

```
import React, { useState } from "react";
import { base44 } from "@/api/base44Client";
import { Button } from "@/components/ui/button";
import { Card, CardContent, CardHeader, CardTitle } from
"@/components/ui/card";
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 {
  Phone,
  Mail,
  MessageCircle,
  MapPin,
  Clock,
  Send,
  CheckCircle,
  Shield,
  Heart,
  HelpCircle
} from "lucide-react";

export default function Contact() {
  const [formData, setFormData] = useState({
    name: "",
    email: "",
    subject: "",
    message: "",
    inquiry_type: "general"
  });
  const [submitted, setSubmitted] = useState(false);
  const [isSubmitting, setIsSubmitting] = useState(false);

  const handleSubmit = async (e) => {
    e.preventDefault();
    setIsSubmitting(true);

    try {
      await base44.integrations.Core.SendEmail({
        to: "support@cloud7wellness.com",
        subject: `Contact Form: ${formData.subject}
(${formData.inquiry_type})`,
        body: `Name: ${formData.name}\nEmail: ${formData.email}\nSubject: ${formData.subject}\nMessage: ${formData.message}`
      });
      setSubmitted(true);
      setIsSubmitting(false);
    } catch (error) {
      console.error(error);
      setIsSubmitting(false);
    }
  };
}
```

```
        body: `

Name: ${formData.name}
Email: ${formData.email}
Type: ${formData.inquiry_type}

Message:
${formData.message}
`)

setSubmitted(true);
setFormData({
  name: '',
  email: '',
  subject: '',
  message: '',
  inquiry_type: "general"
});
} catch (error) {
  console.error("Error sending message:", error);
}

setIsSubmitting(false);
};

const contactMethods = [
{
  icon: Phone,
  title: "Call Us",
  info: "+44 20 1234 5678",
  description: "Mon-Fri, 9am-5pm GMT",
  color: "from-purple-400 to-purple-600"
},
{
  icon: Mail,
  title: "Email Us",
  info: "support@cloud7wellness.com",
  description: "We'll respond within 24 hours",
  color: "from-teal-400 to-teal-600"
},
{
  icon: MessageCircle,
  title: "Live Chat",
}
```

```

        info: "Available on platform",
        description: "Instant support for members",
        color: "from-blue-400 to-blue-600"
    },
    {
        icon: MapPin,
        title: "Visit Us",
        info: "123 Wellness Street, London",
        description: "By appointment only",
        color: "from-indigo-400 to-indigo-600"
    }
];

const faqItems = [
{
    question: "How do I book my first session?",
    answer: "Visit our 'Book a Session' page and select your preferred therapist type, date, and format (online or in-person)."
},
{
    question: "Is my data secure and confidential?",
    answer: "Yes! All your data is encrypted and stored securely. We follow strict GDPR compliance and never share your information."
},
{
    question: "What subscription plan should I choose?",
    answer: "Start with our Free plan to explore. If you need regular professional support, Basic or Premium plans offer more sessions."
},
{
    question: "Can I remain anonymous?",
    answer: "Yes, we offer anonymous session options. Your privacy and comfort are our top priorities."
}
];

return (
<div className="py-12 px-4">
    <div className="max-w-6xl mx-auto space-y-12">
        {/* Header */}
        <div className="text-center space-y-4">

```

```
<div className="inline-flex items-center justify-center w-16  
h-16 bg-gradient-to-br from-purple-400 to-teal-400 rounded-full  
shadow-lg mb-4">  
    <Heart className="w-8 h-8 text-white" />  
</div>  
<h1 className="text-5xl font-bold bg-gradient-to-r  
from-purple-600 to-teal-600 bg-clip-text text-transparent">  
    Get in Touch  
</h1>  
<p className="text-xl text-gray-600 max-w-2xl mx-auto">  
    We're here to support you. Reach out with any questions,  
concerns, or feedback.  
</p>  
</div>  
  
/* Contact Methods */  
<div className="grid md:grid-cols-2 lg:grid-cols-4 gap-6">  
    {contactMethods.map((method, index) => (  
        <Card key={index} className="border-none shadow-lg  
        hover:shadow-xl transition-all duration-300">  
            <CardContent className="p-6 text-center space-y-4">  
                <div className={`w-14 h-14 bg-gradient-to-br  
                ${method.color} rounded-2xl flex items-center justify-center shadow-lg  
                mx-auto`}>  
                    <method.icon className="w-7 h-7 text-white" />  
                </div>  
                <div>  
                    <h3 className="font-bold text-gray-800  
                    mb-1">{method.title}</h3>  
                    <p className="text-sm font-semibold text-purple-600  
                    mb-1">{method.info}</p>  
                    <p className="text-xs  
                    text-gray-500">{method.description}</p>  
                </div>  
            </CardContent>  
        </Card>  
    ))}  
</div>  
  
/* Contact Form & Info */  
<div className="grid lg:grid-cols-3 gap-8">  
    /* Form */  
<div className="lg:col-span-2">
```

```
<Card className="border-none shadow-xl">
  <CardHeader>
    <CardTitle className="flex items-center gap-2">
      <Send className="w-5 h-5" />
      Send Us a Message
    </CardTitle>
  </CardHeader>
  <CardContent>
    {submitted ? (
      <div className="text-center py-12 space-y-4">
        <div className="w-16 h-16 bg-green-100 rounded-full flex items-center justify-center mx-auto">
          <CheckCircle className="w-10 h-10 text-green-600" />
        </div>
        <h3 className="text-2xl font-bold text-gray-800">Message Sent!</h3>
        <p className="text-gray-600">
          Thank you for reaching out. We'll get back to you
          within 24 hours.
        </p>
        <Button
          onClick={() => setSubmitted(false)}
          variant="outline"
          className="mt-4"
        >
          Send Another Message
        </Button>
      </div>
    ) : (
      <form onSubmit={handleSubmit} className="space-y-6">
```