Feature Implementation: AI Prospecting Advice Coach

Overview

Add an intelligent "Prospecting Advice" feature to the existing AI Prospecting section. This feature provides salespeople with real-time, actionable prospecting ideas when they're stuck, in a slump, or need fresh approaches. It uses Google Gemini Pro with grounding (internet search) to deliver current, relevant advice contextualized to our merchant services/payment processing business.

Feature Requirements

1. UI Component: Prospecting Advice Input

Location: Add to the AI Prospecting section, positioned prominently near the top or as a collapsible panel

Component Elements:

- **Header/Label:** "Sales Spark 💡" or "Prospecting Coach"
- **Subtitle:** "Stuck? Tell me what's on your mind and I'll give you actionable ideas for today."
- **Text Input:** Multi-line textarea with placeholder text
- **Microphone Button:** Voice-to-text dictation capability (use Web Speech API)
- **Submit Button:** "Get Ideas" or "Spark My Day"
- **Response Area:** Collapsible/expandable area to display AI-generated advice

Placeholder Text Examples:

```
"I don't know who to call today..."
"I keep getting rejected and I'm losing motivation..."
"I need fresh ideas for finding new merchants..."
"How do I approach restaurants that already have a processor?"
```

2. Technical Implementation

2.1 Frontend Component (React/TypeScript)

```
// components/ProspectingAdvice.tsx

import { useState, useRef } from 'react';
import { Mic, MicOff, Sparkles, Loader2 } from 'lucide-react';

interface ProspectingAdviceProps {
  onAdviceReceived?: (advice: string) => void;
}

export function ProspectingAdvice({ onAdviceReceived }: ProspectingAdviceProps) {
  const [input, setInput] = useState('');
  const [advice, setAdvice] = useState('');
  const [isLoading, setIsLoading] = useState(false);
  const [isListening, setIsListening] = useState(false);
  const recognitionRef = useRef<SpeechRecognition | null>(null);

  // Initialize Speech Recognition
  const startListening = () => {
    if (!('webkitSpeechRecognition' in window) && !('SpeechRecognition' in window
      alert('Speech recognition is not supported in this browser. Please use Chro
      return;
    }

    const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechReco
    recognitionRef.current = new SpeechRecognition();
    recognitionRef.current.continuous = true;
    recognitionRef.current.interimResults = true;

    recognitionRef.current.onresult = (event) => {
      let transcript = '';
      for (let i = 0; i < event.results.length; i++) {
        transcript += event.results[i][0].transcript;
      }
      setInput(transcript);
    };

    recognitionRef.current.onerror = (event) => {
      console.error('Speech recognition error:', event.error);
      setIsListening(false);
    };
```

```
  recognitionRef.current.onend = () => {
    setIsListening(false);
  };

  recognitionRef.current.start();
  setIsListening(true);
};

const stopListening = () => {
  if (recognitionRef.current) {
    recognitionRef.current.stop();
    setIsListening(false);
  }
};

const getAdvice = async () => {
  if (!input.trim()) return;

  setIsLoading(true);
  try {
    const response = await fetch('/api/prospecting-advice', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ userInput: input })
    });

    const data = await response.json();
    setAdvice(data.advice);
    onAdviceReceived?.(data.advice);
  } catch (error) {
    console.error('Error getting advice:', error);
    setAdvice('Unable to get advice at this time. Please try again.');
  } finally {
    setIsLoading(false);
  }
};

return (
  <div className="bg-gradient-to-br from-blue-50 to-indigo-50 dark:from-gray-80
    {/* Header */}
    <div className="flex items-center gap-2 mb-4">
      <Sparkles className="w-6 h-6 text-yellow-500" />
      <h3 className="text-xl font-bold text-gray-800 dark:text-white">Sales Spa
    </div>

    <p className="text-gray-600 dark:text-gray-300 mb-4 text-sm">
```

```
  Stuck in a rut? Tell me what's going on and I'll give you fresh, actionab
</p>

{/* Input Area */}
<div className="relative mb-4">
  <textarea
    value={input}
    onChange={(e) => setInput(e.target.value)}
    placeholder="I don't know who to call today... I keep getting rejected.
    className="w-full p-4 pr-12 rounded-lg border border-gray-300 dark:bord
    rows={3}
  />

  {/* Microphone Button */}
  <button
    onClick={isListening ? stopListening : startListening}
    className={`absolute right-3 top-3 p-2 rounded-full transition-all ${
      isListening
        ? 'bg-red-500 text-white animate-pulse'
        : 'bg-gray-200 dark:bg-gray-700 text-gray-600 dark:text-gray-300 ho
    }`}
    title={isListening ? 'Stop recording' : 'Start voice input'}
  >
    {isListening ? <MicOff className="w-5 h-5" /> : <Mic className="w-5 h-5
  </button>
</div>

{/* Submit Button */}
<button
  onClick={getAdvice}
  disabled={isLoading || !input.trim()}
  className="w-full py-3 px-6 bg-gradient-to-r from-blue-600 to-indigo-600
>
  {isLoading ? (
    <>
      <Loader2 className="w-5 h-5 animate-spin" />
      Getting Ideas...
    </>
  ) : (
    <>
      <Sparkles className="w-5 h-5" />
      Spark My Day
    </>
  )}
</button>

{/* Advice Response Area */}
```

```
      {advice && (
        <div className="mt-6 p-4 bg-white dark:bg-gray-800 rounded-lg border bord
          <h4 className="font-semibold text-green-700 dark:text-green-400 mb-2 fl
            <Sparkles className="w-4 h-4" />
            Your Prospecting Ideas for Today:
          </h4>
          <div className="prose dark:prose-invert max-w-none text-gray-700 dark:t
            {advice}
          </div>
        </div>
      )}
    </div>
  );
}
```

2.2 Backend API Endpoint

```
// server/routes/prospecting-advice.ts (or pages/api/prospecting-advice.ts for Ne

import { GoogleGenerativeAI } from '@google/generative-ai';

// Initialize Gemini
const genAI = new GoogleGenerativeAI(process.env.GEMINI_API_KEY!);

// Business Context - This should pull from your existing training/context
const BUSINESS_CONTEXT = `
You are an expert sales coach for a merchant services/payment processing company

WHAT WE SELL:
- Payment processing solutions (credit card processing, POS systems)
- Cash discount/dual pricing programs that eliminate processing fees for merchant
- Video brochures as innovative sales tools for pattern interrupt prospecting
- Full-service merchant accounts with competitive rates and transparent pricing

OUR VALUE PROPOSITIONS:
1. COST SAVINGS: We can typically save merchants 20-40% on processing fees or eli
2. NO LONG-TERM CONTRACTS: Month-to-month agreements, no early termination fees
3. TRANSPARENT PRICING: No hidden fees, no junk fees, clear statements
4. LOCAL SERVICE: Dedicated local rep (the salesperson) vs. 1-800 number support
5. FAST FUNDING: Next-day or same-day funding available
6. FREE EQUIPMENT: POS terminals, card readers, etc.

IDEAL PROSPECT TYPES:
- Restaurants, bars, cafes (high volume, high ticket)
- Retail stores (clothing, specialty shops, convenience stores)
```

```
- Auto repair shops, mechanics
- Medical/dental offices
- Salons, barbershops, spas
- Professional services (attorneys, accountants)
- B2B businesses
- Any business currently paying processing fees

OBJECTION HANDLING BASICS:
- "I'm in a contract" → We can often help buy out contracts, or set up for future
- "I'm happy with my processor" → Great! Mind if I do a free statement analysis j
- "I don't have time" → This takes 5 minutes and could save you thousands per yea
- "I just switched" → Perfect time to make sure you got what you were promised -
`;

export async function POST(request: Request) {
  try {
    const { userInput } = await request.json();

    if (!userInput || typeof userInput !== 'string') {
      return Response.json({ error: 'Invalid input' }, { status: 400 });
    }

    // Use Gemini Pro with grounding for internet search
    const model = genAI.getGenerativeModel({
      model: 'gemini-2.0-flash-exp',  // or 'gemini-pro' depending on availabilit
      tools: [{
        googleSearch: {}  // Enable Google Search grounding
      }]
    });

    const prompt = `
${BUSINESS_CONTEXT}

---

A salesperson on our team is reaching out for help. Here's what they said:

"${userInput}"

---

Based on what they're experiencing, provide SPECIFIC, ACTIONABLE prospecting advi

Your response should:
1. ACKNOWLEDGE their situation briefly (1 sentence)
2. Provide 3-5 SPECIFIC, ACTIONABLE ideas they can execute TODAY
3. Each idea should be concrete - include specific places to go, types of busines
```

```
4. Search the internet for current prospecting tips and techniques that are relev
5. Incorporate real-world, current sales strategies that are working in 2024/2025
6. Frame everything in the context of selling payment processing and video brochu
7. End with ONE motivational insight or mindset tip

Keep the tone encouraging but practical. These are experienced salespeople who ne

Format the response clearly with numbered action items. Be specific - instead of
`;

    const result = await model.generateContent(prompt);
    const response = await result.response;
    const advice = response.text();

    return Response.json({ advice });

  } catch (error) {
    console.error('Prospecting advice error:', error);
    return Response.json(
      { error: 'Failed to generate advice', advice: 'Unable to generate advice at
      { status: 500 }
    );
  }
}
```

---

3. Environment Setup

Add to your `.env` file:

```
GEMINI_API_KEY=your_gemini_api_key_here
```

4. Package Dependencies

Ensure these packages are installed:

```
npm install @google/generative-ai lucide-react
```

---

5. Integration Points

5.1 Add to Existing Prospecting Page

Import and add the component to your existing AI Prospecting section:

```
import { ProspectingAdvice } from '@/components/ProspectingAdvice';

// In your prospecting page/section:
<ProspectingAdvice />
```

5.2 Optional: Store Advice History

Consider storing generated advice in your database for:

- Tracking what advice resonates with salespeople
- Building a library of successful prospecting ideas
- Analytics on common struggles/questions

---

6. Alternative: Using Existing API Structure

If your app already has a pattern for AI requests (like through an existing chat or AI endpoint), adapt the prompt above to work with your existing infrastructure. The key components are:

1. **The business context** (what we sell, value props, ideal prospects)
2. **The dynamic prompt** that incorporates user input
3. **Internet grounding** for fresh, current advice
4. **Structured output** with numbered action items

---

7. Mobile Considerations

For mobile users:

- The Web Speech API works best in Chrome and Safari
- Consider adding a visual indicator when recording (the pulsing red mic button)
- Ensure the textarea is large enough for touch input
- The submit button should be easily tappable

---

8. Example User Scenarios & Expected Outputs

Scenario 1: "I don't know who to call today, I've burned through my leads" → AI provides: specific business types to cold walk, current events (tax season, holiday prep) that create urgency, geographic areas to canvass

Scenario 2: "I keep getting rejected and I'm losing confidence" → AI provides: mindset reframe, new opening lines to try, specific objection-handling phrases, suggestion to try different business vertical

Scenario 3: "How do I approach a restaurant that says they're happy with their processor?" → AI provides: specific questions to ask, statement analysis offer script, current restaurant industry pain points to reference

---

9. Testing Checklist

- Voice dictation works on desktop Chrome
- Voice dictation works on mobile Safari/Chrome
- API endpoint returns advice within reasonable time (<10 seconds)
- Error states display properly
- Loading state shows during API call
- Advice renders with proper formatting
- Component is responsive on all screen sizes

Summary

This feature adds an AI-powered "Sales Spark" coach that:

1. Accepts text or voice input about the salesperson's current state/challenges
2. Uses Google Gemini Pro with internet grounding to fetch current, relevant advice
3. Contextualizes all advice to your specific business (payment processing, video brochures)
4. Returns numbered, actionable ideas the salesperson can execute immediately

The goal is to get stuck salespeople unstuck with fresh, specific ideas - not generic motivation, but real tactics they can use TODAY.