THE HONEST AI INCOME SERIES

50 WAYS TO MAKE MONEY with AI



IT PROFESSIONALS

- ✓ Ready-to-use tools
- ✓ Automation workflows
- ✓ Spreadsheet templates
- ✓ Where to find customers

THE HONEST AI INCOME SERIES

50 Ways to Make Money with AI

for IT Professionals

Each idea rated for difficulty, cost, risk and evidence, with prompts you can copy and use today.

How this book is checked

Last updated: 02/11/2026.

Each idea is checked against sources. Here is what that means, and what it does not.

- Sources opened and read: for 48 of 50 ideas, at least one source was opened and read (the date is on each idea, in the line "Source check").
- Of those, 31 ideas have every cited link opened, and 17 have some links opened and others only seen in search results.
- 0 ideas rest only on search results we have not yet opened, and 2 have no source we could find. Check those yourself before relying on them.
- Prices shown are asking prices on sellers' listings. They are not what anyone earns. We found no reliable income figures, and we do not promise any income.
- Rules change. Benefits, tax, work-rights, privacy and advertising rules differ by country and over time. The table "Rules to check yourself" at the end of the book names who to ask.
- This is general information, not legal, tax, financial or immigration advice.
- Found a mistake or a broken link? Tell us and we will fix it and update the date.

A note before we start

This book is not financial, legal, tax, employment or immigration advice. We do not promise that you will earn anything. Many people earn nothing from some of these ideas, and some ideas will not suit your skills, your employment contract or your visa.

Where we found a reliable income figure, we show it with its source. Where we did not, the card says so plainly. Prices you see are asking prices on listings, not earnings. We never invent numbers, stories or customers.

Rules change. Platform terms, tax rules and contract law are updated often. Every card shows the date we read each source. Please open the official pages yourself before you act.

Hello, and welcome

You already have a skill most people pay for: you make computers do what they are supposed to do. This book is a menu of 50 small, honest ways to turn that skill into side income, with AI doing the slow drafting and you staying in charge of what is true, tested and safe.

Each idea sits on its own few pages, laid out the same way, so you always know where to look. Pick one that fits your week. Ignore the rest for now. Many of these ideas are crowded, and some will earn you nothing, so start small and treat the first hour as a test.

Every idea has a **Your first hour** section, so you can find out in one sitting whether it suits you. Every idea has **copy-and-paste prompts**, so you never face a blank screen. And every idea ends with a small **helper** you save in your own AI tool, so the second time is quick.

Two things before you sell anything: read your employment contract, and agree access with any client in writing. Both are covered in plain words in the check near the start of this book. A promise from us: we would rather tell you an idea is crowded or risky than sell you a dream.



A gentle path from the first idea to a keepable habit. Money is never guaranteed, and you can stop at any step.

The side-work and client-access check

Every idea in this book points back to this page. Read it once, before you sell anything. It takes five minutes and can save you a dispute with your employer, a client or a regulator.

Your employment contract. Many IT contracts have a moonlighting clause, a conflict of interest clause or an intellectual property (IP) clause. Some say that work you make with company time, equipment, accounts or knowledge belongs to your employer, and a few reach even things made in your own time on your own laptop. Read the contract and any outside-work policy. If a clause is unclear, ask your manager or HR in writing, or ask an employment lawyer.

Your employer's and other clients' material. Never reuse an employer's or another client's code, scripts, runbooks, logs, screenshots, customer lists or internal documents, in a product, a sample or an AI prompt. Build from what you made yourself or from public sources.

Client systems and written permission. Only touch systems you own or that the owner has authorised in writing. Ask for read-only access, keep client credentials in the client's own account, and agree the scope in a signed document. This book does not cover penetration testing, vulnerability scanning or network scanning, and none of the ideas asks you to do either. Do not do either without a signed document that allows it.

AI-written code and licences. Code from an AI tool can be too close to licensed code, and who owns AI-generated code is still unsettled in many countries. Treat it as a draft: review it, rewrite what you need, run it and test it. Check the licence of every library you ship (permissive licences such as MIT and Apache differ from licences such as GPLv3 that require derived works to share their source). Never paste customer data, passwords, API keys or private logs into an AI tool.

Customer data and privacy. Logs, mailboxes and spreadsheets often hold personal data. Process as little as you can, keep it on your own machine

where possible, and delete it when the job ends. Privacy rules differ by country, so ask the regulator named in the table "Rules to check yourself" at the end of the book.

Tax, work rights and visas. Side income is taxed under your own country's rules; ask the IRS in the US, HMRC in the UK, the ATO in Australia or the CRA in Canada. Contractors and visa holders should also check their work rights with the immigration authority (USCIS, UK Visas and Immigration, Home Affairs, IRCC). Nothing in this book is tax or immigration advice.

Promises and liability. Never tell a client that your checklist, review or set-up makes them secure or compliant, and do not claim certifications or insurance you do not hold. Say what you did and what you did not do, in writing. Consider professional liability insurance and take legal advice before accepting paid work.

Scams aimed at IT people. Be careful with a recruiter or client who asks you to run unknown code or install software on your main machine, to pay a fee before work starts, to cash a cheque and send money back, or to hand over a root or admin password for a first small job. Do the work in a clean test environment and ask for a limited account.

Each idea page points back to this check. The table "Rules to check yourself" at the end of the book names who to ask in each country.

How each idea page works

What this is	One or two plain sentences on what you would actually do.
Badges	**Green** easy, low cost or low risk **Amber** some skill, money or care needed **Red** hard, costly or risky
At a glance	How soon money might arrive, what it can earn (only with a source), weekly hours, how crowded it is.
Is this a good fit for you?	Honest notes on why your IT background helps, including what to do if you only have an hour a day.
Your first hour	Small numbered steps so you can start today.
Copy-and-paste prompts	Four steps: research, build a sample, find your first client or listing, then check and improve. Each prompt has a short note on what a good answer looks like.
Keep it	A reusable helper to save inside your AI tool.
Stay safe	The side-work and client-access check and the red flags for that idea.
Where this comes from	Our sources, whether we opened them, and the date we read them.

Using the prompts well

1. **Fill in the brackets.** Anything in [CAPITALS AND BRACKETS] is for you to replace with your own details.
2. **Keep customer and employer data out.** Never paste customer data, passwords, API keys, private logs or your employer's code into an AI tool. Use only what you made yourself or what is public, and strip names first.
3. **Check everything.** AI can sound sure and be wrong. Open the sources it names. Never publish or send a fact you have not checked yourself.
4. **Be honest about using AI.** If a client or platform asks, say how you used it. Some platforms and client briefs require a label or ban it outright.
5. **Treat [CHECK] as a to-do.** Where a prompt or an answer marks something [CHECK], it means "confirm this on an official page before you rely on it".

The map of all 50 ideas

Every idea in one picture. The colour shows which part of the book it lives in.

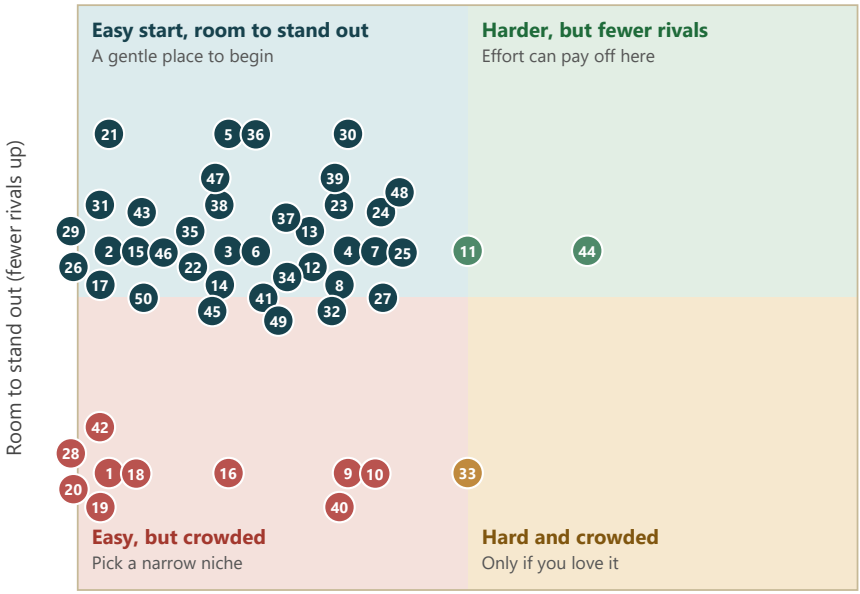
Words that IT people can sell · Small tools you can ship · Templates and kits for developers · Fixed-price jobs for small businesses · Light-touch security, always with written permission · Teach it, package it, polish it · Let the computer do the boring bits

1 README and Quick-Start Rewrites for...	2 Runbook Writing for Tiny IT Teams	3 Home-Lab Hardware Buyer Guides	4 Post-Incident Write-Ups and...	5 Slack Thread to Confluence or...
6 API Quick-Start Guides and Sample...	7 Security Questionnaire Answer Banks...	8 A Chrome Extension for One Narrow Job	9 A Tiny WordPress Plugin for One...	10 Jira Automation Rule Packs
11 Onboarding and offboarding script...	12 Cloud bill reduction review	13 Slack or Teams bot for one team ritual	14 Log and config parser for one niche...	15 VS Code snippet and extension packs...
16 Starter kits for one specific stack	17 Support reply macro library for...	18 SaaS subscription and renewal...	19 Onboarding and offboarding...	20 Incident response plan template for...
21 Network and Rack Documentation...	22 Technical Interview Question Banks...	23 Monthly Tech Health Check for...	24 Microsoft 365 Set-Up and Tidy-Up...	25 Backup Restore-Test Report for...
26 Uptime Monitoring Bundle for Small...	27 Fixed-Price Zapier or Make...	28 Booking and Job Request Form Set-Up...	29 Password Manager and MFA Rollout...	30 Licence and Admin-Account Audit
31 Phishing Awareness Mini-Session for...	32 Home-Office Network Setup Checklist...	33 Authorised Website Hygiene Review...	34 Excel Macro to Python Script...	35 Secure Coding Checklists for Small...
36 Tabletop Exercise Facilitation	37 Plain-Language Data Map Worksheet...	38 A Micro-Course for Office Staff on...	39 Cron job and alert audit for small...	40 Certification Study Guides From...
41 A Niche Dev Newsletter with Sponsor...	42 Screencast Tutorials for a Niche Tool	43 Editing for Developer-Tool Startup...	44 Second-Opinion Review of AI-Written...	45 Rules Files and Prompt Packs for...
46 Test-Case Writing for Small QA Teams	47 Messy CSV to Clean Database	48 Self-Hosted n8n	49 Self-Hosted n8n	50 Self-Hosted n8n

Hard does not mean bad

On each idea page you will see badges for difficulty, cost and risk. A **red badge is not a stop sign**. It is a price tag.

In most markets, the harder an idea is to start, the fewer people finish it. So a harder idea can mean **less competition**, and a skill that is **harder for newcomers to copy**. An easy idea has the opposite problem: anyone can start it tomorrow, so you have to stand out.



Effort to start: difficulty, cost and risk (easier on the left, harder on the right)

Each dot is one idea, numbered as in the menu. Positions are our own rough judgement from the notes on each idea page, not a measurement, and not a promise of income.

- **Top left:** easy and with room. A friendly first project.
- **Top right:** hard but quiet. Worth it if the skill excites you.
- **Bottom left:** easy but busy. Choose a narrow niche.
- **Bottom right:** hard and busy. Only if you truly love it.

Contents

Part 1: Words that IT people can sell

1. README and Quick-Start Rewrites for Commercial Open Source Projects 19
2. Runbook Writing for Tiny IT Teams 25
3. Home-Lab Hardware Buyer Guides 30
4. Post-Incident Write-Ups and Status-Page Copy 35
5. Slack Thread to Confluence or Notion Article Batches 40
6. API Quick-Start Guides and Sample Code for Small API Vendors 45
7. Security Questionnaire Answer Banks for Small Software Vendors 50

Part 2: Small tools you can ship

8. A Chrome Extension for One Narrow Job 56
9. A Tiny WordPress Plugin for One Small-Business Problem 61
10. Jira Automation Rule Packs: a Done-for-You Setup Service 66
11. Onboarding and offboarding script packs for Microsoft 365 and Google Workspace 71
12. Cloud bill reduction review 77
13. Slack or Teams bot for one team ritual 82
14. Log and config parser for one niche application 87
15. VS Code snippet and extension packs for a niche framework 92

Part 3: Templates and kits for developers

16. Starter kits for one specific stack 98
17. Support reply macro library for help desks 103
18. SaaS subscription and renewal tracker sheet 108
19. Onboarding and offboarding checklist pack 113
20. Incident response plan template for 10-person businesses 118
21. Network and Rack Documentation Workbook for Small Installers 123
22. Technical Interview Question Banks for One Role 128

Part 4: Fixed-price jobs for small businesses

23. Monthly Tech Health Check for 5-Person Offices 134
24. Microsoft 365 Set-Up and Tidy-Up for Small Firms 139
25. Backup Restore-Test Report for Small Offices 144
26. Uptime Monitoring Bundle for Small Agencies 149
27. Fixed-Price Zapier or Make Automations for One Vertical 154
28. Booking and Job Request Form Set-Up for Tradies 159
29. Password Manager and MFA Rollout for Small Teams 164

30. Licence and Admin-Account Audit 169

Part 5: Light-touch security, always with written permission

31. Phishing Awareness Mini-Session for Staff 175
32. Home-Office Network Setup Checklist Service 181
33. Authorised Website Hygiene Review of the Client's Own Site 186
34. Excel Macro to Python Script Conversion 192
35. Secure Coding Checklists for Small Teams (Pinned to One Stack, Including AI-Generated Code) 197
36. Tabletop Exercise Facilitation (An Add-On to the Incident Response Plan Template, Idea 20) 202
37. Plain-Language Data Map Worksheet for Small Australian Businesses 207

Part 6: Teach it, package it, polish it

38. A Micro-Course for Office Staff on One Tool 213
39. Cron job and alert audit for small firms 218
40. Certification Study Guides From Official Exam Objectives 223
41. A Niche Dev Newsletter with Sponsor Slots 228
42. Screencast Tutorials for a Niche Tool 234
43. Editing for Developer-Tool Startup Blogs 239
44. Second-Opinion Review of AI-Written Code for Non-Technical Founders 244
45. Rules Files and Prompt Packs for Coding Assistants 250
46. Test-Case Writing for Small QA Teams 255
47. Messy CSV to Clean Database 260

Part 7: Let the computer do the boring bits

48. Self-Hosted n8n: Release Notes to Changelog and Social Drafts 266
49. Self-Hosted n8n: Invoice and Receipt Capture 271
50. Self-Hosted n8n: Contact Form Follow-Up 276

The whole menu at a glance

A quick way to pick: start with green. Each little picture shows the level at a glance: three bars for difficulty, coins for cost, a shield for risk.

#	Idea	Level	Cost	Risk
1	README and Quick-Start Rewrites for Commercial Open Source Projects	Beginner	Free	Low
2	Runbook Writing for Tiny IT Teams	Beginner	Free	Low
3	Home-Lab Hardware Buyer Guides	Beginner	Free to Low	Low to Medium
4	Post-Incident Write-Ups and Status-Page Copy	Intermediate	Free	Medium
5	Slack Thread to Confluence or Notion Article Batches	Beginner	Free	Low to Medium
6	API Quick-Start Guides and Sample Code for Small API Vendors	Intermediate	Free	Low
7	Security Questionnaire Answer Banks for Small Software Vendors	Intermediate	Free	Medium
8	A Chrome Extension for One Narrow Job	Intermediate	Low	Medium
9	A Tiny WordPress Plugin for One Small-Business Problem	Intermediate	Low	Medium
10	Jira Automation Rule Packs: a Done-for-You Setup Service	Intermediate	Free to Low	Medium
11	Onboarding and offboarding script packs for Microsoft 365 and Google Workspace	Intermediate	Low	High
12	Cloud bill reduction review	Intermediate	Free	Medium
13	Slack or Teams bot for one team ritual	Intermediate	Low	Medium

14	Log and config parser for one niche application	Intermediate	Free	Low
15	VS Code snippet and extension packs for a niche framework	Beginner	Free	Low
16	Starter kits for one specific stack	Intermediate	Low	Low
17	Support reply macro library for help desks	Beginner	Free	Low
18	SaaS subscription and renewal tracker sheet	Beginner	Free	Low
19	Onboarding and offboarding checklist pack	Beginner	Free	Low
20	Incident response plan template for 10-person businesses	Beginner	Free	Low
21	Network and Rack Documentation Workbook for Small Installers	Beginner	Free	Low
22	Technical Interview Question Banks for One Role	Beginner	Free	Low to Medium
23	Monthly Tech Health Check for 5-Person Offices	Intermediate	Low	Medium
24	Microsoft 365 Set-Up and Tidy-Up for Small Firms	Intermediate	Free	Medium
25	Backup Restore-Test Report for Small Offices	Intermediate	Free	Medium
26	Uptime Monitoring Bundle for Small Agencies	Beginner	Free	Low
27	Fixed-Price Zapier or Make Automations for One Vertical	Intermediate	Free	Low to Medium
28	Booking and Job Request Form Set-Up for Tradies	Beginner	Free	Low
29	Password Manager and MFA Rollout for Small Teams	Beginner	Free	Low
30	Licence and Admin-Account Audit	Intermediate	Free	Medium

31	Phishing Awareness Mini-Session for Staff	Beginner	Free	Low
32	Home-Office Network Setup Checklist Service	Beginner to Intermediate	Free to Low	Medium
33	Authorised Website Hygiene Review of the Client's Own Site	Intermediate	Free	High
34	Excel Macro to Python Script Conversion	Intermediate	Free	Medium
35	Secure Coding Checklists for Small Teams (Pinned to One Stack, Including AI-Generated Code)	Intermediate	Free	Low
36	Tabletop Exercise Facilitation (An Add-On to the Incident Response Plan Template, Idea 20)	Intermediate	Free	Low
37	Plain-Language Data Map Worksheet for Small Australian Businesses	Intermediate	Free	Medium
38	A Micro-Course for Office Staff on One Tool	Intermediate	Low	Low
39	Cron job and alert audit for small firms	Intermediate	Free	Medium
40	Certification Study Guides From Official Exam Objectives	Beginner to Intermediate	Free	Medium
41	A Niche Dev Newsletter with Sponsor Slots	Intermediate	Free	Low
42	Screencast Tutorials for a Niche Tool	Beginner	Free	Low
43	Editing for Developer-Tool Startup Blogs	Beginner	Free	Low
44	Second-Opinion Review of AI-Written Code for Non-Technical Founders	Advanced	Free	High
45	Rules Files and Prompt Packs for Coding Assistants	Intermediate	Free	Low

46	Test-Case Writing for Small QA Teams	Beginner	Free	Low
47	Messy CSV to Clean Database	Beginner	Free	Medium
48	Self-Hosted n8n: Release Notes to Changelog and Social Drafts	Intermediate	Free	Medium
49	Self-Hosted n8n: Invoice and Receipt Capture	Intermediate	Free	Low
50	Self-Hosted n8n: Contact Form Follow-Up	Beginner	Free	Low

PART 1

Words that IT people can sell

Writing jobs that use what you already know: READMEs, runbooks, incident write-ups, API guides and security answers.

1 2 3 4 5 6 7

1 README and Quick-Start Rewrites for Commercial Open Source Projects

Think of yourself as the first-time user the maintainers forgot. You rewrite the README, install guide and quick-start so a new person gets a working result in five minutes, and you prove it by running every command.

Difficulty Beginner | **Cost** Free | **Risk** Low | **Crowd** High on Fiverr and Con

The balance. Easy to start, which is why many people do it. You will need a narrow niche or a clear edge.

At a glance

How soon might money arrive?	Weeks to months; many first samples never get a reply
What can it earn?	No reliable income figure found. Fiverr and Contra listings seen in search results at about $20 to $180 per README, but those are asking prices, not earnings.
Time each week and what AI does	3 to 6 hrs per week; AI drafts, you run every command yourself
Does it grow?	Mostly one-off jobs; repeat work comes from the same company's other repos
How crowded is it?	High on Fiverr and Contra; a focus on companies with docs budgets helps
Where it fits	 US  UK  AU  CA (English-language projects)

Skill level	Beginner
Money to start	Free
What can go wrong	Low

Is this a good fit for you?

You already read a lot of bad READMEs, which is the skill. The work is small, remote and needs no tooling beyond a terminal. Hobby maintainers rarely have budget, so aim at commercial open source projects and dev-tool startups, which are more likely to pay.

Only 1 hour a day? Spend 40 minutes testing one project's install steps on a clean machine or container and noting where you got stuck. Use 20 minutes to rewrite just that section.

Your first hour

About an hour in all, one small step at a time

1. Pick three tools you actually use and read their open issues about setup.
2. Choose one with a company behind it.
3. Rewrite one section, such as install, and test each command in a fresh environment.
4. Send it as a small pull request or a short note offering a fixed-price rewrite. Never promise the maintainer will merge it.

Copy-and-paste prompts

Paste each one into any AI chat tool. Replace every [BRACKET] with your own details first. Read the answer critically before you use it.

Step 1 Research

```
I want to offer paid README and quick-start rewrites to [TYPE OF
PROJECT, e.g. open source developer tools with a company behind
them] in [YOUR COUNTRY]. List the 8 most common problems in a weak
README (for example missing prerequisites, commands that no longer
work, no expected output). For each, say how a reader can spot it in
2 minutes. Do not invent prices or demand figures. Tell me to check
current prices on the marketplaces myself.
```

What to look for. A list of checks you could run on a real README in minutes, with no made-up market numbers.

Step 2 Build a sample

```
Here is the install section of a public README, plus my notes on
where I got stuck: [PASTE THE PUBLIC TEXT AND YOUR NOTES]. Rewrite
it for a first-time user on [YOUR OS]. Put prerequisites first, use
numbered steps, and show the expected output of each command only if
the text above states it. Mark anything you cannot confirm as
[CHECK]. Do not invent flags, versions or URLs.
```

What to look for. Every command and version should match the original source; run each one yourself and fix any [CHECK] items.

Step 3 Find first client or listing

```
Write a short, friendly message (under 120 words) to a maintainer of
[PROJECT NAME]. Say I tried the quick-start on [YOUR OS], list the
[NUMBER] places I got stuck, and offer a fixed-price rewrite of
[SECTION]. Do not claim experience, past clients or results I do not
have. Do not promise the change will be accepted. Leave [PRICE] and
[YOUR NAME] as placeholders.
```

What to look for. An honest message that names specific problems you really hit, not generic praise.

Step 4 Check and improve

```
Review my rewritten README section below against the original. List
any step that a new user could not follow, any command that may
differ by operating system, and any claim not supported by the
original. Do not add new features or commands. Then give me a 6-
point checklist to run before I submit future rewrites. Original:
[PASTE ORIGINAL] Mine: [PASTE MINE]
```

What to look for. Gaps a beginner would hit; confirm each point by running the steps again from scratch.

Keep it: your own reusable helper

You have just done the job once. Now save it as a small helper inside your own AI tool, so next time it takes two minutes. Name it **My README Rewrite Helper** and paste this into its instructions.

Save Instructions for your helper

```
You are a careful documentation assistant for an IT
professional who rewrites READMEs and quick-start guides for
software projects and is paid for it. The user is the author
and tests every command.

Ask, one short question at a time: (1) Which project and which
section? (2) Which operating system will a new user have? (3)
What did you test yourself and what happened? Then ask the user
to paste the public text.

Output format:
1. The rewritten section with prerequisites first and numbered
steps.
2. A table of changes: old wording, new wording, one-line
reason.
3. A "[CHECK]" list for any command, flag or version you cannot
confirm from the text provided.

Rules: Use only the text and notes pasted. Never invent
commands, flags, versions, URLs or expected output. Do not use
text from the user's employer or any private repository. Remind
the user to read the project's licence and contributing guide,
to say in the pull request that the work is paid, and never to
promise a merge. Plain, friendly language. End every reply
with: "Run every command yourself before sending. You are the
author."
```

Where. Claude: create a Project and paste into Project instructions. ChatGPT: create a Project or custom GPT and paste into Instructions. Gemini: create a Gem and paste into Instructions.

Each time you use it. Paste the public README section and write: "Task = rewrite install section for Ubuntu, I tested steps 1 to 3 and step 4 failed." Then run every command in a clean environment.

Check yourself, every time.

- Run every command in a fresh environment and compare with the output the rewrite promises.
- Confirm the project's licence and contribution rules allow your change and that no employer material was used.

Your file

A one-page checklist for testing and rewriting a README install section.

File: `readme-rewrite-checklist` (a suggested name for your own copy, which you build with the prompts above).

Stay safe

The rules to know. Check the repository's licence and contributing guide before submitting anything; some projects ask for a signed contributor agreement. If you are paid, say so in the pull request. Do not paste your employer's internal docs into any AI tool. AI-written text can quietly invent flags and commands, so run everything. GitHub Sponsors exists for funding maintainers, but it is not a client list.

Red flags. Maintainers who want "free work for exposure", or anyone asking you to run unknown scripts on your main machine.

Where this comes from.

Source check: opened and read on 01/11/2026 (market and price links seen in search results, not yet opened)

About GitHub Sponsors, read 01/11/2026. Fiverr and Contra README listings: seen in search results, not re-opened, check before relying on them.

2 Runbook Writing for Tiny IT Teams

A runbook is a plain "how to do this task" page for the day the one person who knows it is away. You interview the part-time admin, write 5 to 10 short runbooks (restore a backup, add a new starter, renew a certificate, rotate a key), and test each on someone else.

Difficulty Beginner **Cost** Free **Risk** Low

Crowd Medium for templates

The balance. A gentle start with room to stand out. A good place to begin.

At a glance

How soon might money arrive?	Weeks; the first client often comes from your own network
What can it earn?	No reliable income figure found. Runbook templates were seen priced at about $19 to $99 (Gumroad and similar), which are asking prices, not earnings.
Time each week and what AI does	4 to 8 hrs per week; AI tidies drafts, the client's real answers drive the content
Does it grow?	Repeat reviews and updates possible, but it is mostly paid time
How crowded is it?	Medium for templates; lower for interview-based, done-for-you work
Where it fits	 US  UK  AU  CA
Skill level	Beginner
Money to start	Free

What can go wrong	Low

Is this a good fit for you?

You know which tasks nobody has written down. The interviews fit into evenings, and you need no special software. Sell the interview-based version for a 5 to 20 person firm rather than a template, since templates are cheap and common.

Only 1 hour a day? Spend 30 minutes interviewing someone about one dreaded task. Spend 30 minutes turning the notes into a numbered runbook.

Your first hour

About an hour in all, one small step at a time

1. Ask a friend or past colleague who runs IT for a small firm for a 30 minute chat.
2. Pick one task they dread.
3. Write the runbook in numbered steps with screenshots only if the client agrees.
4. Ask a second person to follow it cold and fix where they stumble.

Copy-and-paste prompts

Paste each one into any AI chat tool. Replace every [BRACKET] with your own details first. Read the answer critically before you use it.

Step 1 Research

```
I plan to write runbooks for small IT teams (5 to 20 staff) in [YOUR
COUNTRY]. List 10 common tasks such a team has to repeat, such as
[EXAMPLES]. For each, list 5 questions I should ask the admin so the
runbook is accurate. Do not invent statistics or prices. Remind me
never to ask for passwords or keys.
```

What to look for. Questions about who, when and what goes wrong, not questions asking for secrets.

Step 2 Build a sample

```
Turn these interview notes into a runbook for [TASK NAME]: [PASTE
NOTES WITH ALL SECRETS REMOVED]. Use: purpose, who can do it,
prerequisites, numbered steps, how to confirm it worked, what to do
if it fails, who to call. Mark anything unclear as [CHECK] and do
not invent steps, commands or tool names.
```

What to look for. Each step should come from the notes; test it yourself or with a second person.

Step 3 Find first client or listing

```
Write a friendly message (under 120 words) to the owner of a [TYPE
OF SMALL BUSINESS] offering a fixed set of [NUMBER] runbooks,
written from a short interview and tested by a second person. Say I
will not need passwords or system access. Do not claim past clients
or qualifications I do not have. Leave [PRICE] and [YOUR NAME] as
placeholders.
```

What to look for. A clear promise of what they get, plus the line that you need no access.

Step 4 Check and improve

```
Review this runbook as a stranger who has never seen the system:
[PASTE RUNBOOK]. List steps that are vague, missing a prerequisite
or risky without a backup. Check that no secret, real address or
personal data appears. Do not rewrite it; just list issues and
suggest 3 fixes.
```

What to look for. Hidden assumptions, plus any secret you may have missed.

Keep it: your own reusable helper

You have just done the job once. Now save it as a small helper inside your own AI tool, so next time it takes two minutes. Name it **My Runbook Writer** and paste this into its instructions.

Save Instructions for your helper

```
You are a runbook-writing assistant for an IT professional who
writes runbooks for small IT teams from interviews. The user is
the author, and the client confirms every step.

Ask, one short question at a time: (1) What task is this
runbook for? (2) Who will use it, and how technical are they?
(3) Have you removed all passwords, keys, tokens, real IP
addresses and personal data from the notes? Then ask the user
to paste the cleaned interview notes.

Output format:
1. A runbook with: purpose, who can do it, prerequisites,
numbered steps, how to confirm success, what to do on failure,
who to contact.
2. A "[CHECK]" list of steps that are unclear or missing from
the notes.
3. A "Test with a second person" list of 5 things to watch for.

Rules: Use only the notes pasted. Never invent steps, commands,
tools or names. Never include a secret; write "stored in
[PASSWORD MANAGER NAME]". Do not use any employer's runbooks.
Remind the user to get the client's consent before using AI on
their notes. Plain, friendly language. End every reply with:
"Ask someone else to follow this cold before you hand it over."
```

Where. Claude: create a Project and paste into Project instructions. ChatGPT: create a Project or custom GPT and paste into Instructions. Gemini: create a Gem and paste into Instructions.

Each time you use it. Paste the cleaned notes and write: "Task = renew a TLS certificate, audience = office manager."

Check yourself, every time.

- Search the runbook for any secret, real address or personal data before you send it.

- Have a second person follow it and fix every step where they stumble.

Your file

A blank runbook template and an interview question sheet.

File: `runbook-template-pack` (a suggested name for your own copy, which you build with the prompts above).

Stay safe

The rules to know. Never put passwords, keys, tokens or real IP addresses in a runbook; write "stored in [PASSWORD MANAGER NAME]" instead. Do not copy runbooks from your own employer. Tell the client before using AI on their notes, and scrub names and secrets first. CISA's small business guidance lists tested backups, patching and incident response plans, which makes good runbook topics. Runbooks are advice, not a guarantee; say so in your terms.

Red flags. A client who wants you to log into production "just to check", or to hold their admin credentials. Decline both.

Where this comes from.

Source check: opened and read on 01/11/2026 (market and price links seen in search results, not yet opened)

CISA Incident Response Plan Basics, CISA cyber guidance for small businesses, both read 01/11/2026. Template prices: seen in search results, not re-opened, check before relying on them.

3 Home-Lab Hardware Buyer Guides

A short PDF that helps a hobbyist choose a second-hand mini PC for a home server: what to look for, what to avoid, and how to test it before buying.

Difficulty Beginner **Cost** Free to Low **Risk** Low to Medium **Crowd** Unknown

The balance. A gentle start with room to stand out. A good place to begin.

At a glance

How soon might money arrive?	Months; no confirmed buyers
What can it earn?	No reliable income figure found.
Time each week and what AI does	3 to 5 hrs per guide; AI structures it, you must verify every spec
Does it grow?	A PDF can sell repeatedly, but needs new guides as hardware changes
How crowded is it?	Unknown; free forum and YouTube guides are everywhere
Where it fits	US UK AU CA
Skill level	Beginner
Money to start	Free to Low
What can go wrong	Low to Medium: wrong specs, affiliate disclosure

Is this a good fit for you?

You already know which second-hand mini PCs run quietly, sip power and survive a hypervisor. That hands-on knowledge is the product. Since free content is plentiful, this idea works best as a small, honest side project, not a plan.

Only 1 hour a day? Spend 40 minutes checking specs for one model from the maker's own pages. Spend 20 minutes drafting one section.

Your first hour

About an hour in all, one small step at a time

1. Choose one model line you have really used.
2. List what you tested yourself, such as idle power draw and fan noise.
3. Check every spec against the maker's own page.
4. Draft one page and keep it as a sample.

Copy-and-paste prompts

Paste each one into any AI chat tool. Replace every [BRACKET] with your own details first. Read the answer critically before you use it.

Step 1 Research

```
I want to write a buyer guide for second-hand mini PCs used as home
servers, for readers in [YOUR COUNTRY]. List the 8 questions a first-
time buyer asks (for example power use, noise, RAM limits, storage
options). Do not state prices, model names or specs as facts; ask me
to check those on the manufacturer's page.
```

What to look for. Reader questions only, with no invented specs.

Step 2 Build a sample

```
Create an outline for a 6 page buyer guide about [MODEL FAMILY] used
as a home server. Use these facts that I measured myself: [YOUR
MEASUREMENTS]. Mark every other claim as [CHECK] and do not invent
benchmarks, prices or compatibility claims.
```

What to look for. Only your measured facts should appear as facts.

Step 3 Find first client or listing

```
Write an honest listing description (under 120 words) for a PDF
buyer guide about [TOPIC]. State what is inside, who it is for, and
that the specs were checked against manufacturer pages on [DATE].
Say affiliate links are used only if they really are. Do not promise
savings or results.
```

What to look for. A plain description with a disclosure line if you use affiliate links.

Step 4 Check and improve

```
Check my draft below for claims I cannot back up, specs that need a
source, missing affiliate disclosure and safety warnings about used
electronics. List each issue and suggest wording. Do not add specs.
Draft: [PASTE DRAFT]
```

What to look for. Every spec has a source; unsupported claims are cut.

Keep it: your own reusable helper

You have just done the job once. Now save it as a small helper inside your own AI tool, so next time it takes two minutes. Name it **My Home-Lab Guide Helper** and paste this into its instructions.

Save Instructions for your helper

```
You are a writing assistant for an IT professional who writes
short buyer guides about second-hand hardware for home servers.
The user has tested the hardware and owns the facts.

Ask, one short question at a time: (1) Which model family? (2)
What did you measure yourself, such as idle power, noise,
temperature? (3) Will you use affiliate links? Then ask the
user to paste their notes.

Output format:
1. A guide outline with plain headings and short paragraphs.
2. A table of every factual claim with its source: "measured by
me" or "manufacturer page, link to check".
3. A "[CHECK]" list for any claim without a source.

Rules: Use only facts the user gives. Never invent specs,
benchmarks, prices or compatibility. Do not copy other sites'
tables or photos. If affiliate links are used, add a clear
disclosure line. Include a safety note about used electronics.
Do not give advice about scanning or probing networks the user
does not own. Plain, friendly language. End every reply with:
"Check every spec on the manufacturer's page before publishing."
```

Where. Claude: create a Project and paste into Project instructions. ChatGPT: create a Project or custom GPT and paste into Instructions. Gemini: create a Gem and paste into Instructions.

Each time you use it. Paste your measurements and write: "Model = [MODEL], affiliate links = no." Then verify specs yourself.

Check yourself, every time.

- Make sure every number in the guide is either your own measurement or checked on the maker's page.
- Confirm that affiliate links, if any, are disclosed clearly.

Your file

A one-page checklist for inspecting a second-hand mini PC before buying.

File: `homelab-buyer-checklist` (a suggested name for your own copy, which you build with the prompts above).

Stay safe

The rules to know. Only write what you have tested or can source. If you add affiliate links, disclose them clearly (FTC rules in the US, ASA in the UK, ACCC in Australia, Competition Bureau in Canada; check this yourself). Do not copy specs tables or photos from other sites. Stay honest about used hardware risk. Do not include advice on scanning or probing networks you do not own.

Red flags. Paid "reviews" that expect only positive comments, or pages demanding a login to see claimed earnings.

Where this comes from.

Source check: opened and read on 01/11/2026 (pages cited in the card above) No page opened for this idea (read 01/11/2026). Demand is not verified; check search volume and existing guides yourself before investing time.