import React from "react";

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

import { Badge } from "@/components/ui/badge";

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

import { Tooltip, TooltipContent, TooltipProvider, TooltipTrigger } from "@/components/ui/tooltip";

import {

AlertDialog,

AlertDialogAction,

AlertDialogCancel,

AlertDialogContent,

AlertDialogDescription,

AlertDialogFooter,

AlertDialogHeader,

AlertDialogTitle,

AlertDialogTrigger,

} from "@/components/ui/alert-dialog";

import {

DropdownMenu,

DropdownMenuContent,

DropdownMenuItem,

DropdownMenuTrigger,

} from "@/components/ui/dropdown-menu";

import { FileText, MessageSquare, Calendar, AlertCircle, ChevronRight, MoreVertical, Trash2 } from "lucide-react";

import { format } from "date-fns";

import { Link } from "react-router-dom";

import { createPageUrl } from "@/utils";

const statusColors = {

active: "bg-green-100 text-green-800 border-green-200",

research: "bg-blue-100 text-blue-800 border-blue-200",

drafting: "bg-yellow-100 text-yellow-800 border-yellow-200",

filing: "bg-purple-100 text-purple-800 border-purple-200",

completed: "bg-gray-100 text-gray-800 border-gray-200",

closed: "bg-slate-100 text-slate-800 border-slate-200"

};

const priorityColors = {

low: "bg-blue-100 text-blue-800",

medium: "bg-yellow-100 text-yellow-800",

high: "bg-orange-100 text-orange-800",

urgent: "bg-red-100 text-red-800"

};

const caseTypeLabels = {

personal\_injury: "Personal Injury",

contract\_dispute: "Contract Dispute",

family\_law: "Family Law",

criminal\_defense: "Criminal Defense",

employment: "Employment",

real\_estate: "Real Estate",

corporate: "Corporate",

intellectual\_property: "IP",

immigration: "Immigration",

other: "Other"

};

export default function CaseCard({ case: legalCase, documents, messages, onDelete }) {

const isUrgent = legalCase.deadline && new Date(legalCase.deadline) <= new Date(Date.now() + 7 \* 24 \* 60 \* 60 \* 1000);

const handleDelete = async () => {

if (onDelete) {

await onDelete(legalCase.id);

}

};

// Helper to safely format dates and avoid timezone issues

const formatDate = (dateString) => {

if (!dateString) return null;

return format(new Date(dateString + 'T00:00:00'), "MMM d, yyyy");

};

const formattedDeadline = formatDate(legalCase.deadline);

return (

<Card className="group hover:shadow-xl transition-all duration-300 bg-white/80 backdrop-blur-sm border-0 shadow-lg overflow-hidden">

<CardHeader className="pb-3">

<div className="flex items-start justify-between">

<div className="flex-1">

<div className="flex items-start justify-between mb-2">

<CardTitle className="text-lg font-semibold text-[var(--navy)] line-clamp-2 flex-1">

{legalCase.title}

</CardTitle>

<div className="flex items-center gap-2 ml-2">

{isUrgent && (

<TooltipProvider>

<Tooltip>

<TooltipTrigger asChild>

<div className="cursor-help">

<AlertCircle className="w-5 h-5 text-red-500" />

</div>

</TooltipTrigger>

<TooltipContent>

<p>Urgent: Deadline is within 7 days</p>

<p className="text-xs text-slate-400">

Due: {formattedDeadline}

</p>

</TooltipContent>

</Tooltip>

</TooltipProvider>

)}

<DropdownMenu>

<DropdownMenuTrigger asChild>

<Button variant="ghost" size="icon" className="h-8 w-8 opacity-0 group-hover:opacity-100 transition-opacity">

<MoreVertical className="w-4 h-4" />

</Button>

</DropdownMenuTrigger>

<DropdownMenuContent align="end">

<AlertDialog>

<AlertDialogTrigger asChild>

<DropdownMenuItem onSelect={(e) => e.preventDefault()} className="text-red-600 focus:text-red-600 cursor-pointer">

<Trash2 className="w-4 h-4 mr-2" />

Delete Case

</DropdownMenuItem>

</AlertDialogTrigger>

<AlertDialogContent>

<AlertDialogHeader>

<AlertDialogTitle>Delete Case</AlertDialogTitle>

<AlertDialogDescription>

Are you sure you want to delete "{legalCase.title}"? This action cannot be undone and will permanently delete the case and all associated documents and messages.

</AlertDialogDescription>

</AlertDialogHeader>

<AlertDialogFooter>

<AlertDialogCancel>Cancel</AlertDialogCancel>

<AlertDialogAction onClick={handleDelete} className="bg-red-600 hover:bg-red-700">

Delete Case

</AlertDialogAction>

</AlertDialogFooter>

</AlertDialogContent>

</AlertDialog>

</DropdownMenuContent>

</DropdownMenu>

</div>

</div>

<div className="flex flex-wrap gap-2">

<Badge variant="outline" className={statusColors[legalCase.status]}>

{legalCase.status?.replace(/\_/g, ' ')}

</Badge>

<Badge variant="outline" className="text-xs">

{caseTypeLabels[legalCase.case\_type] || legalCase.case\_type}

</Badge>

{legalCase.priority !== 'medium' && (

<Badge className={priorityColors[legalCase.priority]}>

{legalCase.priority}

</Badge>

)}

</div>

</div>

</div>

</CardHeader>

<CardContent className="space-y-4">

{legalCase.description && (

<p className="text-sm text-slate-600 line-clamp-2">

{legalCase.description}

</p>

)}

{legalCase.jurisdiction && (

<div className="text-sm text-slate-500">

📍 {legalCase.jurisdiction}

</div>

)}

<div className="flex items-center justify-between text-sm text-slate-500">

<div className="flex items-center gap-4">

<div className="flex items-center gap-1">

<FileText className="w-4 h-4" />

<span>{documents.length}</span>

</div>

<div className="flex items-center gap-1">

<MessageSquare className="w-4 h-4" />

<span>{messages.length}</span>

</div>

</div>

{formattedDeadline && (

<div className={`flex items-center gap-1 ${isUrgent ? 'text-red-600 font-medium' : ''}`}>

<Calendar className="w-4 h-4" />

<span>{format(new Date(legalCase.deadline + 'T00:00:00'), "MMM d")}</span>

</div>

)}

</div>

<div className="flex items-center justify-between pt-2 border-t border-slate-100">

<div className="text-xs text-slate-400">

Created {format(new Date(legalCase.created\_date), "MMM d, yyyy")}

</div>

<Link to={createPageUrl(`Chat?caseId=${legalCase.id}`)}>

<Button

size="sm"

className="bg-gradient-to-r from-[var(--navy)] to-[var(--navy-light)] hover:shadow-lg transition-all duration-300 group-hover:translate-x-1"

>

Open Chat

<ChevronRight className="w-4 h-4 ml-1" />

</Button>

</Link>

</div>

</CardContent>

</Card>

);

}