => setNewApp({ ...newApp, category: e.target.value })}>

{categories.slice(1).map((cat) => (

<SelectItem key={cat} value={cat}>{cat}</SelectItem>

))}

</Select>

<Button onClick={addOrEditApp}>

<PlusCircle className="w-5 h-5" />

</Button>

</div>

</div>

);

}