

Five AI tools you can build

Five small builds and one bonus: what to connect, what to test, and the prompts to use.

top 5 ai tools you can build right now

for product creators and PMs



feedback



competitors



prd



prototype



morning



bonus finance

01 Your own customer feedback brain

02 A little competitor stalker

03 PRD helper with actual context

04 A restrained prototype generator

05 Your morning product assistant

+ Bonus: personal finance sanity checker

Start with one annoying problem. A useful first version is a small batch, one source, or one interaction. The prompts are beside the steps where you use them.

01

Your own customer feedback brain

n8n · Claude or ChatGPT · Supabase

01 / Your own customer feedback brain



30 old support messages

The first version

30 old support messages

How to build it

1. Put 30 messages in a Sheet with a source and stable ID.
2. Use n8n to pass one message at a time to the model. Validate the returned fields before storing them.
3. Save the raw quote and structured fields in Supabase. Inspect the grouping before connecting live sources.

What you get

Original feedback beside a structured problem, feature, severity and confidence.

Missing context stays UNKNOWN. Keep source IDs so a second run does not duplicate the same message.

[Supabase tables ↗](#) · [n8n Google Sheets node ↗](#)

Use this prompt for feedback extraction

Read this customer feedback and extract the underlying problem in plain language.

Return:

product_area

problem

severity from 1 to 5

related_feature if clear

confidence from 0 to 1

If you are not sure, write UNKNOWN.

Do not invent context that is not in the original feedback.

Keep the original customer wording separately.

02

A little competitor stalker

n8n · HTTP Request · Supabase · Claude or ChatGPT

02 / A little competitor stalker



The first version

5 competitors, 2 or 3 pages each

How to build it

1. Save a short list of page URLs. Fetch and normalize their text on a schedule.
2. Store each snapshot. Compare with the previous successful fetch before calling the model.
3. Send only changed content for review. Notify on a meaningful change; keep failures separate from change alerts.

What you get

An alert about a material change, with the page link and previous version.

A blocked fetch is not a product change. The first fetch creates a baseline, not an alert.

[n8n HTTP Request ↗](#) · [n8n Schedule Trigger ↗](#)

Use this prompt for competitor changes

Compare these two versions of the same competitor page.
Ignore tiny wording changes, formatting changes and irrelevant legal text.
Tell me only if something meaningful changed for product, pricing, positioning, onboarding or target customer.
Return:
changed: true/false
summary
why_it_matters
confidence

03

PRD helper with actual context

Lovable · Supabase · Claude or ChatGPT

03 / PRD helper with actual context



1 feature, 10 comments, 1 metric note, 1 constraint

The first version

1 feature, 10 comments, 1 metric note, 1 constraint

How to build it

1. Tag feedback, notes and constraints with the same `feature_id` in Supabase.
2. Choose the feature in a small app. Fetch its evidence and send the context to a server-side model call.
3. Generate a draft, then run the engineering-review prompt. Resolve open questions before treating the PRD as ready.

Use this prompt for the PRD draft

You are drafting a PRD from existing evidence only. Use the customer feedback, analytics notes and technical constraints below. Do not invent requirements. Mark assumptions clearly. If something important is missing, create an OPEN QUESTION instead of guessing.

Use this prompt for the engineering review

Read this PRD as an engineer who has never seen the feature before. Point out anything that can be understood in two different ways or cannot be implemented without another decision.

What you get

An evidence-based PRD draft and the questions an engineer still needs answered.

Tags are enough for the first version. A model-generated draft is not an approved specification.

[Supabase tables ↗](#) · [Lovable Plan mode ↗](#)

04

A restrained prototype generator

Claude or ChatGPT · Lovable, v0 or Replit

04 / A restrained prototype generator



1 idea, 1 main interaction

The first version

1 idea, 1 main interaction

How to build it

1. Describe the idea and the question you want to test.
2. Generate a brief with screens, actions and fake data. Remove features unrelated to that question.
3. Paste the brief into one builder. Click through it with a user and keep the questions the test exposes.

Use this prompt for the prototype brief

I want to test this idea: [IDEA]
Create the smallest possible clickable prototype that would let me test the main interaction with a user.
Use fake data.
Do not add billing, settings, admin panels, notifications, roles or other secondary features unless they are required for the test.
Give me:
screens needed
what happens on each screen
sample fake data
what can stay completely fake
what question this prototype should help me answer

What you get

A narrow brief followed by a clickable prototype using fake data.

The brief comes first. A prototype is not the full product.

[Lovable Plan mode](#) ↗

05

Your morning product assistant

n8n · Gmail, Slack or Linear/Jira · a model · one delivery channel

05 / Your morning product assistant



1 source, 1 morning digest

The first version

1 source, 1 morning digest

How to build it

1. Run at 8:45 in your timezone. Fetch items newer than the previous successful run.
2. Deduplicate by source ID. Batch the relevant content for one digest and keep original links.
3. Deliver to the channel you check. After a useful week, add the optional end-of-day diary.

What you get

A short digest with blockers, customer issues and decisions waiting for you.

Only advance the saved timestamp after a successful run. You should not lose updates because delivery failed.

[n8n Schedule Trigger](#) ↗ · [n8n Gmail node](#) ↗

Use this prompt for the morning digest

Summarize only things that may affect product decisions today.
Ignore FYI noise and social conversation.
Show:
blockers
customer issues
decisions waiting for me
important changes since yesterday
Keep it short. If nothing important happened, say that.

Use this prompt for the product diary

Based on today's tasks, notes and decisions, tell me what actually changed.
What did we learn?
What assumption got weaker?
What still should NOT be decided yet?

Bonus

Bonus: personal finance sanity checker

Google Sheets or Supabase · n8n · a model

BONUS / Bonus: personal finance sanity checker

Never ask AI to calculate the totals.

The first version

2 or 3 months of exported CSV data

How to build it

1. Import the CSV. Preserve the original amounts, dates, currencies and descriptions.
2. Send merchant descriptions for classification. Review uncertain categories and repeated merchants.
3. Use formulas or SQL for totals. Compare like-for-like months; keep currencies separate unless you have explicit exchange rates.

Use this prompt for transaction classification

Classify this transaction based on merchant name and description.

Return:

normalized_merchant

category

recurring: true/false/unknown

confidence

Do not estimate the amount, date or currency. Those come from the original transaction data.

What you get

Exact totals with cleaner merchant names and reviewable spending categories.

A temporary hold is not automatically a final expense.

Recurrence needs transaction history, not one merchant name.

[Supabase tables](#) ↗

Your first week

Build the small version first.

Choose one build and check whether the result is useful before adding more sources or features.

01 Your own customer feedback brain

30 old support messages

02 A little competitor stalker

5 competitors, 2 or 3 pages each

03 PRD helper with actual context

1 feature, 10 comments, 1 metric note, 1 constraint

04 A restrained prototype generator

1 idea, 1 main interaction

05 Your morning product assistant

1 source, 1 morning digest

+ Bonus: personal finance sanity checker

2 or 3 months of exported CSV data

Keep the source of truth

Original feedback, previous snapshots, tagged evidence and original transaction fields remain available. Model output adds structure; it does not replace the originals.

[Read the full post ↗](#)