CV SCANNER INTELLIGENCE

Build a $4K/Month Solution That Saves Recruiters 20 Hours Weekly

By Josh | Amalfi AI

I BUILT THIS IN 2 DAYS FOR A RECRUITMENT AGENCY

Here's the problem I walked into:

• Agency screening 150-200 CVs weekly
• Each CV: 10-15 minutes to read, evaluate, note
• That's 25-30 hours weekly = $4,000+ in recruiter time
• They were hiring a second recruiter just to handle volume

What I charged: $5K/month retainer
What I built: AI system that does in 30 seconds what took them 10 minutes
Build time: 2 days
Their time saved: 20+ hours weekly

ROI for them: $4K weekly savings vs $5K monthly cost. They break even in week 2 every month.

This guide shows you how to build this exact system. Every tool. Every prompt. Every integration. By the end, you'll have a working CV scanner you can demo to recruitment agencies tomorrow.

WHY THIS PRODUCT PRINTS MONEY

THE RECRUITMENT PAIN POINT

Every recruitment agency has this problem:

• **Volume:** 100-300 CVs per week per recruiter
• **Time sink:** 10-15 min per CV to extract relevant info
• **Inconsistency:** Different recruiters flag different things
• **Missed details:** Human fatigue = overlooked qualifications
• **No ranking:** Manual sorting by "gut feel"

The manual process:
1. Download CV
2. Read through 2-4 pages
3. Extract: name, contact, experience, education, skills
4. Check against job requirements
5. Write summary notes
6. Rank by fit
7. Add to tracking spreadsheet

Time per CV: 10-15 minutes
Error rate: 15-20% (missed details, inconsistent evaluation)

Your AI system does all 7 steps in 30 seconds. With 95%+ accuracy.

THE MATH THAT CLOSES DEALS

Small Agency (1-2 recruiters):
• Processing 100 CVs/week
• 15 min per CV = 25 hours weekly
• At $80/hour recruiter cost = $2,000 weekly = $8K monthly waste

Your solution: $4K/month
They save: $4K monthly + get better candidate insights

Medium Agency (5-10 recruiters):
• Processing 500+ CVs/week
• 125 hours weekly total
• At $80/hour = $10K weekly = $40K monthly cost

Your solution: $6-8K/month
They save: $32K+ monthly

This isn't a hard sell. The math is obvious.

THE COMPLETE TECH STACK

1. Claude API (Anthropic)
What it does: Extracts and analyzes CV content
Cost: $0.50-1.50 per CV (depending on length)
Why: Best at understanding context, handling multiple formats

2. PyPDF2 or pdf-parse
What it does: Converts PDF to text
Cost: Free (open source)
Why: Handles 90% of CV formats reliably

3. Supabase or Airtable
What it does: Database for storing candidate data
Cost: $0-25/month depending on volume
Why: Easy to set up, client can view directly

4. Make.com or n8n
What it does: Automates the workflow (upload → process → store → notify)
Cost: $9-29/month
Why: Visual builder, no coding needed

5. Google Sheets (for output)
What it does: Client-facing dashboard with ranked candidates
Cost: Free
Why: Everyone knows how to use it

TOTAL MONTHLY COST: $30-80 depending on CV volume
What you charge: $4-6K/month
Your margin: 95%+

THE 2-DAY BUILD PROCESS

DAY 1: CORE EXTRACTION ENGINE

Hour 1-2: Set Up Claude API

1. Get API key from anthropic.com
2. Test in Claude's workbench with a sample CV
3. Build the extraction prompt (next section)

Hour 3-5: Write the Extraction Prompt

This is the entire system. Get this right and everything works. Here's mine:

"You are a professional recruitment analyst. Extract the following from this CV:

CANDIDATE INFO:
- Full name
- Email
- Phone
- Location

EXPERIENCE:
- Total years of relevant experience
- Current/most recent role and company
- Key achievements (bullet points)
- Previous roles (company, title, duration)

EDUCATION:
- Highest qualification
- Institution and year
- Relevant certifications

SKILLS:
- Technical skills (list)
- Soft skills mentioned
- Languages

JOB FIT ANALYSIS:
Compare candidate against these requirements: [JOB_REQUIREMENTS]

Provide:

1. Match score (0-100)
2. Strengths (what aligns well)
3. Gaps (what's missing)
4. Red flags (career gaps, job hopping, etc)
5. Recommendation (Strong Fit / Potential Fit / Not a Fit)

Return as structured JSON."

Hour 6-8: Build PDF Parser

Ask Claude: "Write me a Python script that:
1. Takes a PDF file
2. Extracts all text
3. Sends to Claude API with my extraction prompt
4. Returns structured JSON
5. Handles errors gracefully"

Claude will write the entire script. You test it on 20+ CVs. Fix any parsing errors.

DAY 2: AUTOMATION & CLIENT INTERFACE

Hour 1-3: Build Make.com Workflow

Create scenario:
• **Trigger:** Watch Google Drive folder (client uploads CVs here)
• **Action 1:** Run Python script via webhook
• **Action 2:** Store results in Supabase
• **Action 3:** Add to Google Sheets ranking
• **Action 4:** Send Slack notification to client

Hour 4-6: Build Client Dashboard

Google Sheet with tabs:
• **Tab 1 - Ranked Candidates:** Match score, name, contact, recommendation
• **Tab 2 - Detailed Analysis:** Full extraction for each candidate
• **Tab 3 - Job Requirements:** Editable criteria for matching
• **Tab 4 - Stats:** Total processed, avg match score, top candidates

Hour 7-8: Testing & Polish

- Upload 50+ test CVs (download samples from job boards)
- Verify accuracy of extraction
- Check ranking logic
- Ensure dashboard updates in real-time

THE EXACT CLAUDE PROMPTS

Prompt 1: System Design

"I need to build a CV screening system for recruitment agencies. It should:
1. Extract candidate info from PDFs
2. Rank them against job requirements
3. Output to a client dashboard

What's the best tech stack? I need specific tools and why each one."

Prompt 2: Build the Parser

"Write a Python script that:
- Takes PDF file path as input
- Extracts all text using PyPDF2
- Sends to Claude API with this prompt: [paste extraction prompt]
- Returns structured JSON
- Handles errors if PDF is corrupted or image-based

Include error handling and logging. Explain each section."

Prompt 3: Build Ranking Logic

"I have candidate data in this JSON format: [paste sample]

And job requirements: [paste requirements]

Write a function that ranks candidates 1-100 based on:
- Skills match (40%)
- Experience relevance (30%)
- Education fit (20%)
- Location/availability (10%)

Return ranked list with scores explained."

THE DEMO THAT CLOSES

Pre-Demo: Get Their Real Job Posting

Before the call: "Can you send me a current job posting and 2-3 sample CVs you're screening? Want to show you this working on your actual use case."

Then customize the extraction prompt with THEIR requirements. Demo becomes instantly relevant.

The Live Demo

"Let me show you what 10 minutes of manual work looks like in 30 seconds..."

1. Upload one of their CVs to the drive folder
2. Watch it process in real-time
3. Pull up the dashboard - show the extraction
4. Point out the match score and reasoning
5. Upload 3 more CVs simultaneously - show it handles batch

Then the kicker: *"This just analyzed 4 candidates in under 2 minutes. How long would that take your team?"*

OBJECTION HANDLING

"What if it misses important details?"
"It will catch 95%+. Your recruiters currently miss 15-20% due to fatigue. Plus, I can update the extraction criteria in 5 minutes based on feedback. Training a human takes weeks."

"Can it handle different CV formats?"
"Yes - PDFs, Word docs, even scanned images if needed. I've tested on 200+ formats. Show me one that breaks it and I'll fix it same day."

"We need human judgment"
"Agreed. This handles the data extraction and initial ranking. Your recruiters focus on the top 20% - the actual assessment and client matching. They stop wasting time on data entry."

THE 14-DAY DELIVERY TIMELINE

Week 1: Build & Test

- Day 1-2: Discovery + requirements gathering
- Day 3-5: Build customized version
- Day 6-7: Internal testing with client's real CVs

Week 2: Deploy & Train

- Day 8-9: Client testing and feedback
- Day 10-11: Refinements
- Day 12-13: Train their team on dashboard
- Day 14: Go live, first batch processing

POST-LAUNCH OPTIMIZATION

First 30 days:

- Monitor extraction accuracy
- Adjust ranking criteria based on actual placements
- Add new data fields they discover they need
- Optimize for their specific CV formats

SCALING TO MULTIPLE CLIENTS

After client #1, you have:
• Base extraction prompt (90% reusable)
• Parser script (100% reusable)
• Make.com workflow (100% reusable)
• Dashboard template (100% reusable)

Client #2 setup time: 4 hours (not 2 days)
Client #3+: 2-3 hours

You're customizing job requirements, not rebuilding the system.

INDUSTRY EXPANSIONS

Same system works for:
• **HR departments:** Internal hiring screening
• **Universities:** Student application processing
• **Contracting firms:** Contractor qualification
• **Legal firms:** Paralegal/associate screening

THE UNCOMFORTABLE TRUTH

This system works. But here's what I can't give you:

The willingness to handle imperfection.

The AI will make mistakes. It'll miss a certification. Rank someone wrong. Extract a phone number incorrectly.

That's fine. Your job is to fix it fast. Update the prompt. Adjust the ranking weights. Add error handling.

I refined this system over 3 months with real client feedback. Version 1 was 70% accurate. Version 10 is 95%+.

Perfection isn't the goal. Better than manual is the goal.

WANT THE COMPLETE EXTRACTION PROMPTS AND PARSER CODE?

Join the Amalfi AI Mentorship: $97 one-time

• Complete extraction prompts for 5 industries
• Python parser script (copy-paste ready)
• Make.com workflow template
• Dashboard template with formulas
• Demo script and objection handling
• Client onboarding checklist

www.amalfiai.com/mentorship