

File: client/src/pages/Compatibility.tsx

Type: tsx | Size: 12452 bytes

```
import React, { useState } from 'react';
import { motion } from 'framer-motion';
import { useToast } from '@/hooks/use-toast';
import { useAds } from '@/hooks/use-ads';
import { useForm } from 'react-hook-form';
import { z } from 'zod';
import { zodResolver } from '@hookform/resolvers/zod';
import HeartLoader from '@/components/HeartLoader';
import { getRandomLoaderMessage } from '@/lib/loaderMessages';

const formSchema = z.object({
  partnerName: z.string().min(1, "Partner's name is required"),
  relationship: z.string().min(1, "Please select a relationship type"),
  email: z.string().email("Please enter a valid email").or(z.literal('')),
  phone: z.string().min(5, "Please enter a valid phone number").or(z.literal('')),
});

type FormData = z.infer<typeof formSchema>;

export default function Compatibility() {
  const { toast } = useToast();
  const { trackAction } = useAds();
  const [step, setStep] = useState(1);
  const [isLoading, setIsLoading] = useState(false);
  const [inviteSent, setInviteSent] = useState(false);

  const relationshipTypes = [
    { id: 'dating', label: 'Dating', icon: 'fa-heart' },
    { id: 'situationship', label: 'Situationship', icon: 'fa-question-circle' },
    { id: 'crush', label: 'Crush/Potential', icon: 'fa-seedling' },
    { id: 'committed', label: 'Committed', icon: 'fa-hands-holding-circle' },
    { id: 'married', label: 'Married', icon: 'fa-rings-wedding' },
    { id: 'friendship', label: 'Friendship', icon: 'fa-user-group' },
  ];

  const { register, handleSubmit, watch, formState: { errors } } = useForm<FormData>({
    resolver: zodResolver(formSchema),
    defaultValues: {
      partnerName: '',
      relationship: 'dating',
      email: '',
      phone: '',
    }
  });

  const selectedRelationship = watch('relationship');

  const onSubmit = async (data: FormData) => {
    try {
      setIsLoading(true);

      // For demo purposes, we'll just simulate success after a short delay
      setTimeout(() => {
        setInviteSent(true);
        setIsLoading(false);

        // Track action to potentially show an interstitial ad
        trackAction();

        toast({
          title: "Invitation Sent",
          description: `We've sent an invitation to join your compatibility test.`,
        });
      }, 1500);
    } catch (error) {
      toast({
        title: "Error",
        description: "Failed to send invitation. Please try again.",
        variant: "destructive",
      });
    }
  };
}
```

```

});  
setIsLoading(false);  
};  
  
return (  
    <section className="px-4 py-4">  
        <div className="text-center mb-8">  
            <h2 className="font-serif text-3xl text-mystical-pink">Compatibility Test</h2>  
            <p className="text-mystical-lavender">Discover your cosmic connection</p>  
        </div>  
  
        {!inviteSent ? (  
            <div className="bg-mystical-gradient backdrop-blur-md rounded-xl p-5 mb-8">  
                <div className="flex justify-between items-center mb-6">  
                    <h3 className="font-serif text-xl text-mystical-light">Test Invitation</h3>  
                    <div className="bg-white/20 px-2 py-1 rounded text-xs text-white">  
                        Step {step} of 2  
                    </div>  
                </div>  
            </div>  
  
            <form onSubmit={handleSubmit(onSubmit)}>  
                {step === 1 ? (  
                    <div className="space-y-4">  
                        <div>  
                            <label className="block text-mystical-light mb-2 font-medium">Partner's Name</label>  
                            <input  
                                type="text"  
                                className={`w-full bg-mystical-dark border ${errors.partnerName ? 'border-red-400' : 'border-transparent'}`}  
                                placeholder="Enter their name"  
                                {...register('partnerName')}  
                            />  
                            {errors.partnerName && (  
                                <p className="text-xs text-red-400 mt-1">{errors.partnerName.message}</p>  
                            )}  
                        </div>  
  
                        <div>  
                            <label className="block text-mystical-light mb-2 font-medium">Relationship Type</label>  
                            <div className="grid grid-cols-2 gap-3">  
                                {relationshipTypes.map(type => (  
                                    <label  
                                        key={type.id}  
                                        className={`${`flex items-center p-3 rounded-lg cursor-pointer transition-all ${selectedRelationship === type.id ? 'bg-mystical-purple/30 border border-mystical-purple' : 'bg-white/20 border border-transparent'}`}`}  
                                    >  
                                        <input  
                                            type="radio"  
                                            value={type.id}  
                                            className="hidden"  
                                            {...register('relationship')}  
                                        />  
                                        <div className="w-8 h-8 rounded-full bg-mystical-dark/80 border border-mystical-purple/30 flex items-center justify-center gap-1">  
                                            <i className={`${`fas ${type.icon} text-mystical-pink`}`}></i>  
                                            <div>  
                                                <span className="text-white text-sm">{type.label}</span>  
                                            </div>  
                                        </div>  
                                    </label>  
                                ))}  
                            </div>  
                        </div>  
                    </div>  
                ) : (  
                    <div>  
                        <motion.button  
                            type="button"  
                            className="w-full bg-mystical-purple hover:bg-mystical-lavender text-white font-medium py-3 px-4 rounded-full gap-2">  
                            {step === 1 ? (  
                                <span>Get Started</span>  
                            ) : (  
                                <span>Next Step</span>  
                            )}  
                            <span>Get Started</span>  
                        </motion.button>  
                    </div>  
                )}  
            </form>  
        ) : (  
            <div>  
                <h3>Compatibility Test</h3>  
                <p>Discover your cosmic connection</p>  
            </div>  
        )}  
    </section>

```

```

        </div>
    ) : (
      <div className="space-y-4">
        <p className="text-mystical-light mb-4">How would you like to invite <span className="font-med...</span></p>
        <div>
          <label className="block text-mystical-light mb-2 font-medium">Email Address</label>
          <input
            type="email"
            className={`w-full bg-mystical-dark border ${errors.email ? 'border-red-400' : 'border-mys...`}
            placeholder="partner@example.com"
            {...register('email')}
          />
          {errors.email && (
            <p className="text-xs text-red-400 mt-1">{errors.email.message}</p>
          )}
        </div>

        <div>
          <label className="block text-mystical-light mb-2 font-medium">Phone Number (for SMS)</label>
          <input
            type="tel"
            className={`w-full bg-mystical-dark border ${errors.phone ? 'border-red-400' : 'border-mys...`}
            placeholder="+1 (123) 456-7890"
            {...register('phone')}
          />
          {errors.phone && (
            <p className="text-xs text-red-400 mt-1">{errors.phone.message}</p>
          )}
        </div>

        <div className="flex space-x-3 pt-2">
          <motion.button
            type="button"
            className="w-1/3 border border-mystical-lavender/50 text-mystical-lavender font-medium py-...>
            whileHover={{ scale: 1.02 }}
            whileTap={{ scale: 0.98 }}
            onClick={() => setStep(1)}
          >
            Back
          </motion.button>

          <motion.button
            type="submit"
            className="w-2/3 bg-mystical-purple hover:bg-mystical-lavender text-white font-medium py-3...>
            whileHover={{ scale: 1.02 }}
            whileTap={{ scale: 0.98 }}
            disabled={isLoading}
          >
            {isLoading ? 'Processing...' : 'Send Invitation'}
          </motion.button>
        </div>
      </div>
    )
  </form>
</div>
) : (
  <div className="oracle-card bg-mystical-dark border-2 border-mystical-purple rounded-xl p-5 mb-8 text-...>
    <div className="w-20 h-20 bg-mystical-gradient rounded-full flex items-center justify-center mx-auto...>
      <i className="fas fa-envelope-open-text text-3xl text-white"></i>
    </div>

    <h3 className="font-serif text-2xl text-mystical-pink mb-2">Invitation Sent!</h3>
    <p className="text-mystical-lavender mb-6">
      We've sent an invitation to your partner. Once they complete their part of the test, you'll both r...
    </p>

    <div className="bg-mystical-purple/20 rounded-lg p-4 mb-4">
      <p className="text-sm text-mystical-lavender">
        <i className="fas fa-info-circle mr-2 text-mystical-pink"></i>
        Your partner will need to download the Love Oracle app to participate in this test.
      </p>
    </div>

```

```

        <motion.button
            className="w-full bg-mystical-purple hover:bg-mystical-lavender text-white font-medium py-3 rounded-lg"
            whileHover={{ scale: 1.02 }}
            whileTap={{ scale: 0.98 }}
            onClick={() => setInviteSent(false)}
        >
            Send Another Invitation
        </motion.button>
    </div>
}

/* Information Card */
<div className="oracle-card bg-mystical-dark border-2 border-mystical-purple rounded-xl p-5 mb-8">
    <h3 className="font-serif text-xl text-mystical-pink mb-3">About the Compatibility Test</h3>
    <p className="text-sm text-mystical-lavender mb-4">
        Our unique compatibility test uses AI-powered insights to analyze your cosmic connections with your ...
    </p>

    <div className="grid grid-cols-2 gap-3 mb-4">
        <div className="bg-mystical-purple/20 rounded-lg p-3 text-center">
            <div className="w-10 h-10 rounded-full bg-mystical-purple/40 flex items-center justify-center mx-auto" style="width: 10px; height: 10px; background-color: transparent; border-radius: 50%; border: 2px solid #f0e68c; position: relative; z-index: 1;">
                <i className="fas fa-puzzle-piece text-mystical-pink" style="position: absolute; left: -5px; top: -5px; font-size: 20px; color: #f0e68c; opacity: 0.8; transform: rotate(-45deg);"></i>
            </div>
            <h4 className="font-medium text-sm text-mystical-pink">Identify Patterns</h4>
        </div>

        <div className="bg-mystical-purple/20 rounded-lg p-3 text-center">
            <div className="w-10 h-10 rounded-full bg-mystical-purple/40 flex items-center justify-center mx-auto" style="width: 10px; height: 10px; background-color: transparent; border-radius: 50%; border: 2px solid #f0e68c; position: relative; z-index: 1;">
                <i className="fas fa-heart-pulse text-mystical-pink" style="position: absolute; left: -5px; top: -5px; font-size: 20px; color: #f0e68c; opacity: 0.8; transform: rotate(-45deg);"></i>
            </div>
            <h4 className="font-medium text-sm text-mystical-pink">Discover Compatibility</h4>
        </div>
    </div>

    <div className="premium-shimmer rounded-lg p-3 mb-4">
        <div className="flex items-center">
            <div className="w-10 h-10 rounded-full bg-mystical-gold/20 flex items-center justify-center mr-3" style="width: 10px; height: 10px; background-color: transparent; border-radius: 50%; border: 2px solid #f0e68c; position: relative; z-index: 1;">
                <i className="fas fa-crown text-mystical-gold" style="position: absolute; left: -5px; top: -5px; font-size: 20px; color: #f0e68c; opacity: 0.8; transform: rotate(-45deg);"></i>
            </div>
            <div>
                <h4 className="font-medium text-mystical-gold">Premium Feature</h4>
                <p className="text-xs text-mystical-light">Unlock detailed compatibility reports and relationships with your cosmic connections!</p>
            </div>
        </div>
    </div>

```

{ /* Heart-shaped Loader */ }

```

<HeartLoader
    isVisible={isLoading}
    message={getRandomLoaderMessage('compatibility')}
    size="large"
    color="mystical-pink"
/>
</section>
);
}
}

```