

Build Your Own AI Seller Copilot

From Zero to Live Link
(No Coding Required)

by AI Soft Tech Solution

*Turn your hard-won selling playbooks into an AI chat app
that answers like a specialist - built by you, in a weekend.*

1. The Pain

If you sell on Amazon, TikTok Shop, Shopify, or anywhere online, you know the monthly bill. Seller dashboards and research tools charge **\$49 to \$99 a month** - some charge more once you add extra marketplaces or team seats.

They are good at one thing: showing you numbers. Sales graphs. Keyword ranks. Competitor prices. Review counts.

Here is what they never do: they never answer the question that actually matters - **"What should I DO next?"**

Should you raise the price or lower it? Is this product worth restocking? Your listing gets clicks but no sales - what do you change first? A dashboard shows you the symptom. It never prescribes the treatment.

That missing layer has a name: **strategy**. And strategy is the expensive gap. A decent e-commerce consultant charges **\$150 to \$300 an hour**. A monthly retainer starts around **\$1,000**. Most small sellers simply go without - and make every decision by gut feel.

But here is the thing nobody tells you: **you already own the strategy**. It is in your head. Every pricing rule you have learned, every listing mistake you will never repeat, every supplier red flag - that is a playbook. It just lives in the one place no tool can reach: your memory.

This guide fixes that. You are going to take the playbooks out of your head, put them into an AI chat app, and get yourself a copilot that answers like a specialist instead of a generic chatbot. No coding required.

2. What You Will Build

Where this idea comes from

This guide is built on a simple, proven idea: packaging expert knowledge into plain-text **"skill" files** that teach an AI assistant to think like a product researcher, a pricing strategist, or a listing optimizer - no software, no code, just clear instructions the AI follows.

This guide takes that idea and makes it **yours**. Instead of generic, one-size-fits-all instructions, your copilot runs on **your** playbooks - your rules, your numbers, your niche. Everything you need is in the steps below.

You will build a small **web app** - a page that opens in any browser, on your phone or laptop. (A web app is simply a website that works like an app: no download, just a link.) It has two parts:

- **A chat box.** You type a question in plain English. It answers.
- **A sidebar listing your "skills":** Product Research, Pricing, Listing Optimization, and whatever else you teach it.

A **skill** is just a plain-text file with your rules. For example, your Pricing skill might say: "Never price below \$14.99 - below that, ad costs eat the margin. Always check the top 3 competitors first. If all three are under \$19.99, compete on bundle value, not price."

When you ask, "What should I price my new phone case at?", the app reads your Pricing skill first, then answers like **you** would - not like a generic AI.

What it looks like when you are done

A live link like `yourstore-copilot.lovable.app`. You open it and see your app name at the top, your skill list on the left, and a chat box that greets you. It works on your phone. You can send the link to a business partner and they can use it too.

What it is NOT - honest limits

- It is **not a live data dashboard**. It will not pull real-time sales numbers or keyword ranks from Amazon or TikTok Shop. Those need paid data feeds.
- It does **not replace market-data tools**. Think of it this way: the dashboard tells you WHAT is happening; your copilot tells you what TO DO about it.
- It is only as smart as the playbooks you give it. Thin playbooks, thin answers. (Step 5 shows you how to test this.)

3. Pick Your Coding Agent

A **coding agent** is an AI that does not just chat - it builds. You describe what you want in plain English ("build me a chat app with a sidebar"), and the agent writes the code, fixes its own errors, and publishes the app. You never see the code. You just chat, click, and publish.

The biggest names in AI all have one now - alongside the app builders. Here is an honest comparison for a non-technical founder (facts checked October 2026):

Tool	Best for	Ease 1-5	Free tier	Watch out
Claude (Anthropic)	The strongest all-round coder; Claude Code agent works in terminal, web and desktop	2	Free tier with daily limits	Claude Code lives in a terminal - powerful, but built for developers
ChatGPT + Codex (OpenAI)	Coding agent inside tools you may already use; everyday mode for non-coders	3	Free tier includes quick Codex tasks	Burns through usage fast on long builds - watch your limits
Google Antigravity	Google's agent-first IDE; agents plan, code, test and verify in parallel	2	Free public preview	A desktop IDE you install - still built for developers
Lovable (RECOMMENDED)	Easiest prompt-to-app loop for non-technical founders; one-click publish	5	Free: 5 credits/day	Credits burn on heavy iteration - fine for this project
v0 by Vercel	The most polished-looking interfaces	4	Free: small monthly credits	UI-focused - weaker at full apps with databases
Bolt.new	Fast full-stack builds, runs in your browser	4	Free: daily token allowance	No built-in backend - needs external setup for data
Replit	Building + database + publishing in one place	3	Free: limited daily credits	Free apps sleep when idle; the editor looks like code

Verdict: start with Lovable

The honest ranking: the giants - Claude, ChatGPT with Codex, and Google Antigravity - are the best pure coders on this list. But they live in terminals and IDEs: tools built for developers. For a

non-technical founder building a first web app, the right question is not "which codes best" but "which gets me from an idea to a live link with zero setup." That is Lovable: one browser tab, chat-first prompting, one-click publish. Every step in this guide uses Lovable. (Already paying for ChatGPT Plus? Codex in the ChatGPT web app is the friendliest giant - a fair second choice.)

What to avoid

Do not start with **raw APIs** (a way for two apps to exchange data automatically - powerful, but needs setup, keys, and debugging) or with terminal/IDE-based agents like Claude Code and Antigravity as your **first** build. You will get stuck before you have anything to show. Learn the ropes in Lovable first; graduate to the giants when you outgrow them.

4. The Build - Step by Step

Work through these steps in order. Each step has a **copy-paste prompt** in a gray box - paste it verbatim into your coding agent's chat. Where a prompt needs your details, you get a **filled example** (TikTok Shop phone accessories) and a **blank version**. Copy the filled one first to see how it works, then use the blank one with your own details.

Step 1 - Define your niche and your 10 questions

Before touching any tool, decide what your copilot knows. Paste this planning prompt into Lovable's chat. It will interview you and turn your answers into a skill list.

COPY-PASTE PROMPT - filled example (TikTok Shop phone accessories):

I sell phone accessories (cases, chargers, stands) on TikTok Shop, doing about \$3,000 a month. I want to build an AI copilot trained on my own selling playbooks.

Interview me with 10 questions, one at a time, about how I make decisions: how I pick products, how I price, how I write listings, how I handle returns, how I choose creators for affiliate deals, and what mistakes I never repeat.

After my 10 answers, give me a numbered list of skills my copilot needs (for example: Product Research, Pricing, Listing Optimization, Affiliate Outreach, Returns Handling), with one line under each saying what that skill covers.

COPY-PASTE PROMPT - blank version (fill in the CAPS parts, then paste):

I sell [YOUR PRODUCTS] on [YOUR PLATFORM: Amazon / TikTok Shop / Shopify / other], doing about \$[YOUR MONTHLY REVENUE] a month. I want to build an AI copilot trained on my own selling playbooks.

Interview me with 10 questions, one at a time, about how I make decisions: how I pick products, how I price, how I write listings, how I handle [RETURNS / CUSTOMER MESSAGES / ADS - pick what matters in your business], and what mistakes I never repeat.

After my 10 answers, give me a numbered list of skills my copilot needs, with one line under each saying what that skill covers.

Answer all 10 questions honestly - short answers are fine. **Save the skill list** it gives you at the end. You will paste it into Step 2.

Step 2 - Scaffold the app

"**Scaffold**" just means: build the empty shell of the app - the screens and buttons - before the brains go in. Paste this as your first build prompt.

COPY-PASTE PROMPT - filled example:

Build a web app called "CaseKart Seller Copilot".

Layout:

- A left sidebar listing my skills: Product Research, Pricing, Listing Optimization, Affiliate Outreach, Returns Handling. Each skill is clickable and shows a short description when clicked.
- A main chat area on the right with an input box at the bottom and a welcome message: "Ask me anything about selling phone accessories on TikTok Shop."
- A clean, professional look: white background, dark text, blue accent color.

For now the chat can give a placeholder reply like "Skill system not connected yet." I will connect the AI brain in the next step.

COPY-PASTE PROMPT - blank version:

Build a web app called "[YOUR STORE] Seller Copilot".

Layout:

- A left sidebar listing my skills: [PASTE YOUR SKILL LIST FROM STEP 1]. Each skill is clickable and shows a short description when clicked.
- A main chat area on the right with an input box at the bottom and a welcome message: "Ask me anything about selling [YOUR PRODUCTS] on [YOUR PLATFORM]."
- A clean, professional look: white background, dark text, [YOUR ACCENT COLOR: blue / green / orange] accent color.

For now the chat can give a placeholder reply like "Skill system not connected yet." I will connect the AI brain in the next step.

Wait for it to finish building, then click around. If something looks off, just tell it in plain English: "Make the sidebar narrower" or "Use a darker blue."

Step 3 - Write your skill library

A **playbook** is a plain-text file with your rules - the stuff from your head, written down. You do not write these by hand. Let the AI draft them from your interview answers, then **you** correct them. Do this once per skill; three solid playbooks beat ten thin ones.

COPY-PASTE PROMPT - filled example (Pricing skill):

Using my interview answers from Step 1, write the full playbook file for my "Pricing" skill.

It should be plain text with clear sections:

1. My pricing rules (numbered, specific - include my actual numbers like minimum price and margin targets)
2. How I check competitors before pricing
3. When I raise vs. lower a price
4. My bundle and discount rules
5. Mistakes I never repeat

Write it the way I talk - direct and practical. Leave anything you are unsure about marked as [CONFIRM WITH ME] instead of guessing.

COPY-PASTE PROMPT - blank version:

Using my interview answers from Step 1, write the full playbook file for my "[SKILL NAME]" skill.

It should be plain text with clear sections:

1. My rules for [SKILL NAME] (numbered, specific - include my actual numbers where relevant)
2. How I handle the most common situation in [SKILL NAME]
3. How I handle the difficult edge cases
4. Mistakes I never repeat

Write it the way I talk - direct and practical. Leave anything you are unsure about marked as [CONFIRM WITH ME] instead of guessing.

Read what it writes and **fix the numbers** - only you know your real margins. Repeat for your next skills.

Step 4 - Wire the AI chat to read the playbooks

Two terms, one line each: a **system prompt** is a hidden instruction the AI reads before every answer - like a job description. A **knowledge base** is the set of files the AI is allowed to look things up in - your playbooks.

COPY-PASTE PROMPT - filled example:

Connect the chat to my skill library.

1. Create a system prompt that says: "You are CaseKart's seller copilot. Always check the relevant skill playbook before answering. If the question matches no playbook, say so honestly and give your best general advice, marked as general advice. Never invent my numbers - if a playbook doesn't state a number, ask me instead of guessing."
2. Load my playbook files (Pricing, Product Research, Listing Optimization) as the knowledge base.
3. When I click a skill in the sidebar, show its playbook summary. When I ask a question in chat, the answer must reference which skill it used, e.g. "Based on your Pricing playbook..."

Replace the placeholder reply with the real connected chat.

COPY-PASTE PROMPT - blank version:

Connect the chat to my skill library.

1. Create a system prompt that says: "You are [YOUR STORE]'s seller copilot. Always check the relevant skill playbook before answering. If the question matches no playbook, say so honestly and give your best general advice, marked as general advice. Never invent my numbers - if a playbook doesn't state a number, ask me instead of guessing."
2. Load my playbook files ([YOUR SKILL NAMES]) as the knowledge base.
3. When I click a skill in the sidebar, show its playbook summary. When I ask a question in chat, the answer must reference which skill it used, e.g. "Based on your Pricing playbook..."

Replace the placeholder reply with the real connected chat.

Test it now: ask "What should I price my new phone case at?" The answer should mention your Pricing playbook by name. If it does not, tell your coding agent: "The chat is not reading my playbooks - fix the connection."

Step 5 - Test with 5 real questions

A copilot is only as good as its playbooks. Run these 5 tests (shown with the phone-accessories example - swap in your own products):

- 1. "I found a supplier offering magnetic chargers at \$4.20/unit. Should I add them?"
- 2. "My best-selling case dropped from #3 to #11 in search this week. What do I check first?"
- 3. "A creator with 800K followers wants a 25% commission. Deal or no deal?"
- 4. "Write me a listing title for a clear MagSafe case with a stand."
- 5. "A customer wants a refund 40 days after delivery. What is my policy?"

The rule: if an answer sounds generic - the kind any chatbot could give - your playbook is too thin. **Do not blame the AI.** Go back to Step 3, add the specific rule you wish it had used, and test again. Three rounds of this will transform the app.

Step 6 - Polish: name, colors, mobile check

COPY-PASTE PROMPT - filled example:

```
Polish the app:
1. Set the app title to "CaseKart Seller Copilot" and add a tagline under it: "Your playbooks, on demand."
2. Use blue consistently for buttons and highlights.
3. Make sure everything works on a phone screen: the sidebar should collapse into a menu button, the chat input must stay visible above the phone keyboard, and text must not overflow.
4. Add a small "How to use" hint under the welcome message: "Pick a skill on the left, or just ask a question."
```

COPY-PASTE PROMPT - blank version:

```
Polish the app:
1. Set the app title to "[YOUR STORE] Seller Copilot" and add a tagline under it: "Your playbooks, on demand."
2. Use [YOUR ACCENT COLOR] consistently for buttons and highlights.
3. Make sure everything works on a phone screen: the sidebar should collapse into a menu button, the chat input must stay visible above the phone keyboard, and text must not overflow.
4. Add a small "How to use" hint under the welcome message: "Pick a skill on the left, or just ask a question."
```

Open the preview on your phone. Tap through everything. If anything is broken, describe it in plain English and the coding agent will fix it.

Step 7 - Deploy: publish to a live link

"Deploy" just means: put your app on the internet so it has a real link you can open and share.

In Lovable: click the **Share** button (top-right), then **Publish**. Lovable gives you a live link like `yourstore-copilot.lovable.app`. That is it - your app is live.

In v0 / Bolt / Replit: look for the **Deploy** or **Publish** button - same idea, one click, you get a link.

Open the link on your phone. Send it to a friend and ask them to try one question. If it works for them, you are done with the core build.

ADVANCE - optional, only when the basics work

- **Connect live data with APIs.** An **API** is a way for two apps to exchange data automatically. Seller APIs exist: Amazon's Selling Partner API and TikTok Shop's Open API can feed real orders and listings into your app. This is the first step that feels technical - save it for version 2.
- **Add login and saved chats**, so each user gets their own history. In Lovable, prompt: "Add user login with email and save each user's chat history."
- **Charge for access.** Once other sellers want your copilot, put it behind a paywall. (Lovable and Replit both support **Stripe** - the standard way to accept card payments online.) That is how a weekend project becomes a product.

5. Troubleshooting - 5 Common Failures

1. The AI gives generic answers

Your playbooks are too thin. A playbook that says "price competitively" is useless - the AI needs YOUR numbers and YOUR rules. Open the skill file, add two specific rules with real numbers from your business, and test again. Specific input is the whole game.

2. The app looks broken on your phone

The desktop layout does not automatically fit a small screen. Paste this: "Fix the mobile layout: make the sidebar collapse into a menu button, keep the chat input visible above the keyboard, and make sure no text overflows the screen width." Then check it on your actual phone, not just the preview.

3. Publishing / deploy fails

Read the error message - then copy it and paste it back into your coding agent's chat with: "I got this error when publishing: [paste the error here]. Fix it." Nine times out of ten, the builder fixes its own error on the second try.

4. A skill does not load when clicked

This is almost always a file-name mismatch - the sidebar button points to "pricing.md" but the file is actually called "Pricing-Playbook.md". Ask your coding agent: "Check that every skill button in the sidebar loads the correct playbook file, and fix any mismatches."

5. It "forgets" the playbook mid-conversation

Long chats can push the playbook out of the AI's short-term memory. Two fixes: keep chats short (start a new chat per topic), and strengthen the system prompt with: "Re-read the relevant playbook before every answer, even in long conversations."

6. Build Checklist

Print this page or keep it open while you build. Tick each box as you go:

- ☐ **Step 1:** Answered the 10 interview questions, saved my skill list
- ☐ **Step 2:** App shell built - sidebar with skills, chat area, welcome message
- ☐ **Step 3:** At least 3 playbook files written and corrected with my real numbers
- ☐ **Step 4:** Chat connected to playbooks - answers name the skill they used
- ☐ **Step 5:** Tested with 5 real questions - fixed thin playbooks (3 rounds)
- ☐ **Step 6:** Polished - title, colors, mobile layout checked on my phone
- ☐ **Step 7:** Published - live link opens on my phone and on a friend's phone
- ☐ **Bonus:** Shared the link with one real user and watched them use it

7. What's Next

You now have something most sellers do not: a strategy brain trained on your own experience, available 24/7, at zero monthly cost.

Want it built for you instead? AI Soft Tech Solution builds custom AI apps for businesses - reply to the post or DM.