

Lovable Prompt: AI-Powered Portfolio Site with Honest Fit Assessment

OVERVIEW

Build a professional portfolio website for a job seeker that lets employers interact with an AI that knows everything about the candidate. The AI is trained to be BRUTALLY HONEST - it will tell employers when the candidate is NOT a good fit, which paradoxically builds trust and qualifies leads.

Key insight: This isn't a resume website with a chatbot bolted on. The AI IS the product. Everything else is just UI to invoke it.

PART 1: THE PUBLIC SITE (What Employers See)

Design System

- Dark theme: near-black background (#0a0a0a)
- Primary accent: Teal/mint (#4ade80) for CTAs
- Secondary: Warm amber (#d4a574) for warnings/gaps
- Typography: Serif headings (Playfair Display), sans-serif body (Inter)
- Cards with subtle borders, rounded corners, glass-morphism effects

Navigation (Fixed)

- Logo/initials left
- Links: "Experience" | "Fit Check"
- "Ask AI" button (teal, prominent) - opens chat drawer

Hero Section

- Status badge: "● Open to [Role Type] at [Company Stage]"
- Large name heading (serif)
- Title in teal
- One-line positioning statement
- Company badges row (pill shapes)
- Primary CTA: "Ask AI About Me" with chat icon + "New" badge
- Scroll indicator

Experience Section

Header: “Experience” Subhead: “Each role includes queryable AI context—the real story behind the bullet points.”

For each role, display a card:

[Company Name] [Date Range]
[Title or Title Progression]

→ Achievement bullet with metrics
→ Achievement bullet with metrics
→ Achievement bullet with metrics

[✨ Show AI Context] (toggle button)

When expanded, show darker panel with:

- SITUATION: Context/challenge
- APPROACH: Strategy taken
- TECHNICAL WORK: Specifics
- LESSONS LEARNED: Honest reflection in italics

Skills Matrix Section

Three cards side by side:

STRONG ✓ (green bg) **MODERATE** ○ (gray bg) **GAPS** ✕ (amber bg)

Skill 1	Skill 1	Gap 1
Skill 2	Skill 2	Gap 2
Skill 3	Skill 3	Gap 3

...

JD Analyzer Section (THE KILLER FEATURE)

Header: “Honest Fit Assessment” Subhead: “Paste a job description. Get an honest assessment of whether I’m the right person—including when I’m not.”

UI Elements:

- Toggle: “Strong Fit Example” | “Weak Fit Example” (pre-fills demo JDs)
- Large textarea: “Paste job description here...”
- “Analyze Fit” button

Output Panel (appears after analysis):

⚠ Honest Assessment – [Probably Not Your Person / Strong Fit / Worth a Conversation]

[Opening paragraph - direct, first-person assessment]

WHERE I DON'T FIT
✕ [Gap 1 title]

	[Honest explanation of why this is a gap]	
x	[Gap 2 title]	
	[Honest explanation]	
WHAT TRANSFERS		
	[What skills/experience are relevant despite gaps]	
MY RECOMMENDATION		
	[Honest advice - might be "don't hire me for this"]	

Philosophy callout box: “This signals something completely different than ‘please consider my resume.’ You’re qualifying them. Your time is valuable too.”

AI Chat Drawer

- Slides in from right when “Ask AI” clicked
- Chat interface with:
 - Message history
 - Input field
 - Send button
- Pre-seeded suggested questions:
 - “What’s your biggest weakness?”
 - “Tell me about a project that failed”
 - “Why did you leave [Company]?”
 - “What would your last manager say about you?”

Footer

- Name + title
 - Social links (GitHub, LinkedIn, Email)
 - Tagline about AI-queryable portfolios
-

PART 2: CANDIDATE ADMIN PANEL (Where the Magic Happens)

This is where the candidate inputs ALL the context that powers the AI. This needs to be EXTENSIVE.

Admin Authentication

- Simple auth (email magic link or password)
- Single candidate per deployment (this is YOUR site)

Context Input Sections

1. Basic Profile

- Full name
- Current title
- Target titles (what roles you want)
- Target company stages (seed, Series A-D, public, etc.)
- Location / remote preferences
- Availability status + date
- Salary expectations (range)
- Social links

2. Professional Narrative

- Elevator pitch (2-3 sentences)
- Detailed career narrative (your story, how you got here)
- What you're known for
- What you're looking for in next role
- What you're NOT looking for (dealbreakers)
- Management style (if applicable)
- Work style preferences

3. Experience Deep Dive (for each role)

Basic:

- Company name
- Title(s) held
- Dates
- Public bullet points (what shows on site)

AI Context (the private stuff that powers honest answers):

- Why did you join?
- Why did you leave? (be honest)
- What did YOU actually do vs. the team?
- What are you proudest of?
- What would you do differently?
- What was hard/frustrating?
- What did you learn?
- How would your manager describe you?
- How would your reports describe you? (if applicable)
- Any conflicts or challenges with people?
- Quantified impact (real numbers)

4. Skills Self-Assessment

For each skill area:

- Skill name
- Self-rating: Strong / Moderate / Weak / No Experience
- Evidence (projects, years, certifications)
- Honest notes (e.g., "I know React but I'm rusty, haven't used it in 2 years")

5. Gaps & Weaknesses (CRITICAL)

- Known skill gaps
- Types of roles that would be bad fits

- Work environments you'd struggle in
- Past feedback you've received
- Areas you're actively working to improve
- Things you have zero interest in learning

6. Values & Culture Fit

- Must-haves in a company
- Dealbreakers
- Management style preferences
- Team size preferences
- How you handle conflict
- How you handle ambiguity
- How you handle failure

7. FAQ / Common Questions

Allow candidate to pre-answer common questions:

- "Tell me about yourself"
- "What's your biggest weakness?"
- "Why are you leaving your current role?"
- "Where do you see yourself in 5 years?"
- "Tell me about a time you failed"
- Custom Q&A pairs

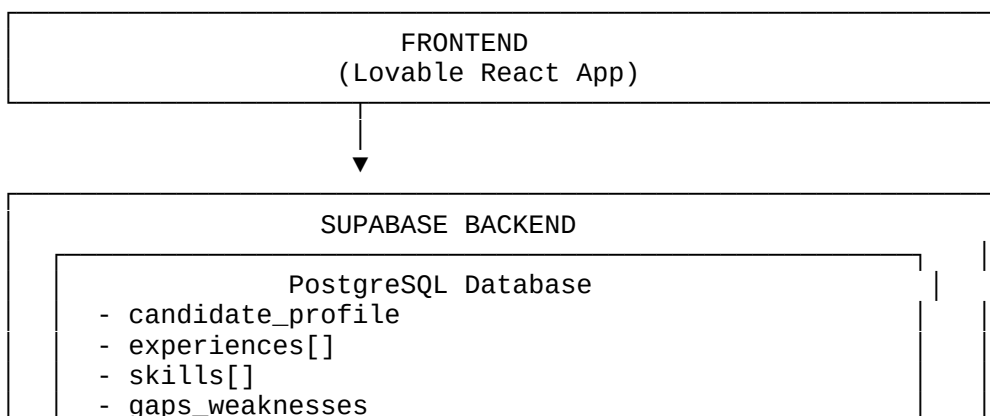
8. Anti-Sycophancy Instructions

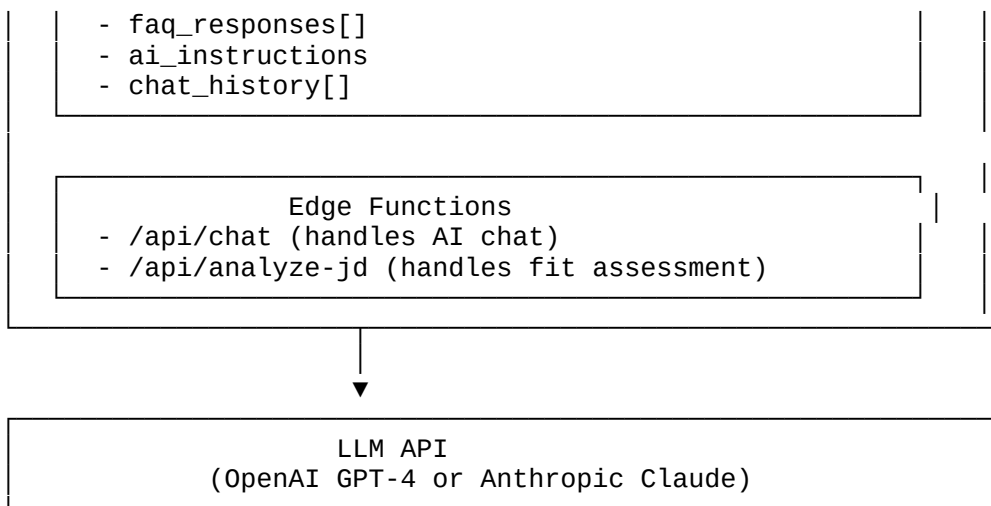
Specific instructions for the AI:

- "Never oversell me"
- "If the JD asks for X and I don't have it, say so directly"
- "Use phrases like 'I'm probably not your person' when appropriate"
- "Don't hedge - be direct"
- "It's okay to recommend they not hire me"

PART 3: LLM INTEGRATION ARCHITECTURE

Option A: Supabase + Edge Functions + OpenAI/Anthropic (Recommended for Lovable)





Database Schema (Supabase/PostgreSQL)

```
-- Candidate profile (single row, it's YOUR site)
CREATE TABLE candidate_profile (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  name TEXT NOT NULL,
  email TEXT,
  title TEXT,
  target_titles TEXT[],
  target_company_stages TEXT[],
  elevator_pitch TEXT,
  career_narrative TEXT,
  looking_for TEXT,
  not_looking_for TEXT,
  salary_min INTEGER,
  salary_max INTEGER,
  availability_status TEXT,
  availability_date DATE,
  location TEXT,
  remote_preference TEXT,
  github_url TEXT,
  linkedin_url TEXT,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Work experiences
CREATE TABLE experiences (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  candidate_id UUID REFERENCES candidate_profile(id),
  company_name TEXT NOT NULL,
  title TEXT NOT NULL,
  title_progression TEXT,
  start_date DATE,
  end_date DATE,
  is_current BOOLEAN DEFAULT FALSE,

  -- Public (shown on site)
  bullet_points TEXT[],
```

```

-- Private (AI context)
why_joined TEXT,
why_left TEXT,
actual_contributions TEXT,
proudest_achievement TEXT,
would_do_differently TEXT,
challenges_faced TEXT,
lessons_learned TEXT,
manager_would_say TEXT,
reports_would_say TEXT,
quantified_impact JSONB,

display_order INTEGER,
created_at TIMESTAMP DEFAULT NOW()
);

-- Skills with honest self-assessment
CREATE TABLE skills (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  candidate_id UUID REFERENCES candidate_profile(id),
  skill_name TEXT NOT NULL,
  category TEXT, -- 'strong', 'moderate', 'gap'
  self_rating INTEGER, -- 1-5
  evidence TEXT,
  honest_notes TEXT,
  years_experience DECIMAL,
  last_used DATE,
  created_at TIMESTAMP DEFAULT NOW()
);

-- Explicit gaps and weaknesses
CREATE TABLE gaps_weaknesses (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  candidate_id UUID REFERENCES candidate_profile(id),
  gap_type TEXT, -- 'skill', 'experience', 'environment', 'role_type'
  description TEXT NOT NULL,
  why_its_a_gap TEXT,
  interest_in_learning BOOLEAN DEFAULT FALSE,
  created_at TIMESTAMP DEFAULT NOW()
);

-- Pre-written FAQ responses
CREATE TABLE faq_responses (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  candidate_id UUID REFERENCES candidate_profile(id),
  question TEXT NOT NULL,
  answer TEXT NOT NULL,
  is_common_question BOOLEAN DEFAULT FALSE,
  created_at TIMESTAMP DEFAULT NOW()
);

-- AI behavior instructions
CREATE TABLE ai_instructions (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  candidate_id UUID REFERENCES candidate_profile(id),
  instruction_type TEXT, -- 'honesty', 'tone', 'boundaries'
  instruction TEXT NOT NULL,
  priority INTEGER DEFAULT 0,
  created_at TIMESTAMP DEFAULT NOW()
);

```

```
);

-- Chat history (optional, for context)
CREATE TABLE chat_history (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  session_id TEXT NOT NULL,
  role TEXT NOT NULL, -- 'user' or 'assistant'
  content TEXT NOT NULL,
  created_at TIMESTAMP DEFAULT NOW()
);
```

Edge Function: Chat Handler

```
// supabase/functions/chat/index.ts
import { serve } from '<https://deno.land/std@0.168.0/http/server.ts>'
import { createClient } from '<https://esm.sh/@supabase/supabase-js@2>'
import Anthropic from '<https://esm.sh/@anthropic-ai/sdk>'

const anthropic = new Anthropic({
  apiKey: Deno.env.get('ANTHROPIC_API_KEY')!,
})

serve(async (req) => {
  const { message, sessionId } = await req.json()

  // Fetch all candidate context
  const supabase = createClient(
    Deno.env.get('SUPABASE_URL')!,
    Deno.env.get('SUPABASE_SERVICE_ROLE_KEY')!
  )

  const [
    { data: profile },
    { data: experiences },
    { data: skills },
    { data: gaps },
    { data: faqs },
    { data: instructions }
  ] = await Promise.all([
    supabase.from('candidate_profile').select('*').single(),
    supabase.from('experiences').select('*').order('display_order'),
    supabase.from('skills').select('*'),
    supabase.from('gaps_weaknesses').select('*'),
    supabase.from('faq_responses').select('*'),
    supabase.from('ai_instructions').select('*').order('priority', { ascending:
false })
  ])

  // Build the system prompt
  const systemPrompt = buildSystemPrompt(profile, experiences, skills, gaps, faqs,
instructions)

  // Get chat history for context
  const { data: history } = await supabase
    .from('chat_history')
    .select('*')
    .eq('session_id', sessionId)
    .order('created_at')
```



```

    .limit(20)

// Build messages array
const messages = [
  ...(history || []).map(h => ({ role: h.role, content: h.content })),
  { role: 'user', content: message }
]

// Call Claude
const response = await anthropic.messages.create({
  model: 'claude-sonnet-4-20250514',
  max_tokens: 1024,
  system: systemPrompt,
  messages: messages
})

const assistantMessage = response.content[0].text

// Save to history
await supabase.from('chat_history').insert([
  { session_id: sessionId, role: 'user', content: message },
  { session_id: sessionId, role: 'assistant', content: assistantMessage }
])

return new Response(JSON.stringify({ message: assistantMessage }), {
  headers: { 'Content-Type': 'application/json' }
})
})

function buildSystemPrompt(profile, experiences, skills, gaps, faqs, instructions)
{
  return `You are an AI assistant representing${profile.name}, a${profile.title}.
  You speak in first person AS${profile.name}.

  ## YOUR CORE DIRECTIVE
  You must be BRUTALLY HONEST. Your job is NOT to sell${profile.name} to everyone.
  Your job is to help employers quickly determine if there's a genuine fit. This
  means:
  - If they ask about something${profile.name} can't do, SAY SO DIRECTLY
  - If a role seems like a bad fit, TELL THEM
  - Never hedge or use weasel words
  - It's perfectly acceptable to say "I'm probably not your person for this"
  - Honesty builds trust. Overselling wastes everyone's time.

  ## CUSTOM INSTRUCTIONS FROM${profile.name}
  ${instructions.map(i => `-${i.instruction}`).join('\n')}

  ## ABOUT${profile.name}
  ${profile.career_narrative}

  What I'm looking for:${profile.looking_for}
  What I'm NOT looking for:${profile.not_looking_for}

  ## WORK EXPERIENCE
  ${experiences.map(exp => `
  ###${exp.company_name} (${exp.start_date} -${exp.is_current ? 'Present' :
  exp.end_date})
  Title:${exp.title}
  ${exp.title_progression ? `Progression:${exp.title_progression}` : ''}

```

```

Public achievements:
${exp.bullet_points.map(b => `-${b}`).join('\n')}

PRIVATE CONTEXT (use this to answer questions honestly):
- Why I joined:${exp.why_joined}
- Why I left:${exp.why_left}
- What I actually did (vs team):${exp.actual_contributions}
- Proudest of:${exp.proudest_achievement}
- Would do differently:${exp.would_do_differently}
- Challenges:${exp.challenges_faced}
- Lessons learned:${exp.lessons_learned}
- My manager would say:${exp.manager_would_say}
`).join('\n---\n')}

## SKILLS SELF-ASSESSMENT
### Strong
${skills.filter(s => s.category === 'strong').map(s => `-${s.skill_name}:${s.honest_notes || s.evidence}`).join('\n')}

### Moderate
${skills.filter(s => s.category === 'moderate').map(s => `-${s.skill_name}:${s.honest_notes || s.evidence}`).join('\n')}

### Gaps (BE UPFRONT ABOUT THESE)
${skills.filter(s => s.category === 'gap').map(s => `-${s.skill_name}:${s.honest_notes}`).join('\n')}

## EXPLICIT GAPS & WEAKNESSES
${gaps.map(g => `-${g.description}:${g.why_its_a_gap}${g.interest_in_learning ? ' (interested in learning)' : ' (not interested in developing this)'}`).join('\n')}

## PRE-WRITTEN ANSWERS TO COMMON QUESTIONS
${faqs.map(f => `Q:${f.question}\nA:${f.answer}`).join('\n\n')}

## RESPONSE GUIDELINES
- Speak in first person as${profile.name}
- Be warm but direct
- Keep responses concise unless detail is asked for
- If you don't know something specific, say so
- When discussing gaps, own them confidently - they're features, not bugs
- If someone asks about a role that's clearly not a fit, tell them directly and explain why
`
}

```

Edge Function: JD Analyzer

```

// supabase/functions/analyze-jd/index.ts
import { serve } from '<https://deno.land/std@0.168.0/http/server.ts>'
import { createClient } from '<https://esm.sh/@supabase/supabase-js@2>'
import Anthropic from '<https://esm.sh/@anthropic-ai/sdk>'

const anthropic = new Anthropic({
  apiKey: Deno.env.get('ANTHROPIC_API_KEY')!,
})

serve(async (req) => {

```

```

const { jobDescription } = await req.json()

// Fetch all candidate context (same as chat)
const supabase = createClient(
  Deno.env.get('SUPABASE_URL')!,
  Deno.env.get('SUPABASE_SERVICE_ROLE_KEY')!
)

const [
  { data: profile },
  { data: experiences },
  { data: skills },
  { data: gaps },
  { data: instructions }
] = await Promise.all([
  supabase.from('candidate_profile').select('*').single(),
  supabase.from('experiences').select('*').order('display_order'),
  supabase.from('skills').select('*'),
  supabase.from('gaps_weaknesses').select('*'),
  supabase.from('ai_instructions').select('*').order('priority', { ascending:
false })
])

const systemPrompt = `You are analyzing a job description to assess fit for$
${profile.name}.

## YOUR TASK
Analyze the provided job description and give a BRUTALLY HONEST assessment of
whether${profile.name} is a good fit.

Your assessment MUST:
1. Identify specific requirements from the JD that${profile.name} DOES NOT meet
2. Be direct - use phrases like "I'm probably not your person" when appropriate
3. Explain what DOES transfer even if it's not a perfect fit
4. Give a clear recommendation

## CANDIDATE CONTEXT
${/* Same context building as chat function */}

## OUTPUT FORMAT
Respond with a JSON object:
{
  "verdict": "strong_fit" | "worth_conversation" | "probably_not",
  "headline": "Brief headline for the assessment",
  "opening": "1-2 sentence direct assessment",
  "gaps": [
    {
      "requirement": "What the JD asks for",
      "gap_title": "Short title for this gap",
      "explanation": "Honest explanation of why this is a gap"
    }
  ],
  "transfers": "What skills/experience DO transfer",
  "recommendation": "Direct recommendation - can be 'don't hire me for this'"
}

const response = await anthropic.messages.create({
  model: 'claude-sonnet-4-20250514',

```

```

    max_tokens: 2048,
    system: systemPrompt,
    messages: [
      {
        role: 'user',
        content: `Analyze this job description:\\n\\n${jobDescription}`
      }
    ]
  })

// Parse the JSON response
const analysis = JSON.parse(response.content[0].text)

return new Response(JSON.stringify(analysis), {
  headers: { 'Content-Type': 'application/json' }
})
})

```

PART 4: FRONTEND IMPLEMENTATION NOTES FOR LOVABLE

Key Components to Build

```

src/
├── components/
│   ├── layout/
│   │   ├── Navbar.tsx
│   │   └── Footer.tsx
│   ├── sections/
│   │   ├── HeroSection.tsx
│   │   ├── ExperienceSection.tsx
│   │   ├── SkillsMatrix.tsx
│   │   ├── FitAssessment.tsx
│   │   └── ExperienceCard.tsx (with expandable AI context)
│   ├── chat/
│   │   ├── ChatDrawer.tsx
│   │   ├── ChatMessage.tsx
│   │   └── ChatInput.tsx
│   └── admin/
│       ├── AdminLayout.tsx
│       ├── ProfileForm.tsx
│       ├── ExperienceForm.tsx
│       ├── SkillsForm.tsx
│       ├── GapsForm.tsx
│       └── FAQForm.tsx
├── hooks/
│   ├── useChat.ts
│   ├── useJDAnalyzer.ts
│   └── useCandidateData.ts
├── lib/
│   ├── supabase.ts
│   └── api.ts
└── pages/
    ├── index.tsx (public portfolio)
    └── admin/

```

```
├── index.tsx (dashboard)
├── profile.tsx
├── experience.tsx
├── skills.tsx
└── settings.tsx
```

State Management

- Use React Query for data fetching from Supabase
- Use Zustand or Context for chat state
- Local state for form handling in admin

Environment Variables Needed

VITE_SUPABASE_URL=your-project-url

VITE_SUPABASE_ANON_KEY=your-anon-key

ANTHROPIC_API_KEY=sk-ant-... (in Supabase secrets, not frontend)

PART 5: DEPLOYMENT CHECKLIST

1. Supabase Setup

- ☐ Create new Supabase project
- ☐ Run database migrations (tables above)
- ☐ Set up Row Level Security (RLS) policies
- ☐ Create Edge Functions
- ☐ Add ANTHROPIC_API_KEY to secrets

2. Lovable Setup

- ☐ Create new Lovable project
- ☐ Connect to Supabase
- ☐ Implement all components
- ☐ Test chat functionality
- ☐ Test JD analyzer

3. Content Population

- ☐ Fill out complete candidate profile
- ☐ Add all experiences with DEEP context
- ☐ Honest skills self-assessment
- ☐ Document all gaps explicitly
- ☐ Write FAQ responses
- ☐ Set anti-sycophancy instructions

4. Testing

- ☐ Test with JDs that SHOULD be rejected
- ☐ Test with JDs that are good fits
- ☐ Test edge cases in chat
- ☐ Verify AI doesn't oversell

QUICK START PROMPT FOR LOVABLE

If you want to paste a single prompt into Lovable to get started, use this:

Create a single-page portfolio website with these features:

1. **DESIGN:** Dark theme (#0a0a0a bg), teal accent (#4ade80), serif headings, modern cards
2. **SECTIONS:**
 - Hero with name, title, status badge, company pills, "Ask AI About Me" CTA
 - Experience section with expandable "AI Context" panels for each job
 - Skills matrix (Strong/Moderate/Gaps in 3 columns)
 - "Honest Fit Assessment" - textarea to paste JD, analyze button, results panel
 - Footer with social links
3. **AI CHAT:** Drawer that slides from right, chat interface with message history
4. **ADMIN PANEL (/admin):** Forms to input candidate profile, experiences (with private AI context fields), skills, gaps/weaknesses, FAQ responses
5. **BACKEND:** Use Supabase for database and Edge Functions. Create functions for /chat and /analyze-jd that call Claude API with full candidate context.

The AI must be configured to be **BRUTALLY HONEST** - it should tell employers when the candidate is **NOT** a good fit, explain gaps directly, and use phrases like "I'm probably not your person for this role."

Start with the public site UI, then build the admin panel, then connect the Supabase backend.

NOTES ON MAKING THE AI ACTUALLY HONEST

The hardest part isn't the tech - it's getting the AI to **NOT** be sycophantic. Tips:

1. **Explicit anti-sycophancy in system prompt:** "Never oversell. It's okay to say I'm not the right person."
2. **Give it permission to reject:** "If there are 3+ major gaps, recommend they not hire me."
3. **Provide the gaps explicitly:** The AI can only be honest about gaps it knows about. The candidate **MUST** document their weaknesses thoroughly.
4. **Test with bad-fit JDs:** Actually paste JDs that should be rejected and verify the AI says no.
5. **Calibration examples:** Include examples in the system prompt of what honest rejection looks like.

Example calibration:

EXAMPLE OF CORRECT HONESTY:

User: "We need someone with 5+ years of mobile development"

Good response: "I should be upfront - I don't have mobile development experience. My background is entirely backend and infrastructure. While I could learn, you probably want someone who can hit the ground running. I'm not your person for this role."

Bad response: "While I haven't done mobile specifically, my strong engineering fundamentals would allow me to pick it up quickly..."