File: client/src/pages/InteractiveReading.tsx

Type: tsx | Size: 17707 bytes

```tsx
import React, { useState } from "react";
import { motion, AnimatePresence } from "framer-motion";
import { useToast } from "@/hooks/use-toast";
import { useAds } from "@/hooks/use-ads";

export default function InteractiveReading() {
  const { toast } = useToast();
  const { trackAction } = useAds();
  const [step, setStep] = useState(0);
  const [selectedChoices, setSelectedChoices] = useState<number[]>([]);
  const [isTransitioning, setIsTransitioning] = useState(false);

  // Mock data for the interactive reading
  const readingSteps = [
    {
      id: 0,
      title: "Your Situation",
      description: "You have mentioned feeling uncertain about whether to reach out to someone you have been t...
      imageSrc: "crystal-ball",
      choices: [
        { id: 1, text: "I want to understand if they are thinking of me too" },
        { id: 2, text: "I want to know if I should make the first move" },
      ],
    },
    {
      id: 1,
      title: "The Connection",
      description: "I sense there is a strong energy between you two, but there is hesitation on both sides. T...
      imageSrc: "string-lights",
      choices: [
        { id: 3, text: "Our last conversation ended positively but faded" },
        { id: 4, text: "We had a small disagreement or miscommunication" },
      ],
    },
    {
      id: 2,
      title: "Your Initiative",
      description: "Making the first move requires courage. Let me help you understand what energy you bring t...
      imageSrc: "moon-stars",
      choices: [
        { id: 5, text: "I worry about being too forward or intense" },
        { id: 6, text: "I worry they might not reciprocate my feelings" },
      ],
    },
    {
      id: 3,
      title: "Faded Connection",
      description: "Many connections fade not from lack of interest, but from timing and life circumstances. L...
      imageSrc: "sunset",
      choices: [
        { id: 7, text: "I think they might be busy with other priorities" },
        { id: 8, text: "I wonder if they are waiting for me to reach out" },
      ],
    },
    {
      id: 4,
      title: "The Miscommunication",
      description: "Miscommunications happen when two people interpret the same situation differently. Underst...
      imageSrc: "tangled-string",
      choices: [
        { id: 9, text: "I think they misunderstood my intentions" },
        { id: 10, text: "I might have misinterpreted their actions" },
      ],
    },
    {
      id: 5,
      title: "Finding Balance",
      description: "Being aware of your intensity is a strength. Let's explore how you can channel this energy...
      imageSrc: "scales",
      choices: [
      { id: 11, text: "I want to show interest without overwhelming them" },
      { id: 12, text: "I want to be authentic but measured in my approach" },
    ],
  },
  {
    id: 6,
    title: "Fear of Rejection",
    description: "The fear of not having our feelings returned is universal. Understanding this fear can hel...
    imageSrc: "shield-heart",
    choices: [
      { id: 13, text: "I have been hurt before and am cautious" },
      { id: 14, text: "I value this connection and do not want to risk it" },
    ],
  },
];

const conclusions = [
  {
    id: "thinking-of-you",
    title: "The Oracle Reveals: Mutual Thoughts",
    message: "The cards suggest that yes, they are thinking of you too. There is a reciprocal energy in your...
    advice: "Consider sending a casual, no-pressure message. Something simple that opens the door but does n...
    actionStep: "A short, warm message acknowledging your connection could be the perfect way to reestablish...
    card: { color: "mystical-lavender", icon: "fa-brain" }
  },
  {
    id: "waiting-for-you",
    title: "The Oracle Reveals: The Waiting Game",
    message: "The cards indicate they may indeed be waiting for you to make the first move. Sometimes people...
    advice: "Taking the initiative does not mean laying all your feelings on the table at once. A simple ges...
    actionStep: "Reach out with something specific—perhaps a memory, a question, or something that made you ...
    card: { color: "mystical-purple", icon: "fa-hourglass-half" }
  },
  {
    id: "misunderstood-intentions",
    title: "The Oracle Reveals: Crossed Signals",
    message: "The cards show there is likely been a misunderstanding of intentions. What you meant and what ...
    advice: "Clear communication, without assumptions, is key. Being direct about your intentions—while rema...
    actionStep: "Consider addressing the misunderstanding directly, but gently. 'I realized we might have ha...
    card: { color: "mystical-pink", icon: "fa-comment-dots" }
  },
  {
    id: "authentic-approach",
    title: "The Oracle Reveals: Authentic Connection",
    message: "The cards emphasize that authenticity, even with measured approach, will serve you best. Your ...
    advice: "Being authentic does not mean sharing everything at once. It means being honest about where you...
    actionStep: "Reach out with honesty about your desire to connect, but without expectations. This balance...
    card: { color: "mystical-gold", icon: "fa-sparkles" }
  },
  {
    id: "protecting-connection",
    title: "The Oracle Reveals: Valuable Bonds",
    message: "The cards reflect your wisdom in recognizing the value of this connection. Some connections ar...
    advice: "Consider that protecting a connection sometimes means taking risks to nurture it. By reaching o...
    actionStep: "A message that acknowledges both the value you place on the connection and your respect for...
    card: { color: "mystical-lavender", icon: "fa-heart-pulse" }
  }
];

// This function would typically use the choices to determine
// which conclusion to show based on the path taken
const getConclusion = () => {
  // For demo purposes, select a conclusion based on the last choice
  const lastChoice = selectedChoices[selectedChoices.length - 1];

  if (lastChoice >= 7 && lastChoice <= 8) {
    return conclusions[1]; // waiting-for-you
  } else if (lastChoice >= 9 && lastChoice <= 10) {
    return conclusions[2]; // misunderstood-intentions
  } else if (lastChoice >= 11 && lastChoice <= 12) {
    return conclusions[3]; // authentic-approach
  } else if (lastChoice >= 13 && lastChoice <= 14) {
    return conclusions[4]; // protecting-connection
  } else {
    return conclusions[0]; // thinking-of-you (default)
  }
};

const currentStep = readingSteps.find(s => s.id === step);
const isAtFinalStep = !currentStep?.choices || currentStep.choices.length === 0;
const conclusion = isAtFinalStep ? getConclusion() : null;

const handleChoiceSelect = (choiceId: number) => {
  if (isTransitioning) return;

  // Start transition animation
  setIsTransitioning(true);

  // Delay the actual step change to allow for fade out animation
  setTimeout(() => {
    setSelectedChoices([...selectedChoices, choiceId]);

    // Find the next step based on the choice
    const nextStep = readingSteps.find(s => s.id === choiceId);
    if (nextStep) {
      setStep(nextStep.id);
    } else {
      // If no next step is found, we're at a conclusion
      setStep(-1); // Use -1 to indicate conclusion

      // Track completion of interactive reading and potentially show an ad
      trackAction();
    }

    // After another delay, end the transition state
    setTimeout(() => {
      setIsTransitioning(false);
    }, 400);
  }, 400);
};

const resetReading = () => {
  setIsTransitioning(true);

  setTimeout(() => {
    setStep(0);
    setSelectedChoices([]);
    setIsTransitioning(false);
  }, 400);
};

// Animation variants for steps and choices
const containerVariants = {
  hidden: { opacity: 0 },
  visible: {
    opacity: 1,
    transition: {
      duration: 0.8,
      ease: "easeInOut"
    }
  },
  exit: {
    opacity: 0,
    transition: {
      duration: 0.6,
      ease: "easeInOut"
    }
  }
};

return (
  <section className="px-4 py-4">
    <div className="text-center mb-6">
      <h2 className="font-serif text-3xl text-mystical-pink">Interactive Oracle</h2>
      <p className="text-mystical-lavender">Journey through your situation with the oracle&apos;s guidance</...
    </div>
```

```
<AnimatePresence mode="wait">
  {step === -1 ? (
    <motion.div
      key="conclusion"
      initial="hidden"
      animate="visible"
      exit="exit"
      variants={containerVariants}
      className="bg-mystical-gradient backdrop-blur-md rounded-xl p-5 mb-6"
    >
      <div className="text-center mb-5">
        <div className={`w-20 h-20 rounded-full bg-mystical-purple/20 flex items-center justify-center m...
          <i className={`fas ${conclusion?.card.icon} text-mystical-pink text-3xl`}></i>
        </div>
        <h3 className="font-serif text-2xl text-mystical-light mb-1">{conclusion?.title}</h3>
      </div>

      <div className="bg-white/10 backdrop-blur-sm rounded-lg p-4 mb-4 text-mystical-light">
        <p className="mb-3">{conclusion?.message}</p>
        <p className="mb-3 font-medium">{conclusion?.advice}</p>
        <p className="italic">{conclusion?.actionStep}</p>
      </div>

      <div className="space-y-3 mt-5">
        <motion.button
          className="w-full bg-mystical-purple/30 border border-mystical-purple text-white font-medium p...
          whileHover={{ scale: 1.02 }}
          whileTap={{ scale: 0.98 }}
          onClick={resetReading}
        >
          Start New Interactive Reading
        </motion.button>

        <motion.button
          className="w-full bg-white/10 border border-mystical-light/30 text-mystical-light font-medium ...
          whileHover={{ scale: 1.02 }}
          whileTap={{ scale: 0.98 }}
          onClick={() => {
            toast({
              title: "Reading Saved",
              description: "Your reading has been saved to your history.",
            });
          }}
        >
          Save This Reading
        </motion.button>
      </div>
    </motion.div>
  ) : (
    <motion.div
      key={`step-${step}`}
      initial="hidden"
      animate="visible"
      exit="exit"
      variants={containerVariants}
      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">{currentStep?.title}</h3>

        {step > 0 && (
          <button
            className="text-sm text-mystical-light/70 hover:text-mystical-light"
            onClick={() => {
              if (isTransitioning) return;

              setIsTransitioning(true);

              setTimeout(() => {
                // Go back to previous step
                const newChoices = [...selectedChoices];
                newChoices.pop(); // Remove last choice
        setSelectedChoices(newChoices);

        // Find the previous step ID
        const prevStepId = newChoices.length > 0
         ? newChoices[newChoices.length - 1]
         : 0;

        setStep(prevStepId);

        setTimeout(() => {
         setIsTransitioning(false);
        }, 400);
       }, 400);
      }}
      disabled={isTransitioning}
     >
      <i className="fas fa-arrow-left mr-1"></i> Back
     </button>
    )}
   </div>

   <div className="mb-5">
    <div className="mx-auto w-16 h-16 rounded-full bg-white/10 flex items-center justify-center mb-3...
     <i className={`fas fa-${currentStep?.imageSrc} text-mystical-pink text-xl`}></i>
    </div>

    <p className="text-mystical-light mb-4">{currentStep?.description}</p>

    <div className="space-y-3">
     {currentStep?.choices?.map(choice => (
      <motion.button
       key={choice.id}
       className={`w-full bg-white/10 backdrop-blur-sm border border-mystical-light/30 hover:bord...
       whileHover={!isTransitioning ? { scale: 1.02, backgroundColor: "rgba(255, 255, 255, 0.15)"...
       whileTap={!isTransitioning ? { scale: 0.98 } : {}}
       onClick={() => handleChoiceSelect(choice.id)}
       disabled={isTransitioning}
      >
       {choice.text}
      </motion.button>
     ))}
    </div>
   </div>

   <div className="flex justify-between text-xs text-mystical-light/70">
    <span>Step {step + 1} of {readingSteps.length}</span>
    <span>Your path: {selectedChoices.length} choices made</span>
   </div>
  </motion.div>
 )}
</AnimatePresence>

{/* Premium Banner */}
<div className="premium-shimmer rounded-xl p-5 mb-6 bg-mystical-dark/50 border border-mystical-gold">
 <div className="flex items-center">
  <div className="w-12 h-12 rounded-full bg-mystical-gold/20 flex items-center justify-center mr-4">
   <i className="fas fa-crown text-mystical-gold"></i>
  </div>
  <div>
   <h4 className="font-medium text-mystical-gold">Premium Interactive Readings</h4>
   <p className="text-xs text-mystical-light mb-2">Unlock deeper, more personalized interactive journ...
   <motion.button
    className="bg-mystical-gold text-mystical-deep text-xs font-medium py-1.5 px-3 rounded-lg"
    whileHover={{ scale: 1.05 }}
    whileTap={{ scale: 0.95 }}
    onClick={() => toast({ description: "Premium features are not available in this demo version." }...
   >
    Upgrade Now
   </motion.button>
  </div>
 </div>
</div>
```

```
      {/* Info Card */}
      <div className="oracle-card bg-mystical-dark border-2 border-mystical-purple rounded-xl p-5">
        <h3 className="font-serif text-xl text-mystical-pink mb-3">About Interactive Readings</h3>
        <p className="text-sm text-mystical-lavender mb-4">
          Interactive readings create a personalized journey based on your specific situation and choices. As ...
        </p>

        <div className="grid grid-cols-2 gap-3">
          <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-a...
              <i className="fas fa-road-circle-check text-mystical-pink"></i>
            </div>
            <h4 className="font-medium text-sm text-mystical-pink">Follow Your Path</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-a...
              <i className="fas fa-lightbulb-on text-mystical-pink"></i>
            </div>
            <h4 className="font-medium text-sm text-mystical-pink">Discover Insights</h4>
          </div>
        </div>
      </div>
    </section>
  );
}
```