File: client/src/pages/LoveCircles.tsx

Type: tsx | Size: 14347 bytes

```tsx
import React, { useState } from 'react';
import { motion } from 'framer-motion';
import { useToast } from '@/hooks/use-toast';
import { useForm } from 'react-hook-form';
import { z } from 'zod';
import { zodResolver } from '@hookform/resolvers/zod';

const createCircleSchema = z.object({
  circleName: z.string().min(3, "Circle name must be at least 3 characters").max(30, "Circle name is too long"...
  description: z.string().max(100, "Description is too long").optional(),
  isPrivate: z.boolean().default(true),
});

type CreateCircleFormData = z.infer<typeof createCircleSchema>;

export default function LoveCircles() {
  const { toast } = useToast();
  const [activeTab, setActiveTab] = useState('my-circles');
  const [showCreateForm, setShowCreateForm] = useState(false);
  const [isLoading, setIsLoading] = useState(false);

  // Mock data for circles
  const [myCircles, setMyCircles] = useState([
    {
      id: 1,
      name: "Heart Squad",
      members: 4,
      lastActivity: "2 hours ago",
      unread: 3,
    },
    {
      id: 2,
      name: "Dating Diaries",
      members: 6,
      lastActivity: "Yesterday",
      unread: 0,
    }
  ]);

  const [discoverCircles, setDiscoverCircles] = useState([
    {
      id: 3,
      name: "Love Coven",
      members: 12,
      description: "For discussing spiritual aspects of love and relationships",
      isPopular: true,
    },
    {
      id: 4,
      name: "Situationship Support",
      members: 28,
      description: "Navigate the gray areas of modern dating together",
      isPopular: true,
    },
    {
      id: 5,
      name: "First Date Fears",
      members: 9,
      description: "Share first date stories and get advice",
      isPopular: false,
    }
  ]);

  const { register, handleSubmit, reset, formState: { errors } } = useForm<CreateCircleFormData>({
    resolver: zodResolver(createCircleSchema),
    defaultValues: {
      circleName: '',
      description: '',
      isPrivate: true,
    }
```

```
});

const onCreateCircle = async (data: CreateCircleFormData) => {
  try {
    setIsLoading(true);

    // Simulate API call
    await new Promise(resolve => setTimeout(resolve, 1000));

    // Add to my circles
    const newCircle = {
      id: Math.random(),
      name: data.circleName,
      members: 1,
      lastActivity: "Just now",
      unread: 0,
    };

    setMyCircles([newCircle, ...myCircles]);
    setShowCreateForm(false);
    reset();

    toast({
      title: "Circle Created",
      description: `Your new circle "${data.circleName}" has been created.`,
    });

  } catch (error) {
    toast({
      title: "Error",
      description: "Failed to create circle. Please try again.",
      variant: "destructive",
    });
  } finally {
    setIsLoading(false);
  }
};

const handleJoinCircle = (id: number, name: string) => {
  toast({
    description: `Joined the "${name}" circle.`,
  });

  // Move from discover to my circles
  const circle = discoverCircles.find(c => c.id === id);
  if (circle) {
    const newCircle = {
      id: circle.id,
      name: circle.name,
      members: circle.members + 1,
      lastActivity: "Just now",
      unread: 0,
    };

    setMyCircles([newCircle, ...myCircles]);
    setDiscoverCircles(discoverCircles.filter(c => c.id !== id));
  }
};

const handleViewCircle = (id: number) => {
  toast({
    description: `Opening circle chat (demo).`,
  });
};

return (
  <section className="px-4 py-4">
    <div className="text-center mb-6">
      <h2 className="font-serif text-3xl text-mystical-pink">Love Circles</h2>
      <p className="text-mystical-lavender">Share and discuss with your friends</p>
    </div>

    {/* Tabs */}
```

```
<div className="flex bg-mystical-dark/80 rounded-xl overflow-hidden mb-6">
  <button
    className={`flex-1 py-3 text-center text-sm transition-colors ${
      activeTab === 'my-circles'
        ? 'bg-mystical-gradient text-white'
        : 'text-mystical-lavender'
    }`}
    onClick={() => setActiveTab('my-circles')}
  >
    My Circles
  </button>
  <button
    className={`flex-1 py-3 text-center text-sm transition-colors ${
      activeTab === 'discover'
        ? 'bg-mystical-gradient text-white'
        : 'text-mystical-lavender'
    }`}
    onClick={() => setActiveTab('discover')}
  >
    Discover
  </button>
</div>

{/* Create Circle Button */}
{activeTab === 'my-circles' && !showCreateForm && (
  <motion.button
    className="w-full flex items-center justify-center bg-mystical-purple/30 border border-mystical-purp...
    onClick={() => setShowCreateForm(true)}
    whileHover={{ scale: 1.02 }}
    whileTap={{ scale: 0.98 }}
  >
    <i className="fas fa-plus-circle mr-2 text-mystical-pink"></i>
    <span className="text-mystical-light">Create New Circle</span>
  </motion.button>
)}

{/* Create Circle Form */}
{showCreateForm && (
  <div className="bg-mystical-gradient backdrop-blur-md rounded-xl p-5 mb-6">
    <div className="flex justify-between items-center mb-4">
      <h3 className="font-serif text-xl text-mystical-light">Create Circle</h3>
      <button
        className="text-mystical-light opacity-70 hover:opacity-100"
        onClick={() => setShowCreateForm(false)}
      >
        <i className="fas fa-times"></i>
      </button>
    </div>

    <form onSubmit={handleSubmit(onCreateCircle)}>
      <div className="mb-4">
        <label className="block text-mystical-light mb-2 text-sm">Circle Name</label>
        <input
          type="text"
          className={`w-full bg-white/90 backdrop-blur-sm border ${errors.circleName ? 'border-red-400' ...
          placeholder="e.g., Heart Squad, Love Coven"
          {...register('circleName')}
        />
        {errors.circleName && (
          <p className="text-xs text-red-400 mt-1">{errors.circleName.message}</p>
        )}
      </div>

      <div className="mb-4">
        <label className="block text-mystical-light mb-2 text-sm">Description (Optional)</label>
        <textarea
          className={`w-full bg-white/90 backdrop-blur-sm border ${errors.description ? 'border-red-400'...
          placeholder="What's this circle about?"
          {...register('description')}
        ></textarea>
        {errors.description && (
          <p className="text-xs text-red-400 mt-1">{errors.description.message}</p>
        )}
```

```
      </div>

      <div className="mb-5">
        <label className="flex items-center text-mystical-light text-sm cursor-pointer">
          <input
            type="checkbox"
            className="rounded text-mystical-purple mr-2"
            {...register('isPrivate')}
          />
          Private Circle (invitation only)
        </label>
      </div>

      <div className="flex space-x-3">
        <button
          type="button"
          className="flex-1 border border-mystical-lavender/30 text-mystical-lavender rounded-lg py-2"
          onClick={() => setShowCreateForm(false)}
        >
          Cancel
        </button>
        <motion.button
          type="submit"
          className="flex-1 bg-mystical-purple hover:bg-mystical-lavender text-white font-medium py-2 ro...
          whileHover={{ scale: 1.02 }}
          whileTap={{ scale: 0.98 }}
          disabled={isLoading}
        >
          {isLoading ? (
            <>
              <i className="fas fa-spinner fa-spin mr-2"></i>
              Creating...
            </>
          ) : (
            'Create Circle'
          )}
        </motion.button>
      </div>
    </form>
  </div>
)}

{/* My Circles Tab */}
{activeTab === 'my-circles' && (
  <div>
    {myCircles.length > 0 ? (
      <div className="space-y-3 mb-6">
        {myCircles.map(circle => (
          <motion.div
            key={circle.id}
            className="oracle-card bg-white backdrop-blur-md rounded-xl p-4 relative"
            whileHover={{ scale: 1.02 }}
            onClick={() => handleViewCircle(circle.id)}
          >
            <div className="flex justify-between items-start">
              <div>
                <h4 className="font-medium text-mystical-deep">{circle.name}</h4>
                <p className="text-xs text-gray-500">{circle.members} members • {circle.lastActivity}</p...
              </div>
              <div className="w-8 h-8 rounded-full bg-mystical-gradient/20 flex items-center justify-cen...
                <i className="fas fa-user-group text-mystical-purple"></i>
              </div>
            </div>

            {circle.unread > 0 && (
              <div className="absolute top-2 right-2 w-5 h-5 rounded-full bg-mystical-pink flex items-ce...
                <span className="text-white text-xs">{circle.unread}</span>
              </div>
            )}
          </motion.div>
        ))}
      </div>
    ) : (
```

```
      <div className="text-center py-8 text-mystical-lavender">
        <i className="fas fa-heart-crack text-2xl mb-2"></i>
        <p>You haven't joined any circles yet. Create one or discover public circles!</p>
      </div>
    )}
  </div>
)}

{/* Discover Tab */}
{activeTab === 'discover' && (
  <div>
    <div className="mb-4">
      <input
        type="text"
        className="w-full bg-white/90 backdrop-blur-sm border border-mystical-lavender/30 rounded-lg p-3...
        placeholder="Search circles..."
      />
    </div>

    <div className="space-y-3 mb-6">
      {discoverCircles.map(circle => (
        <motion.div
          key={circle.id}
          className="oracle-card bg-white backdrop-blur-md rounded-xl p-4 relative"
          whileHover={{ scale: 1.02 }}
        >
          {circle.isPopular && (
            <div className="absolute top-2 right-2 px-2 py-0.5 bg-mystical-gold/20 rounded-full">
              <span className="text-xs text-mystical-gold">Popular</span>
            </div>
          )}

          <div className="mb-3">
            <h4 className="font-medium text-mystical-deep">{circle.name}</h4>
            <p className="text-xs text-gray-500">{circle.members} members</p>
          </div>

          <p className="text-sm text-gray-600 mb-3">{circle.description}</p>

          <motion.button
            className="w-full bg-mystical-purple hover:bg-mystical-lavender text-white text-sm py-1.5 ro...
            whileHover={{ scale: 1.02 }}
            whileTap={{ scale: 0.98 }}
            onClick={() => handleJoinCircle(circle.id, circle.name)}
          >
            Join Circle
          </motion.button>
        </motion.div>
      ))}
    </div>
  </div>
)}

{/* Premium Banner */}
<div className="premium-shimmer rounded-xl p-5 mb-8">
  <div className="flex items-center justify-between">
    <div>
      <h3 className="font-serif text-xl text-mystical-gold mb-1">Premium Circles</h3>
      <p className="text-sm text-mystical-light/80 mb-2">
        Join expert-led circles with personalized advice
      </p>
      <motion.button
        className="bg-mystical-gold/80 hover:bg-mystical-gold text-mystical-deep text-sm font-medium py-...
        whileHover={{ scale: 1.05 }}
        whileTap={{ scale: 0.95 }}
        onClick={() => toast({ description: "Premium features are not available in this demo version." }...
      >
        Explore
      </motion.button>
    </div>
    <div className="w-16 h-16 opacity-70">
      <i className="fas fa-crown text-6xl text-mystical-gold"></i>
    </div>
```

```
        </div>
      </div>

      {/* Group Tips */}
      <div className="oracle-card bg-white backdrop-blur-md rounded-xl p-5 mb-6">
        <h3 className="font-serif text-xl text-mystical-deep mb-3">Circle Tips</h3>
        <ul className="text-sm text-gray-600 space-y-2">
          <li className="flex items-start">
            <i className="fas fa-circle-info text-mystical-purple mt-1 mr-2"></i>
            <span>Share your readings with your circle for group insights</span>
          </li>
          <li className="flex items-start">
            <i className="fas fa-circle-info text-mystical-purple mt-1 mr-2"></i>
            <span>Create private circles for discussions with close friends</span>
          </li>
          <li className="flex items-start">
            <i className="fas fa-circle-info text-mystical-purple mt-1 mr-2"></i>
            <span>Join public circles to get fresh perspectives on your situation</span>
          </li>
        </ul>
      </div>
    </section>
  );
}
```