import { useState } from "react";

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

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

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

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

import { Alert, AlertDescription } from "@/components/ui/alert";

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

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

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

interface AuthFormProps {

mode: "login" | "signup";

}

export const AuthForm = ({ mode }: AuthFormProps) => {

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const [fullName, setFullName] = useState("");

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

const [error, setError] = useState("");

const navigate = useNavigate();

const { toast } = useToast();

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

e.preventDefault();

setLoading(true);

setError("");

try {

if (mode === "signup") {

const { error } = await supabase.auth.signUp({

email,

password,

options: {

data: {

full\_name: fullName,

},

},

});

if (error) throw error;

toast({

title: "Success!",

description: "Check your email to confirm your account.",

});

} else {

const { error } = await supabase.auth.signInWithPassword({

email,

password,

});

if (error) throw error;

navigate("/dashboard");

}

} catch (error: any) {

setError(error.message);

} finally {

setLoading(false);

}

};

return (

<div className="min-h-screen bg-gradient-warm flex items-center justify-center p-4">

<Card className="w-full max-w-md shadow-creative">

<CardHeader className="space-y-1 text-center">

<CardTitle className="text-2xl font-bold">

{mode === "login" ? "Welcome Back" : "Join CreativeMinds"}

</CardTitle>

<CardDescription>

{mode === "login"

? "Sign in to your creative space"

: "Start your creative journey today"

}

</CardDescription>

</CardHeader>

<CardContent>

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

{mode === "signup" && (

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

<Label htmlFor="fullName">Full Name</Label>

<Input

id="fullName"

type="text"

value={fullName}

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

required

/>

</div>

)}

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

<Label htmlFor="email">Email</Label>

<Input

id="email"

type="email"

value={email}

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

required

/>

</div>

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

<Label htmlFor="password">Password</Label>

<Input

id="password"

type="password"

value={password}

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

required

/>

</div>

{error && (

<Alert>

<AlertDescription>{error}</AlertDescription>

</Alert>

)}

<Button

type="submit"

className="w-full"

variant="creative"

disabled={loading}

>

{loading ? "Please wait..." : mode === "login" ? "Sign In" : "Sign Up"}

</Button>

</form>

<div className="mt-4 text-center text-sm">

{mode === "login" ? (

<>

Don't have an account?{" "}

<Button variant="link" onClick={() => navigate("/signup")} className="p-0">

Sign up

</Button>

</>

) : (

<>

Already have an account?{" "}

<Button variant="link" onClick={() => navigate("/login")} className="p-0">

Sign in

</Button>

</>

)}

</div>

</CardContent>

</Card>

</div>

);

};