import React, { useState } from "react";

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

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

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

import { Textarea } from "@/components/ui/textarea";

import { motion } from "framer-motion";

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

import { saveAs } from "file-saver";

import { PieChart, Pie, Cell, Tooltip, Legend, ResponsiveContainer } from "recharts";

import { toast } from "sonner";

const initialIdeas = [

{

title: "Cimento Verde Apodi",

description: "Redução de CO₂ com uso de pozolanas e argilas calcinadas.",

category: "Sustentabilidade",

date: "2025-05-07",

author: "Iara"

},

{

title: "Logística Inteligente",

description: "Roteirização otimizada com IA para entregas mais eficientes.",

category: "Logística",

date: "2025-05-06",

author: "Carlos"

}

];

const COLORS = ["#8884d8", "#82ca9d", "#ffc658", "#ff7f50", "#8dd1e1"];

export default function MassaCinzenta() {

const [ideas, setIdeas] = useState(initialIdeas);

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

const [keyword, setKeyword] = useState("");

const [title, setTitle] = useState("");

const [description, setDescription] = useState("");

const [category, setCategory] = useState("");

const [date, setDate] = useState("");

const [author, setAuthor] = useState("");

const [newCategory, setNewCategory] = useState("");

const handleSubmit = (e) => {

e.preventDefault();

if (title && description && category) {

const ideaDate = date || new Date().toISOString().split("T")[0];

setIdeas([...ideas, { title, description, category, date: ideaDate, author }]);

setTitle("");

setDescription("");

setCategory("");

setDate("");

setAuthor("");

toast.success("Ideia cadastrada com sucesso!");

} else {

toast.error("Por favor, preencha todos os campos obrigatórios.");

}

};

const allCategories = [

...new Set(["Sustentabilidade", "Logística", "Digitalização", "Produto", ...ideas.map((i) => i.category)])

];

const filteredIdeas = ideas.filter((idea) => {

const categoryMatch = filter === "Todas" || idea.category === filter;

const keywordMatch = keyword === "" || idea.title.toLowerCase().includes(keyword.toLowerCase()) || idea.description.toLowerCase().includes(keyword.toLowerCase());

return categoryMatch && keywordMatch;

});

const totalIdeas = ideas.length;

const ideasByCategory = ideas.reduce((acc, curr) => {

acc[curr.category] = (acc[curr.category] || 0) + 1;

return acc;

}, {});

const pieData = Object.entries(ideasByCategory).map(([name, value]) => ({ name, value }));

const exportToCSV = () => {

const csvContent = [

["Título", "Descrição", "Categoria", "Data", "Autor"],

...ideas.map(({ title, description, category, date, author }) => [title, description, category, date, author])

].map(row => row.map(field => `"${field?.replace(/"/g, '""')}"`).join(",")).join("\n");

const blob = new Blob([csvContent], { type: "text/csv;charset=utf-8;" });

saveAs(blob, "massa\_cinzenta\_ideias.csv");

};

return (

<div className="min-h-screen bg-gray-100 p-8">

<motion.h1

className="text-4xl font-bold mb-6 text-center text-gray-800"

initial={{ opacity: 0, y: -20 }}

animate={{ opacity: 1, y: 0 }}

>

Massa Cinzenta – Ideias que Constroem o Futuro

</motion.h1>

<div className="max-w-3xl mx-auto mb-8 bg-white p-6 rounded-2xl shadow-md">

<h2 className="text-2xl font-semibold text-gray-800 mb-4">Painel de Indicadores</h2>

<p className="text-gray-700 mb-2">Total de ideias: <strong>{totalIdeas}</strong></p>

<div className="grid grid-cols-2 gap-4">

{Object.entries(ideasByCategory).map(([cat, count], index) => (

<div key={index} className="text-gray-600">

{cat}: <strong>{count}</strong>

</div>

))}

</div>

<Button className="mt-4" onClick={exportToCSV}>Exportar CSV</Button>

<div className="h-72 mt-6">

<ResponsiveContainer width="100%" height="100%">

<PieChart>

<Pie

data={pieData}

dataKey="value"

nameKey="name"

cx="50%"

cy="50%"

outerRadius={80}

label

>

{pieData.map((entry, index) => (

<Cell key={`cell-${index}`} fill={COLORS[index % COLORS.length]} />

))}

</Pie>

<Tooltip />

<Legend />

</PieChart>

</ResponsiveContainer>

</div>

</div>

<div className="max-w-3xl mx-auto mb-6 space-y-4">

<Input

type="text"

placeholder="Buscar por palavra-chave..."

value={keyword}

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

/>

<Select onValueChange={(val) => setFilter(val)}>

<SelectTrigger className="w-full">

<SelectValue placeholder="Filtrar por categoria" />

</SelectTrigger>

<SelectContent>

<SelectItem value="Todas">Todas</SelectItem>

{allCategories.map((cat, idx) => (

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

))}

</SelectContent>

</Select>

</div>

<div className="max-w-3xl mx-auto space-y-4">

{filteredIdeas.map((idea, index) => (

<Card key={index} className="shadow-lg rounded-2xl">

<CardContent className="p-4">

<h2 className="text-xl font-semibold text-gray-700">{idea.title}</h2>

<p className="text-sm text-gray-500 mb-1">Categoria: {idea.category}</p>

<p className="text-sm text-gray-400">Data: {idea.date}</p>

{idea.author && <p className="text-sm text-gray-500">Autor(a): {idea.author}</p>}

<p className="text-gray-600 mt-2">{idea.description}</p>

</CardContent>

</Card>

))}

</div>

<div className="max-w-3xl mx-auto mt-10 p-6 bg-white rounded-2xl shadow-md">

<h2 className="text-2xl font-semibold mb-4 text-gray-800">Compartilhar Nova Ideia</h2>

<form className="space-y-4" onSubmit={handleSubmit}>

<Input type="text" placeholder="Título da Ideia" className="w-full" value={title} onChange={(e) => setTitle(e.target.value)} />

<Textarea placeholder="Descrição da ideia" className="w-full min-h-[120px]" value={description} onChange={(e) => setDescription(e.target.value)} />

<Select onValueChange={(val) => setCategory(val)}>

<SelectTrigger className="w-full">

<SelectValue placeholder="Categoria" />

</SelectTrigger>

<SelectContent>

{allCategories.map((cat, idx) => (

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

))}

</SelectContent>

</Select>

<Input type="text" placeholder="Nome do autor (opcional)" className="w-full" value={author} onChange={(e) => setAuthor(e.target.value)} />

<Input type="date" className="w-full" value={date} onChange={(e) => setDate(e.target.value)} />

<Button type="submit">Enviar</Button>

</form>

</div>

</div>

);

}