

A PRACTICAL FIELD GUIDE

AI Tips&Tools #1

Test an idea. Build an MVP. Automate the work that keeps stealing your day.



Viktor Zaika · 28 September 2026

01 / Validate before building

02 / Choose a sane MVP stack

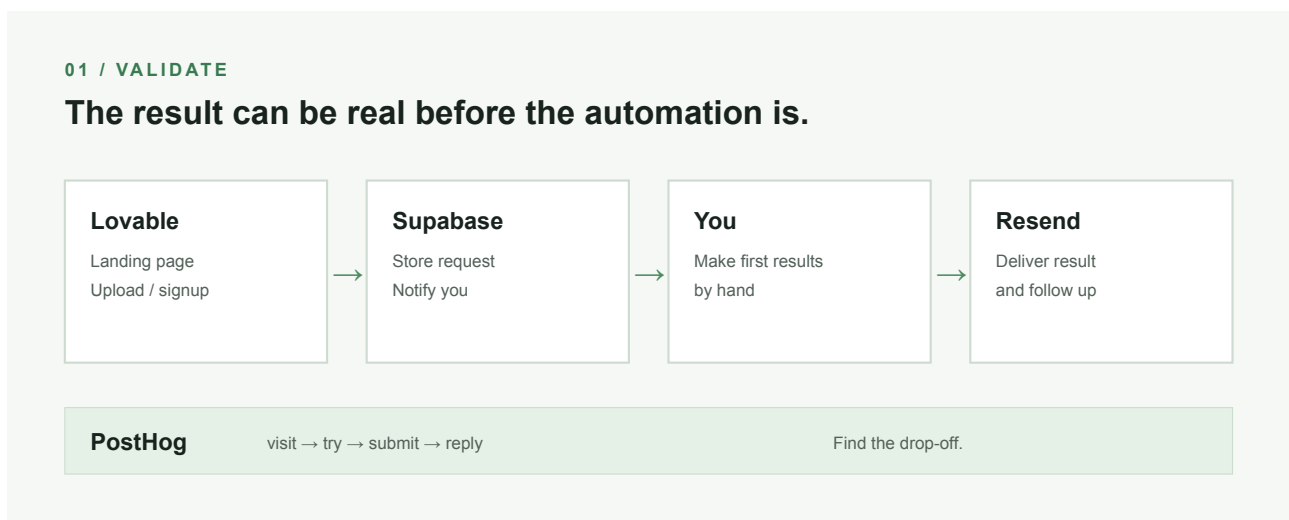
03 / Give automation rules, approval and memory

01 / VALIDATE

Test the result before you build the machine.

The first version can look real to the customer while the expensive part still happens by hand.

Lovable + Supabase + PostHog + Resend



A validation workflow: the page receives a request, you create the first results manually, and Resend delivers them. PostHog measures the steps.

Make the first ten by hand

Imagine a service that analyzes a PDF. Let people upload it. You do the analysis manually and email the report back. That tests demand before you spend weeks automating the analysis.

Architecture example, not a promise that these products connect automatically.

Count actions, not visits

382clicked Try
it**117**started
form**83**finished
form**19**

replied

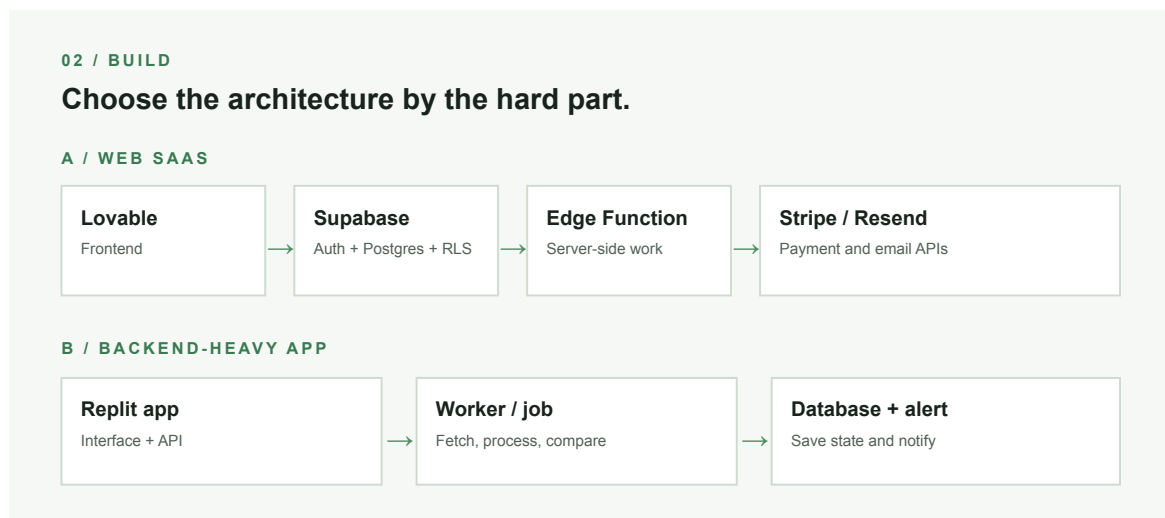
Illustrative numbers. In your own project, use actual event counts.

Sources: [Lovable + Supabase ↗](#) · [PostHog funnels ↗](#) · [Resend API ↗](#)

02 / BUILD

Choose the stack by what is difficult.

A normal web SaaS and a backend-heavy product need different starting points.



Two possible MVP paths. Use the web-app path for UI, login and data; use a backend app when workers, file processing or unusual APIs are the product.

Web SaaS

Lovable + Supabase

Add Stripe, Resend and PostHog when you need payments, email and behavioral data. Supabase Auth identifies the user; row-level security controls which rows they may access.

Backend-heavy work

Replit

Workers, queues, scraping, Python packages and long-running jobs fit more naturally in a backend app than a generated frontend.

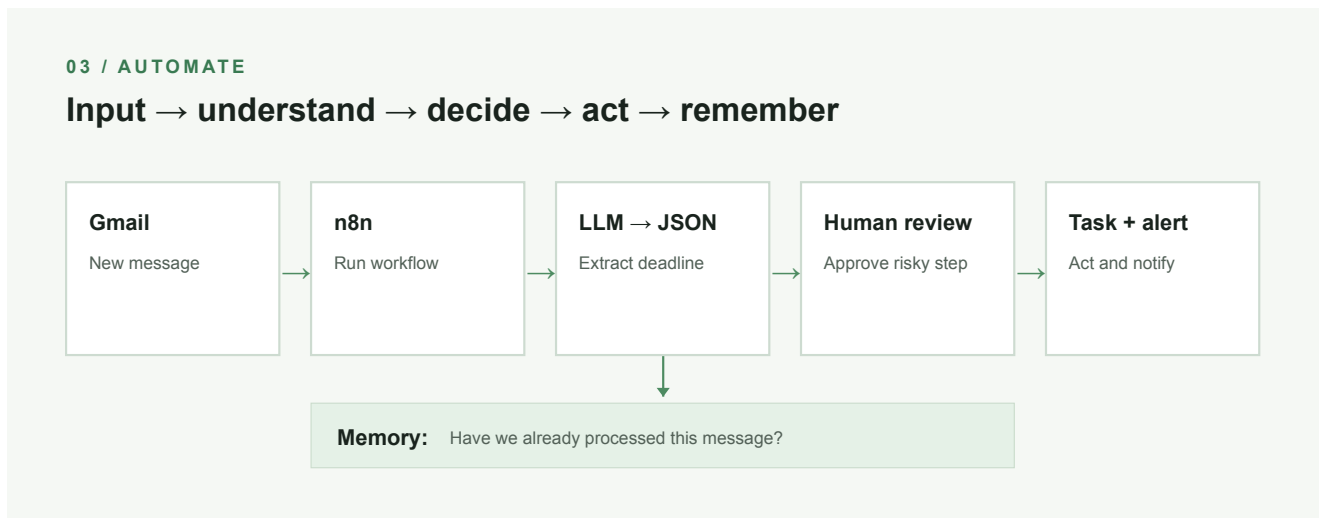
Keep secrets on the server. An OpenAI key, Stripe secret or Resend key visible in browser code is exposed. Put external API calls and webhooks behind a server endpoint or function.

Sources: [Supabase RLS](#) · [Supabase Edge Functions](#) · [Replit Agent](#)

03 / AUTOMATE

Give the workflow a brain, rules and memory.

Input → understand → decide → act → remember.



A daily workflow: structured AI output helps a normal program make decisions; a human confirms actions that are costly to undo.

AI produces data

Ask the LLM for fields such as category, urgency, deadline and confidence. A later workflow step can route those fields predictably.

Use AI where the input is fuzzy: language, messy documents, classification and summaries.

Code handles exact rules

Use ordinary logic for prices, IDs, access control and exact comparisons. Add human approval before sending, publishing or deleting when a wrong decision would hurt.

Remember what happened

A table with **source_url**, **hash**, **first_seen**, **summary**, **topic** can stop a research pipeline from sending you the same story again and again.

Sources: [n8n structured output](#) · [n8n human review](#) · [Supabase Postgres](#)

A FEW USEFUL PIPELINES

Automate the work around your work.

Research

RSS and newsletters → deduplicate → rank relevance → one Telegram digest.

Meetings

Transcript → decisions and actions → tasks and short summary.

Content

Idea in Telegram → Notion → research questions → wait. It does not need to write the post.

Online observability

Alerts, reviews and search results → deduplicate → classify topic → notify when something actually changes.

Support triage

Ticket → JSON with category, urgency and requires_human → route by ordinary rules → draft or escalate.

Reversible by default

“Probably spam” → label for review. A model should not jump straight from a guess to DELETE.

The most valuable automation often removes the annoying organizing work, not the human judgment.

Tools: [n8n](#) · [Make integrations](#) · [Supabase database](#)

KEEP AND ADAPT

Three prompts to start from.

Use this prompt for validating your idea

I want to validate this product idea: [DESCRIBE IDEA].

I do NOT want to build the full product yet.

Design the smallest possible validation architecture using Lovable, Supabase, PostHog and Resend.

Explain:

1. what should actually be built
2. what can be manually faked behind the scenes
3. what data should be stored in Supabase
4. which PostHog events I should track
5. what conversion funnel I should create
6. what emails should be sent through Resend
7. what would count as enough evidence to proceed to an MVP

Keep the architecture intentionally simple. Do not introduce extra services unless there is a very good reason.

Use this prompt for choosing your MVP architecture

I want to build this MVP: [DESCRIBE PRODUCT].

Help me choose between:

A. Lovable + Supabase

B. Replit

C. a combination of both

Do not choose based on which tool you like more. Choose based on the actual technical requirements.

First identify:

1. frontend requirements
2. authentication
3. database entities and relationships
4. authorization rules
5. external APIs
6. background or long running jobs
7. file storage
8. payments
9. transactional email
10. analytics

Then propose the smallest architecture that is still technically sane.

Explicitly tell me which operations must happen server side and which secrets must NEVER be exposed in the frontend.

If using Supabase, propose the tables and RLS rules as well.

KEEP AND ADAPT

Make the workflow dependable.

Use this prompt for automating your own work

I currently do this manually:
[DESCRIBE THE PROCESS IN NORMAL HUMAN LANGUAGE]
Design an automation for it.
Separate the workflow into:
deterministic steps that should use normal logic,
fuzzy steps where an LLM actually makes sense,
actions that should require human approval,
state that needs to be stored so the workflow remembers what it already did.
Prefer n8n and services I already use. Keep the number of moving parts as small as possible.
For every LLM step, define the exact structured JSON output it should return.
Also tell me how this workflow can fail and what I should log so I can debug it later.

The three stacks to remember

Test demand: Lovable + Supabase + PostHog + Resend.

Build an app: Lovable + Supabase, or Replit when backend logic is the interesting part.

Automate your work: n8n + your existing tools + an LLM + a small database for memory.

AI does not mean engineering stopped mattering. Know where the dangerous parts are and where the model should not freestyle.

SOURCES / FURTHER READING

Read the documentation behind the diagrams.

Lovable + Supabase ↗docs.lovable.dev/integrations/supabase**Supabase Edge Functions** ↗supabase.com/docs/guides/functions**PostHog session replay** ↗posthog.com/docs/session-replay**Replit Agent** ↗docs.replit.com/features/agent/overview**n8n human review** ↗docs.n8n.io/advanced-ai/human-in-the-loop-tools**Stripe webhooks** ↗docs.stripe.com/webhooks**Supabase RLS** ↗supabase.com/docs/guides/database/postgres/row-level-security**PostHog funnels** ↗posthog.com/docs/product-analytics/funnels**Resend email API** ↗resend.com/features/email-api**n8n structured output** ↗docs.n8n.io/integrations/builtin/cluster-nodes/sub-nodes/n8n-nodes-langchain.outputparserstructured**Make integrations** ↗www.make.com/en/integrations**Full article and current links** ↗zaika.pro/blog/ai-tips-tools-1-october-2026/

Diagrams are illustrative workflows designed by Viktor Zaika. Product names and logos belong to their respective owners.