

# Take-home assessment: AI Software Engineer

InLogic | AI Software Consulting

---

## What this is

This is a 7-day take-home project. You will build and deploy an AI-powered document processing application for a fictional property management client.

You are expected to use AI coding tools (Cursor, Claude Code, Copilot, ChatGPT, etc.) as part of your workflow. We are an AI consulting company, so how you work with AI is part of what we evaluate.

**Duration:** 7 days from receipt of this brief.

---

## The scenario

Acme Property Group is a fictional property management company with 200+ rental properties across Melbourne.

When a new tenant signs a lease, someone on the operations team reads through the agreement and manually types the key details into a Word document called the "Tenant Welcome Pack." This currently takes 15-30 minutes per tenant and is prone to errors.

Your task is to build a web application that automates this process:

1. Upload a signed lease agreement (PDF or DOCX)
  2. Extract the relevant tenant and property fields using AI
  3. Populate a Tenant Welcome Pack (.docx) from the provided template
  4. Deploy the application at a publicly accessible URL
- 

## Fields to extract

Your application must extract the following 14 fields from each lease agreement:

Field	Notes
Tenant name(s)	Some leases have two tenants (joint tenancy)
Property address	Full address including suburb, state, and postcode
Lease start date	Date formats vary between leases
Lease end date	Date formats vary between leases
Rent amount	Payment frequency varies (monthly, fortnightly)
Bond/deposit amount	Dollar amount
Number of occupants	As stated in the lease
Pet permission	"Not permitted" or a description including any conditions
Parking	"Not included" or details such as space number and location
Special conditions	Free text if present. If none exist, the Special Conditions section must be <b>omitted entirely</b> from the generated Welcome Pack. Do not display a heading with "None" or "Nil" underneath.
Landlord name	As listed in the parties section
Property manager name	Contact person
Property manager email	
Property manager phone	

## Materials provided

Included with this brief:

**5 sample lease agreements.** These are documents for tenants at properties managed by the fictional Acme Property Group. Each lease is formatted differently and contains different content. The variations are intentional and include things like: joint tenancy (two tenant names), different date formats, fortnightly vs. monthly rent, pet clauses with conditions, parking details, and special conditions. Your application needs to handle all variations.

**1 Tenant Welcome Pack template (.docx).** This is the Word document your application populates. It contains {{placeholder}} markers (e.g. {{tenant\_name}} ,

`{{property_address}}`, `{{rent_amount}}` ). Your app replaces these markers with the extracted data and produces a clean output document.

---

## Tech requirements

<b>Backend</b>	Python (FastAPI)
<b>Document output</b>	Word (.docx)
<b>Frontend</b>	Your choice
<b>Database</b>	Supabase (free tier) for storing extracted data, upload history, and generated files
<b>AI/LLM</b>	Free-tier APIs only. We do not provide API keys or credits.
<b>Hosting</b>	Must be deployed and accessible at a public URL (Use any free deployment service, e.g. Railway)
<b>Source code</b>	Git repository (GitHub)

---

## What to submit

At the end of the 7-day period, submit the following:

1. **Deployed application URL** where we can upload leases and test the full flow
  2. **Git repository** with full commit history
  3. **AI chat logs** exported from every AI tool you used during development (Cursor conversations, Claude Code transcripts, ChatGPT threads, etc.). This is a required part of the submission.
  4. **Daily progress updates** sent throughout the week (see below)
  5. **Final write-up** covering your approach, the trade-offs you made, and what you would do differently with more time
- 

## Daily progress updates

We'd like you to send a progress update at the end of each day you work on the project. Include a short Loom video (2-5 minutes) walking through what you built that day, alongside a written summary.

## **Written format:**

Day [X] – [Date]

What I completed:

– ...

Challenges:

– ...

How I used AI:

– ...

Next steps:

– ...

You don't need to send one every single day. If you take a day off or don't make meaningful progress, that's fine. But we'd like to see at least 3-4 updates over the week. The updates help us understand how you work, how you communicate progress, and how you think through problems.

Send these updates through email.

---

## **AI tools and chat logs**

You are expected to use AI coding tools during this project. We want to understand how you work with them.

As part of your submission, export and include your chat logs from all AI tools used (Cursor, Claude Code, ChatGPT, Copilot, etc.). We review these to see how you break down problems, structure prompts, iterate on outputs, and handle cases where the AI produces incorrect results.

---

## **How we evaluate**

We review each submission holistically. There is no fixed scoring rubric. The areas we pay attention to:

**Extraction accuracy.** Does the application correctly extract all 14 fields from all 5 leases? The leases vary in formatting, structure, and content. We check whether the extraction handles the

differences (joint tenancy names, varying date formats, fortnightly rent, pet clauses with conditions, etc.).

**Document generation.** Is the generated Welcome Pack populated correctly? Does the formatting hold up? Is the Special Conditions section omitted when there are no special conditions, and included when there are?

**Deployment.** We visit the URL and test the application. It needs to be working when we check.

**Code quality.** We read the code. We're looking for clear structure, reasonable naming, and sensible architectural decisions.

**Supabase usage.** How you designed the schema, how you handle file storage.

**Communication.** The quality of your daily updates, Loom videos, and final write-up.

**AI tool usage.** We read the chat logs. We're looking at how you prompt, how you iterate, and whether you catch and correct mistakes.

---

## Additional information

**Questions are welcome.** If something in this brief is unclear, reach out and ask.

**Originality.** Your submission must be your own work (with AI assistance). Copying from existing repositories or other candidates will result in disqualification.

**After submission.** If your submission passes our review, we will schedule a final interview. In that interview you will walk through your solution, discuss your architectural decisions, and we will look at specific parts of your AI chat logs together.

---

Good Luck!