PCBancard Interactive AI Training - Implementation Guide

Overview

This guide explains how to add the Interactive AI Training System to your existing Replit app. The system includes:

1. **Live Roleplay Simulator** - Practice against 20 AI merchant personas
2. **Objection Gauntlet** - Rapid-fire objection handling with scoring
3. **Scenario Trainer** - "What would you do?" situational training
4. **Delivery Analyzer** - Full presentation analysis with stage detection

---

Files to Add

1. Main Training Component

Copy `pcbancard-interactive-training.tsx` to your project:

```
src/components/training/InteractiveTraining.tsx
```

2. Add Route

In your router configuration (likely `App.tsx` or a routes file):

```
import InteractiveTraining from './components/training/InteractiveTraining';

// Add this route
<Route path="/training/interactive" element={<InteractiveTraining />} />
```

3. Add Navigation Link

Add a link to the interactive training from your existing "Teach Me the Presentation" section:

```
<Link
```

```
  to="/training/interactive"
  className="p-4 bg-emerald-500/10 border border-emerald-500/30 rounded-xl hover:
>
  <h3>🎭 Interactive AI Practice</h3>
  <p>Roleplay with AI merchants, handle objections, and get coached</p>
</Link>
```

Claude API Integration

The component includes placeholder response functions. For production, you need to connect to the Claude API.

Option A: Direct API Calls (Client-Side)

If your app already calls Claude from the frontend, update the `generateMerchantResponse` function:

```
// Replace the placeholder function with this:
async function generateMerchantResponse(
  persona: MerchantPersona,
  userInput: string,
  history: Message[]
): Promise<string> {
  const messages = history.map(m => ({
    role: m.role === 'user' ? 'user' : 'assistant',
    content: m.content
  }));

  // Add the new user message
  messages.push({ role: 'user', content: userInput });

  const response = await fetch('https://api.anthropic.com/v1/messages', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      // Your API key handling
    },
    body: JSON.stringify({
      model: 'claude-sonnet-4-20250514',
      max_tokens: 500,
      system: persona.systemPrompt,
      messages: messages
    })
  });
```

```
  const data = await response.json();
  return data.content[0].text;
}
```

Option B: Server-Side API Route (Recommended)

Create an API route to handle Claude calls securely:

/api/training/roleplay.ts

```
export async function POST(request: Request) {
  const { personaId, messages } = await request.json();

  // Find persona by ID
  const persona = MERCHANT_PERSONAS.find(p => p.id === personaId);
  if (!persona) {
    return new Response('Persona not found', { status: 404 });
  }

  const response = await fetch('https://api.anthropic.com/v1/messages', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      'x-api-key': process.env.ANTHROPIC_API_KEY,
      'anthropic-version': '2024-01-01'
    },
    body: JSON.stringify({
      model: 'claude-sonnet-4-20250514',
      max_tokens: 500,
      system: persona.systemPrompt + `

ROLEPLAY RULES:
1. You ARE ${persona.name}. Stay in character completely.
2. Respond as this merchant would - use their concerns, speech patterns, skeptici
3. Your objection style: "${persona.objectionStyle}"
4. If the rep does something smart (targets your weak points), warm up SLIGHTLY.
5. Keep responses to 1-3 sentences unless deeply engaged.
6. Never break character. Never give advice. Just BE this merchant.`,
      messages: messages
    })
  });

  const data = await response.json();
  return new Response(JSON.stringify({
    response: data.content[0].text
```

```
  }));
}
```

Then update the component to call your API:

```
const handleSend = async () => {
  // ... existing code ...

  const response = await fetch('/api/training/roleplay', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
      personaId: persona.id,
      messages: [...messages, { role: 'user', content: input.trim() }]
    })
  });

  const data = await response.json();

  const merchantResponse: Message = {
    role: 'merchant',
    content: data.response,
    timestamp: new Date()
  };

  setMessages(prev => [...prev, merchantResponse]);
  setIsLoading(false);
};
```

The 20 Merchant Personas

Easy (Practice fundamentals)

Persona	Business	Key Trait
Curious Carol	Coffee Shop	Asks lots of questions, open-minded
Friendly Fred	Hardware Store	Chatty, loves stories, needs to feel heard

Medium (Build real skills)

Persona	Business	

		Key Trait
Skeptical Sam	Restaurant	Seen it all, demands proof
Busy Barbara	Hair Salon	2 minutes max, values efficiency
Price-Only Patty	Convenience Store	Only cares about rates
Comparison Carla	Boutique	Shopping 4 processors, has spreadsheet
New Owner Nick	Sandwich Shop	Just bought business, overwhelmed

Hard (Handle real objections)

Persona	Business	Key Trait
Loyal Larry	Auto Shop	12-year relationship with processor
Burned Ben	Pizza Shop	Got screwed before, angry
Know-It-All Kevin	Electronics	Thinks he's the expert
Silent Steve	Dry Cleaner	One-word answers, uses silence
Contract Connie	Gym	"Locked in" (probably isn't)
Retiring Rita	Florist	"Closing next year, why bother?"

Expert (Master-level challenges)

Persona	Business	Key Trait
Tech-Resistant Tom	Diner	Hates all new technology
Family Frank	Deli	"Talk to my brother-in-law"
Cash-Heavy Carlos	Barber	"80% of my business is cash"
Just-Looking Janet	Gift Shop	Never makes decisions
Aggressive Al	Sports Bar	Confrontational, tests backbone
Conspiracy Carl	Pawn Shop	"What's the catch?"
Multi-Location Maria	Restaurant Group	Needs enterprise presentation

Key Features Explained

1. Dynamic AI Responses

The persona's `systemPrompt` gives Claude everything it needs to stay in character:

- Background (business type, years in business, volume)
- Emotional state and trust level
- Specific objection patterns
- What actually works on them

Example from Burned Ben:

```
You are Ben, a 48-year-old pizza shop owner who got BURNED.
Three years ago, a processor promised 1.9% and within 6 months
raised it to 3.5% with hidden fees. You process $45,000/month
and you're still angry about it. You assume every rep is a liar.
If someone acknowledges that the industry has problems and offers
concrete protection (rate locks, written guarantees, 90-day outs),
you'll crack slightly. But any vague promise makes you angrier.
```

2. Coaching Analysis

The coaching feature analyzes the conversation and provides:

- What the rep did well
- What to improve
- Specific suggestions based on the persona's weak points

3. Stage Detection

The Delivery Analyzer looks for keywords from each stage:

- Stage 1 (Visceral Opening): "knot in your stomach", "end of month", "not your fault"
- Stage 2 (Problem Quantification): "3 to 4 percent", "every swipe", "fees"
- Stage 3 (Marcus Story): "Marcus", "taqueria", "$17,412"
- etc.

Customization Options

Add New Personas

Add to the `MERCHANT_PERSONAS` array:

```
{
  id: 'your-persona-id',
  name: 'Display Name',
  title: 'Their Role',
  businessType: 'Business Type',
  avatar: '🤖',  // Emoji
  difficulty: 'Medium',
  personality: 'One-sentence personality description',
  openingLine: "What they say when you walk in",
  triggerPhrases: ['Things they say often', 'Verbal tics'],
  objectionStyle: 'How they push back',
  weakPoints: ['What works on them', 'Their vulnerabilities'],
  systemPrompt: `Full character description for Claude...`
}
```

Add New Objections

Add to the `OBJECTION_BANK` array:

```
{
  id: 'obj-new',
  category: 'Category Name',
  objection: "The exact objection text",
  difficulty: 'Hard',
  bestResponse: "The ideal response...",
  keyPrinciples: ['Key principle 1', 'Key principle 2'],
  commonMistakes: ['Mistake 1', 'Mistake 2']
}
```

Add New Scenarios

Add to the `SCENARIOS` array:

```
{
  id: 'scen-new',
  title: 'Scenario Title',
  setup: 'The situation description...',
  question: 'What do you do?',
```

```
  options: [
    { text: 'Option A', points: 3, feedback: 'Why this is okay/bad' },
    { text: 'Option B', points: 10, feedback: 'Why this is best' },
    { text: 'Option C', points: 6, feedback: 'Why this is decent' },
    { text: 'Option D', points: 1, feedback: 'Why this fails' }
  ],
  stage: 'Which presentation stage'
}
```

Testing Checklist

After implementation, verify:

- Interactive Training loads from main menu
- All 20 personas appear and are selectable
- Difficulty filter works (Easy/Medium/Hard/Expert)
- Roleplay conversation flows naturally
- AI responses stay in character
- “Get Coaching” button provides feedback
- Objection Gauntlet cycles through all 12 objections
- Scoring works correctly
- Scenario Trainer options are clickable
- Feedback appears after selection
- Delivery Analyzer detects stages correctly
- Mobile layout works
- Back buttons return to correct screens

Performance Notes

- Roleplay uses Claude API per message - expect 1-2 second response times
- Coaching analysis is a single API call

- Objection Gauntlet and Scenario Trainer work offline (no API needed)
- Delivery Analyzer can work offline (keyword matching) or use API for deeper analysis

---

Cost Estimate

Using Claude Sonnet for roleplay:

- ~200 tokens per exchange (input + output)
- 10 exchanges per practice session = ~2,000 tokens
- At $3/million input + $15/million output ≈ $0.02 per session
- 100 practice sessions/month = ~$2/month

Very cost-effective for training value delivered.