import { useState } from "react";

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

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

import { Label } from "@/components/ui/label";

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

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

import { Dialog, DialogContent, DialogDescription, DialogHeader, DialogTitle } from "@/components/ui/dialog";

import { supabase } from "@/lib/supabase";

import { useToast } from "@/hooks/use-toast";

interface CreateProjectDialogProps {

open: boolean;

onOpenChange: (open: boolean) => void;

onProjectCreated: () => void;

}

export const CreateProjectDialog = ({ open, onOpenChange, onProjectCreated }: CreateProjectDialogProps) => {

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

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

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

const [maxCollaborators, setMaxCollaborators] = useState("3");

const [loading, setLoading] = useState(false);

const { toast } = useToast();

const handleSubmit = async (e: React.FormEvent) => {

e.preventDefault();

setLoading(true);

try {

const { data: { user } } = await supabase.auth.getUser();

if (!user) throw new Error("Not authenticated");

const { error } = await supabase

.from('projects')

.insert({

title,

description,

category,

max\_collaborators: parseInt(maxCollaborators),

creator\_id: user.id,

status: 'open'

});

if (error) throw error;

toast({

title: "Success!",

description: "Your project has been created.",

});

setTitle("");

setDescription("");

setCategory("");

setMaxCollaborators("3");

onOpenChange(false);

onProjectCreated();

} catch (error: any) {

toast({

title: "Error",

description: error.message,

variant: "destructive",

});

} finally {

setLoading(false);

}

};

return (

<Dialog open={open} onOpenChange={onOpenChange}>

<DialogContent className="max-w-md">

<DialogHeader>

<DialogTitle>Create New Project</DialogTitle>

<DialogDescription>

Start a creative collaboration and invite others to join

</DialogDescription>

</DialogHeader>

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

<div className="space-y-2">

<Label htmlFor="title">Project Title</Label>

<Input

id="title"

value={title}

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

placeholder="Enter project title"

required

/>

</div>

<div className="space-y-2">

<Label htmlFor="description">Description</Label>

<Textarea

id="description"

value={description}

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

placeholder="Describe your project idea..."

rows={3}

/>

</div>

<div className="space-y-2">

<Label htmlFor="category">Category</Label>

<Select value={category} onValueChange={setCategory} required>

<SelectTrigger>

<SelectValue placeholder="Select category" />

</SelectTrigger>

<SelectContent>

<SelectItem value="writing">Writing & Storytelling</SelectItem>

<SelectItem value="design">Art & Design</SelectItem>

<SelectItem value="coding">Coding & Development</SelectItem>

<SelectItem value="music">Music & Audio</SelectItem>

<SelectItem value="other">Other Creative</SelectItem>

</SelectContent>

</Select>

</div>

<div className="space-y-2">

<Label htmlFor="maxCollaborators">Max Collaborators</Label>

<Select value={maxCollaborators} onValueChange={setMaxCollaborators}>

<SelectTrigger>

<SelectValue />

</SelectTrigger>

<SelectContent>

<SelectItem value="2">2 people</SelectItem>

<SelectItem value="3">3 people</SelectItem>

<SelectItem value="4">4 people</SelectItem>

<SelectItem value="5">5 people</SelectItem>

</SelectContent>

</Select>

</div>

<div className="flex gap-3 pt-4">

<Button type="button" variant="outline" onClick={() => onOpenChange(false)} className="flex-1">

Cancel

</Button>

<Button type="submit" variant="creative" disabled={loading} className="flex-1">

{loading ? "Creating..." : "Create Project"}

</Button>

</div>

</form>

</DialogContent>

</Dialog>

);

};