

Shahzaib Hassan

AI AUTOMATION ENGINEER

I build automation systems, AI agents and the applications around them. Eight of them are below, with the workflows and the screens rather than a description.

shahxeebhassan@gmail.com · +92 302 7701345 · Lahore, Pakistan
github.com/Shahzaib-Hasaan · linkedin.com/in/shahzaib-hassan-ai-developer

CONTENTS

- 01 **Strives.ai** Autonomous social media system
- 02 **Sentience** Automated cold outreach
- 03 **Mathani** Multi-agent system for a prayer and Quran app
- 04 **Telegram Order Agent** Open source, MIT
- 05 **ANESI** Lead-intelligence platform
- 06 **Slide Generator** AI teaching assistant
- 07 **Toolbelt** Browser-only utilities
- 08 **YTDM** Open source, MIT

01 Strives.ai

Autonomous social media system

365 posts across seven channels, from reading the internet to publishing, with nobody writing them.

Make.com · Airtable · OpenAI · Claude · Apify

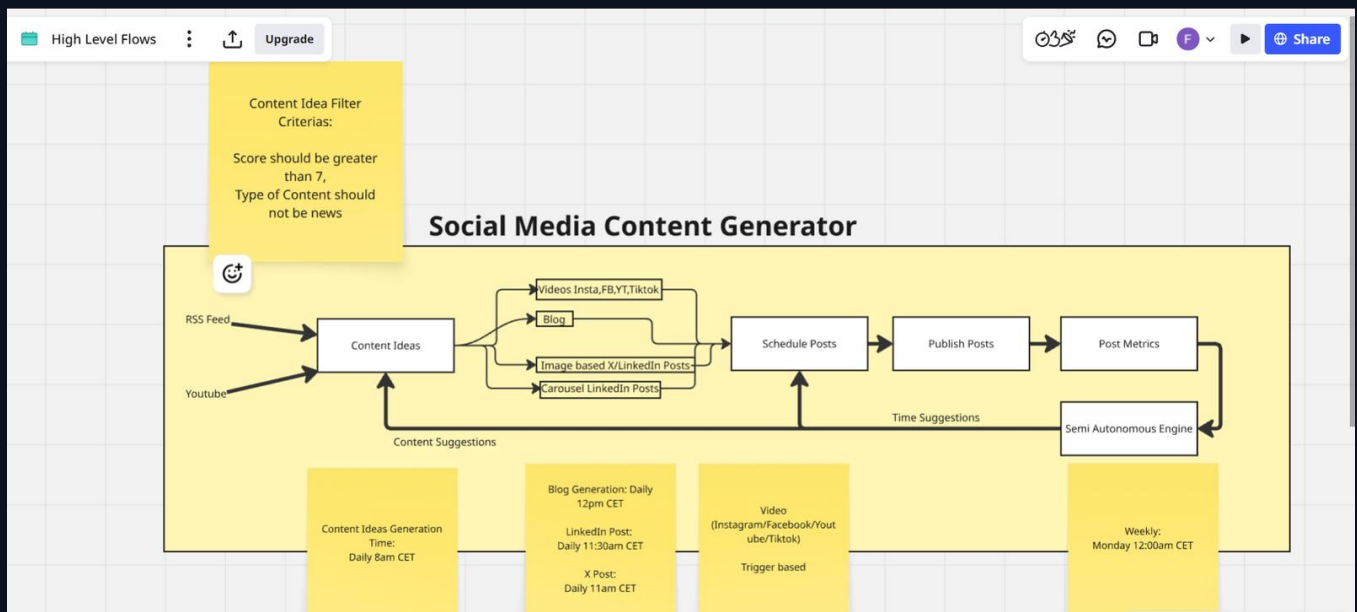
A content engine for a B2B marketing company. It reads sources, decides what is worth writing about, writes it, schedules it, publishes it to seven channels, then measures what happened and feeds that back into what it writes next. 365 posts on the calendar at the time of these screenshots.

It starts with 85 sources, 45 of them RSS feeds, each carrying a mean quality score so a source that keeps producing weak ideas gets deactivated. Three extraction branches run in parallel: RSS articles get downloaded and stripped to text, YouTube videos get transcribed, and LinkedIn profiles get scraped through Apify. Everything lands in Airtable as a scored content idea.

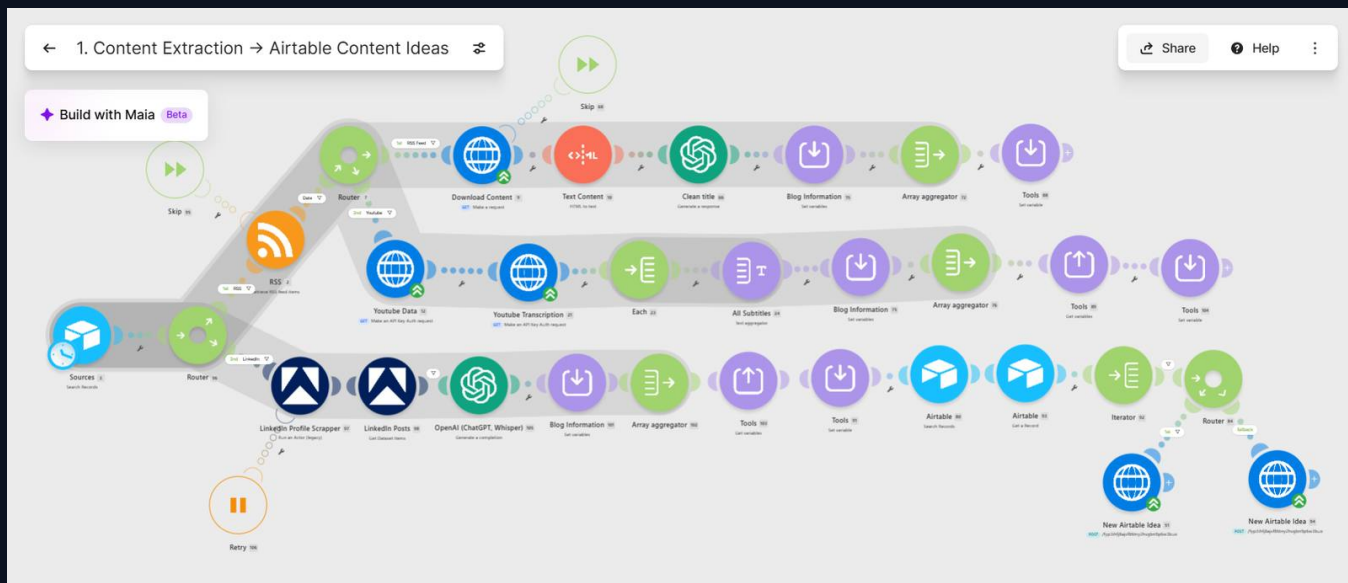
Ideas only move forward if they pass a filter, a score above seven and not classified as news. Generation then runs per channel, and the models are assigned per step rather than globally. A cheap model summarises, a stronger one writes titles and posts, Perplexity handles research on related topics, and images are generated separately. Every prompt lives in a prompt manager with its own model attached, so changing the model behind one step does not mean touching a scenario.

Publishing is a single webhook-triggered scenario that routes to the right channel: Instagram, YouTube, TikTok, Facebook, X, LinkedIn and a blog. Metrics come back from Apify on a schedule, with a retry path for when a platform is slow, and the engine uses them to suggest both what to write about and when to post it.

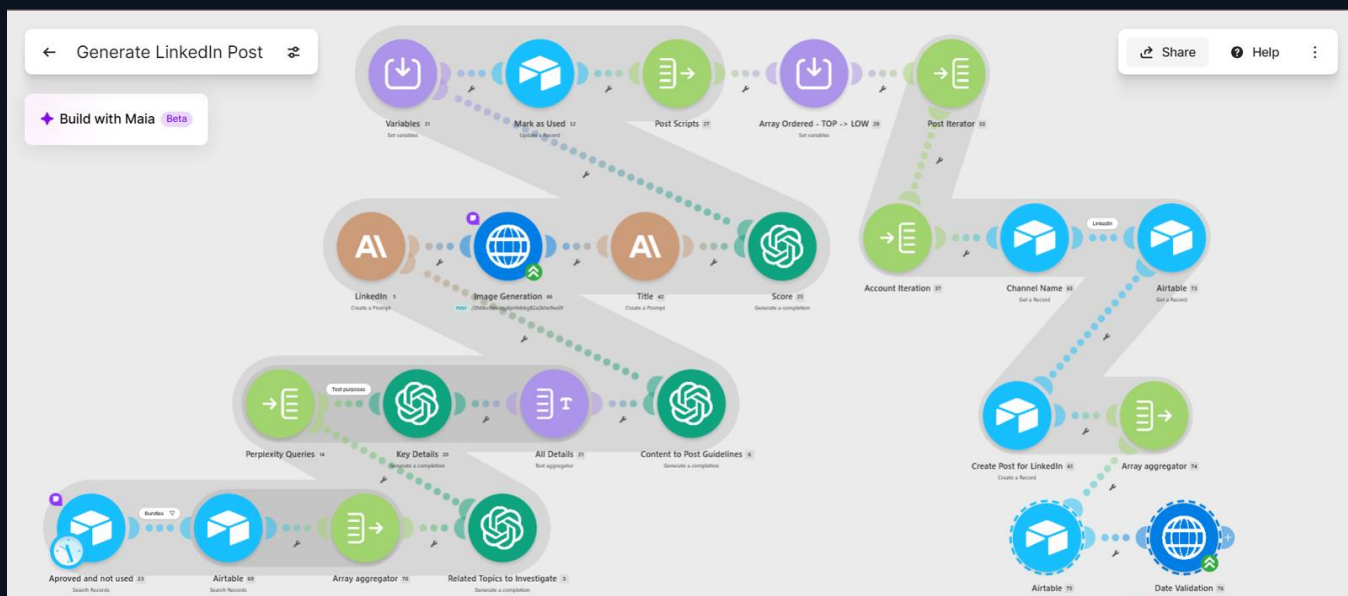
A review board sits in the middle of all this. Ideas queue as pending and a person can approve or reject them, which is the one place a human can step in. At the time of these screenshots there were 97 waiting.



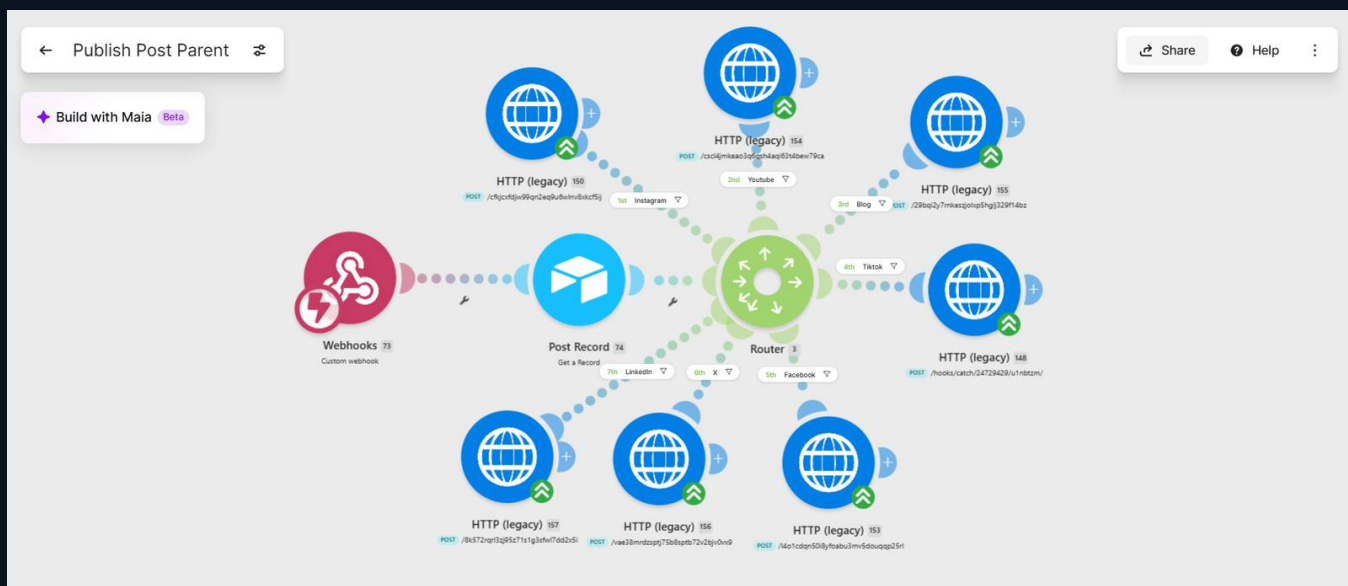
The whole loop, including the feedback path. Metrics from published posts return as content and timing suggestions.



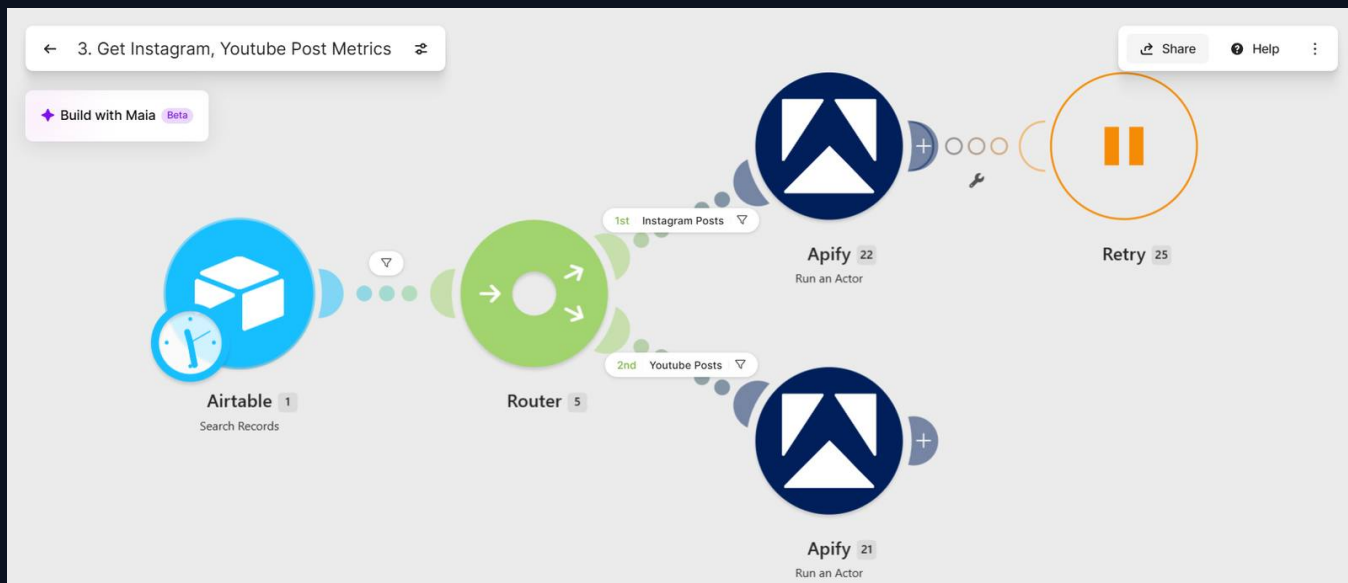
Extraction. Three branches in one scenario: RSS articles, YouTube transcription, and LinkedIn scraping, all normalised into scored ideas.



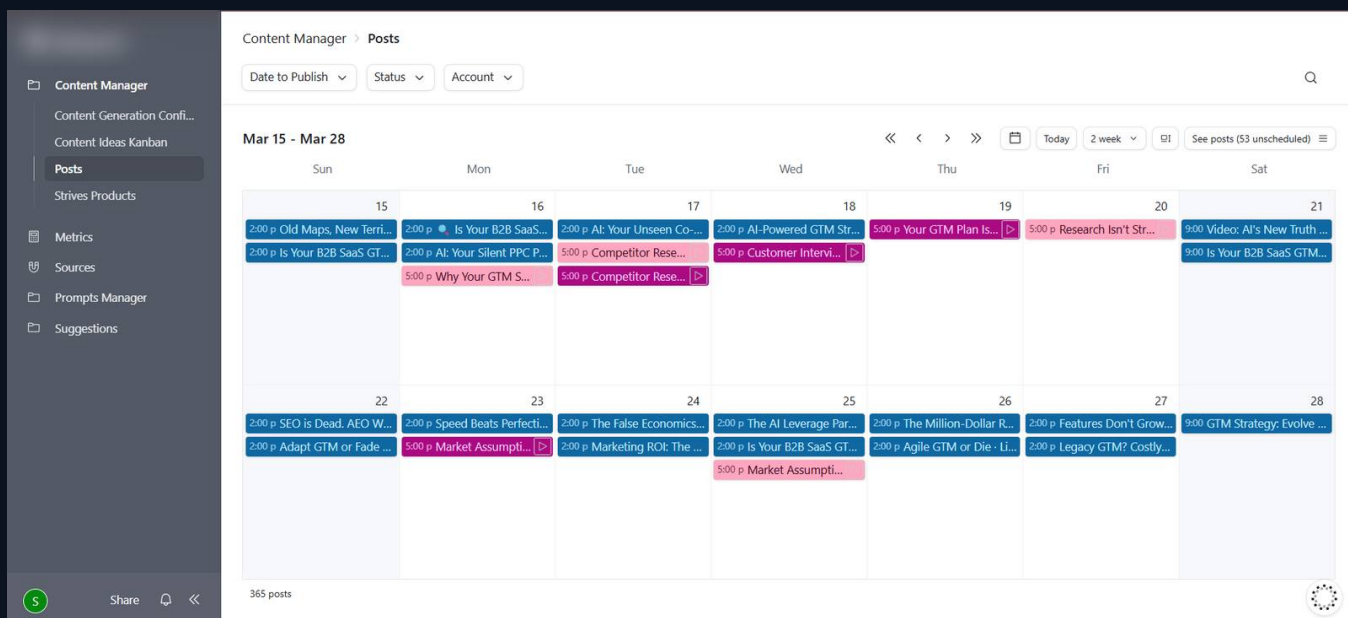
Generation for one channel. Research, key details, title, scoring and image generation, with a different model behind each step.



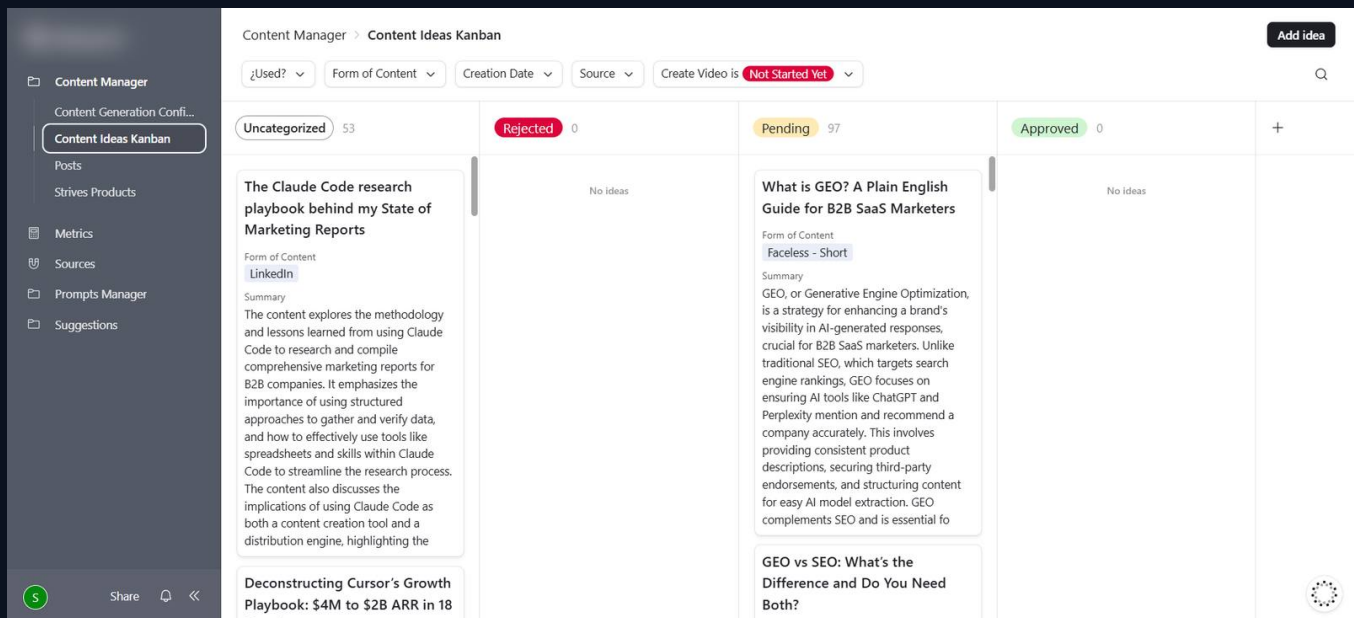
Publishing. One webhook, one router, seven channels.



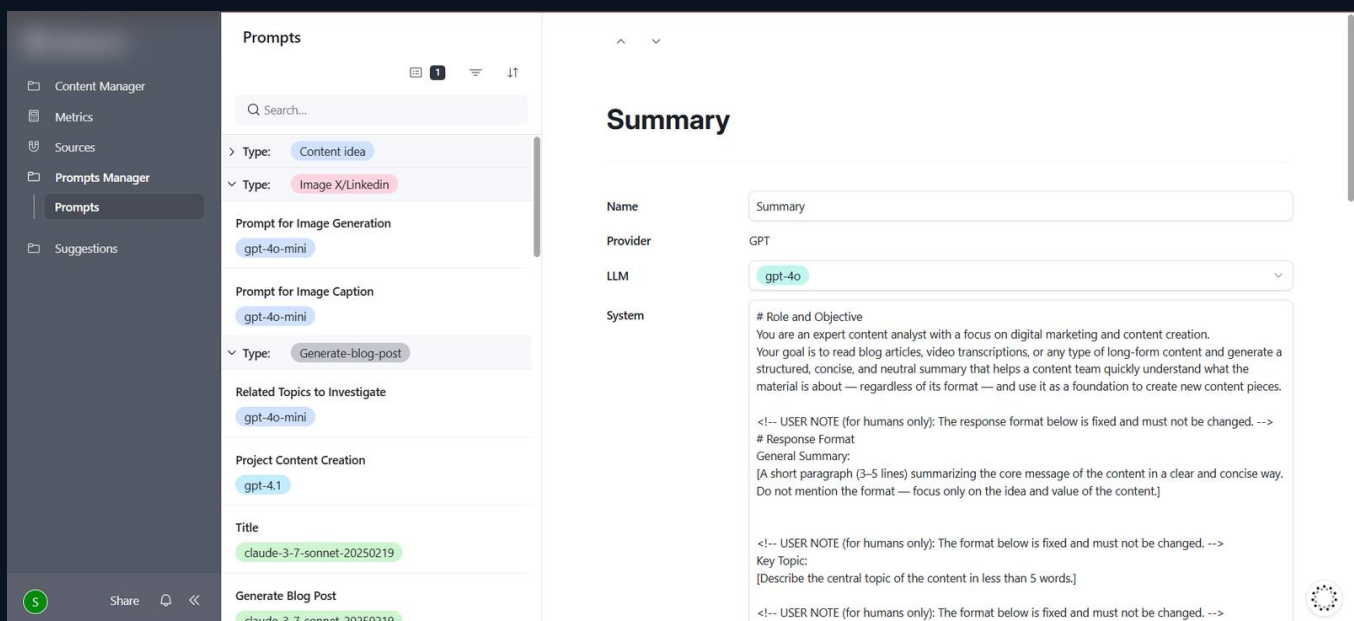
Metrics collection on a schedule, with a retry path for when a platform does not answer.



The publishing calendar. 365 posts scheduled, colour coded by channel and status.



The review board. Generated ideas queue as pending for a person to approve or reject.



Prompt manager. Each step has its own prompt and its own model, editable without opening a scenario.

Content Manager

Metrics

Sources

Sources

Prompts Manager

Suggestions

Sources > Sources

Type

RSS Feed 45

Resources	Status	URL	Primary Content Type	Company	Source Mean Score
Hubspot - RSS Feed	Active	https://blog.hubspot...	Expert Blog	Hubspot	8
Forentpreneurs - RSS Feed	Active	https://www.forentre...	Professional Community	Forentpreneurs	NaN
First Round - RSS Feed	Active	https://firstround.firs...	Expert Blog	First Round	NaN
TechCrunch - RSS Feed	Deactivated	https://techcrunch.co...	Tech News Site	TechCrunch	8
Sales Hacker - RSS Feed	Active	https://gtmnow.com/...	Expert Blog	Sales Hacker	8
Strategy+Business (PwC) - RSS Feed	Deactivated	https://www.strategy...	Consulting Insights	Strategy+Business (PwC)	NaN
Harvard Business Review - RSS Feed	Deactivated	http://feeds.harvardb...	Business Journal	Harvard Business Review	NaN

85 resources

Sources, each with a mean score. A feed that keeps producing weak ideas gets deactivated.

02 Sentience

Automated cold outreach

A CSV of leads becomes a running email campaign, with the AI stepping back the moment someone replies with interest.

Make.com · Airtable · Instantly · OpenAI

An outbound system for a B2B behavioural intelligence company. Their team was running outreach by hand: building lists, researching each prospect, writing each email, chasing replies in an inbox. It capped out at a handful of prospects a day, and personalisation got worse the more they pushed.

Now they upload a CSV, pick the email type and the angle, and the system takes it from there. Leads get enriched, then split: anyone with enough profile data to work with gets a personalised email, anyone without gets a solid generic one. Both go out through Instantly, and every status change writes back to Airtable so the operator dashboard always shows the real state of a campaign.

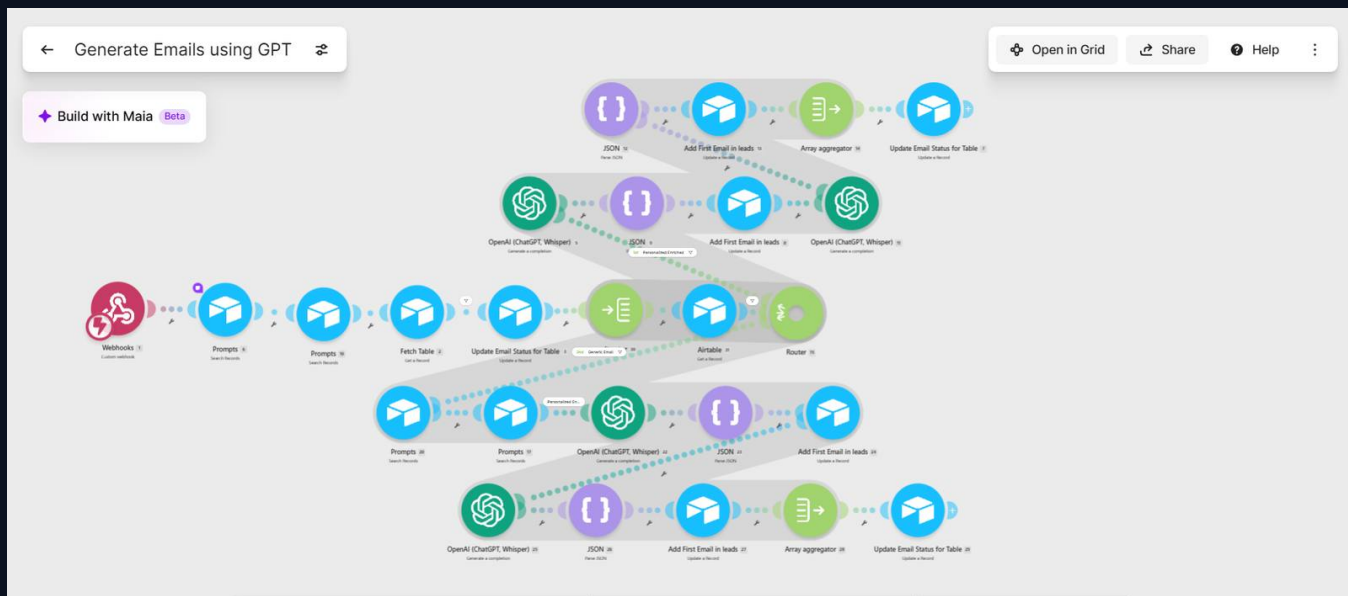
The prompts live in a library rather than inside the scenarios, each one with its own model attached. Personalised first email, personalised follow-up, generic first email, generic follow-up, reply generation. Changing how the copy sounds means editing a record, not opening Make.

The part I would point at is what happens when someone replies. The system reads the reply and drafts a response, but if the lead is actually interested it adds them to a stop table and the AI stops replying to that thread. A person picks it up from there. Anything past initial interest is a conversation about money, and that is not somewhere I wanted a language model making the running.

Campaigns can also be driven one lead at a time from the dashboard, because the useful thing in practice was not bulk sending. It was being able to intervene on a single prospect without breaking the run.



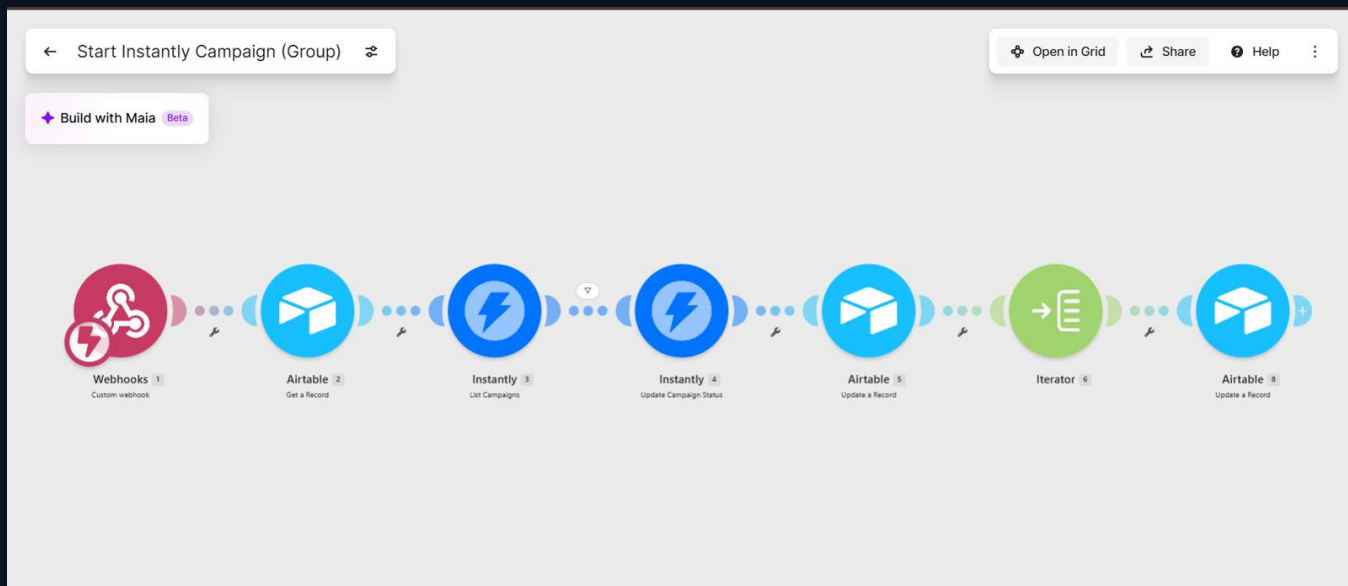
Reply handling. The AI drafts a response, but an interested lead goes into a stop table and the automation steps out of that thread.



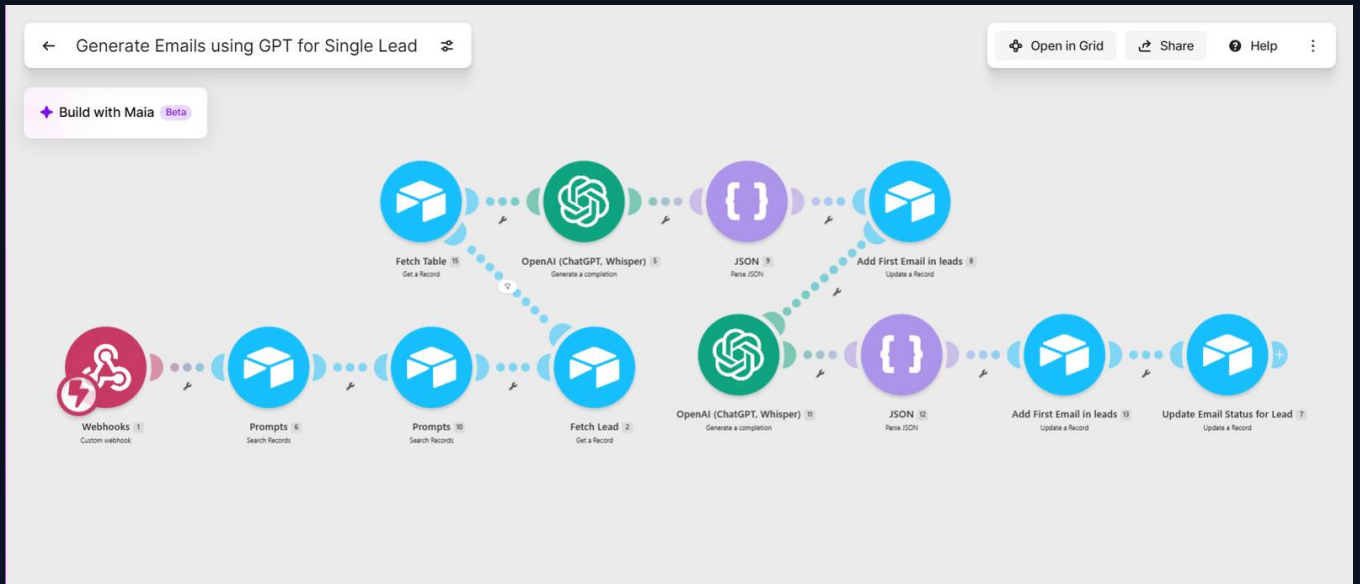
Email generation. Two branches, personalised for enriched leads and generic for the rest, with prompts fetched per run rather than hardcoded.



Campaign creation against the Instantly API, with the lead set built and the campaign record written back to Airtable.



Launch. Webhook triggered, campaign status updated at both ends so the dashboard never disagrees with the sending platform.



The same generation path for a single lead, so one prospect can be handled without touching the rest of the campaign.

Lead Enriching

Import New Leads

Campaign Management

Dashboard

Content Library

Import New Leads

When you submit this form, the system will parse the provided CSV with leads and launch a campaign for those leads.
You can provide one CSV file per request

Leads Table Name *

Enter the table name you want to use for these leads

Email Type

Select the type of Email for this Campaign

GIG

Wellness

Insurance

Angle

Choose the Email Style for this Campaign

Industry

Specify Industry Set for this Campaign

Share

Clear form

Create

The operator side. Upload a CSV, choose email type and angle, and the campaign builds itself.

Lead Enriching

Campaign Management

Email and Campaigns M...

Dashboard

Content Library

Email & Campaign Management

Select Table to view Email Generation ...

2026_01_28_gig_food_delivery.q...

Import Status

Success

Enrichment Status

Completed

Email Status

Generated

Campaign Status

Started

Clicking the Generate AI Email Button will generate personalized emails for all leads. Similarly clicking the Start Campaign Button will start the campaign for all leads.

Generate AI Email

Start Campaign

To generate an email/start campaign for a single lead, update that lead's action below. This will automatically trigger the relevant process for that lead.

	Lead Name	Action	Email Type	Lead Email	Linkedin Url	Summary
LEAD STATUS	Campaign Started	34				
1	Julia Zhu	Start Campaign	Personalized	julia@slice.com		
2	Peter Tilton	Start Campaign	Personalized	peter.tilton@instacart.com		
3	Brian Gorman	Start Campaign	Generic (Not Enriched)	brian.gorman@shipt.com		
4	Patrick Roff	Start Campaign	Generic (Not Enriched)	patrick.roff@instacart.com		
5	Brian Donato	Start Campaign	Generic (Not Enriched)	bdonato@ontrac.com		
6	David O'leary, Cpa	Start Campaign	Generic (Not Enriched)	doleary@ups.com		
7	Kristina Pham	Start Campaign	Generic (Not Enriched)	kristina@shipt.com		

Campaign management. Import, enrichment, email and campaign status in one row, with per-lead actions underneath.

Lead Enriching

Campaign Management

Email and Campaigns M...

Dashboard

Content Library

Campaign Management

View Leads, Generate AI Emails and Start Campaigns using this Interface

View Uploaded Leads

Select Table

2026_01_28_gig_food_delivery.quick_commerce_last Mile_US

Import Status

Success

Newly Uploaded Tables will take some time to show up as the leads are being enriched. Please wait for some time and come again later !

Selected Table Leads

Email Type

Personalized 6

Leads	Lead Email	Linkedin Url	Summary	Company	Com
Julia Zhu	julia@slice.com	—	—	Slice	—
Peter Tilton	peter.tilton@instacart.com	—	—	Instacart	—
Veronica Yu	veronica.yu@instacart.com	https://www.linkedin.com/...	Chief of Staff to the Gener...	Instacart	http

Leads after enrichment, grouped by whether there was enough to personalise from.

Lead Enriching

Campaign Management

Dashboard

Content Library

Prompts

5

Share

<<

Prompts

Q Search...

Personalized Email Generation

gpt-5-mini

Personalized Follow Up Emails Generation

gpt-5-mini

Generic Email Generation

gpt-5-mini

Generic Follow Up Email Generation

gpt-5-mini

Reply Generation

gpt-5-mini

^

v

Personalized Email Generation

Name

Personalized Email Generation

Models

gpt-5-mini

Provider

ChatGPT

Prompt

You are a B2B outbound email copywriter working for Sentiance, an AI-native behavioral intelligence company.

LANGUAGE & BRAND GUIDELINES (MANDATORY):

Language: English (US)

Tone must ALWAYS reflect the Sentiance brand:

Passionate: confident and positive about creating real-world impact

Inspirational: concise, forward-looking, and focused on new possibilities

Reliable: trustworthy, ethical, privacy-first, sustainability-aware

Casual: conversational, clear, easy to read, and human

Avoid complex technical terms or jargon

Sound smart, warm, and grounded — never hype-driven or salesy

INPUTS YOU WILL RECEIVE:

Lead Name: {{Lead_Name}}

LinkedIn Profile Text or URL: {{LinkedIn_Profile}}

Company where this lead works: {{Lead_Company}}

Email Type: {{Email_Type}} (one of: GIG, Insurance, Wellness)

Angle: {{Angle}} (one of the 7 angles listed below)

The prompt library. Five prompts, each with its own model, editable without opening a scenario.

03 Mathani

Multi-agent system for a prayer and Quran app

A set of agents running behind a mobile app: reading its reviews, answering questions about them, and handling the work that follows.

n8n · LangChain · pgvector · OpenAI · MCP

An ongoing build for a prayer and Quran app. Several agents, each doing one job, sharing a Postgres database.

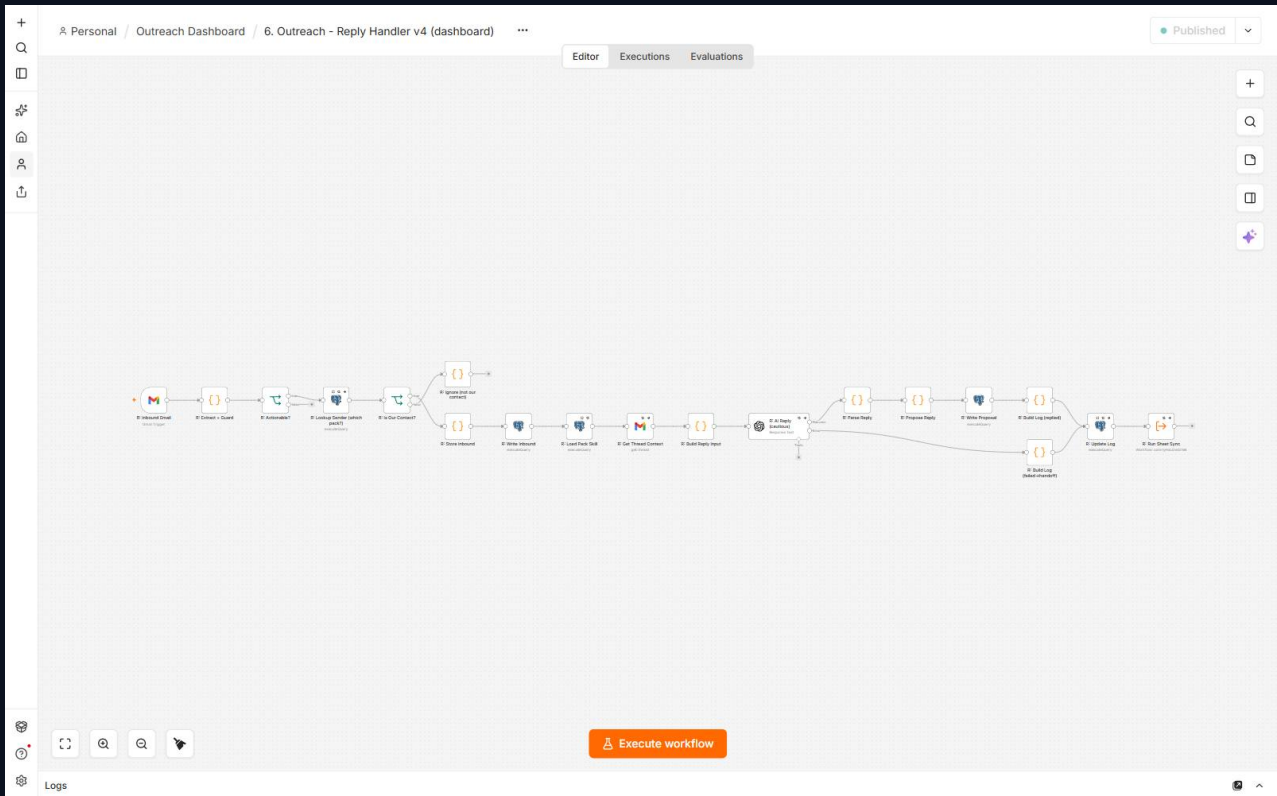
The first one reads the app store. Every day it pulls new Google Play reviews, skips anything it has already seen, classifies each one with an LLM, and stores it alongside an embedding. A second agent sits on top of that with a chat interface, so someone can ask what people are complaining about this week and get an answer drawn from the actual reviews rather than a summary someone wrote by hand. It has both a vector store for finding relevant reviews and a direct SQL tool for counting and filtering, because some questions are better answered by a query than by similarity search.

Others handle the work that comes out of that. A ticketing agent creates and routes Linear issues from a chat message, using an MCP tool and the GraphQL API, and assigns them by team. It does not create anything on its own budget: it calls the budget agent first. A budget agent prices every proposed ticket against the monthly budget before it can be created, with release, hold and defer routes and an escalation path when something does not fit. A responder agent replies to comments on Linear issues, checks whether the comment is relevant first, and escalates instead of guessing.

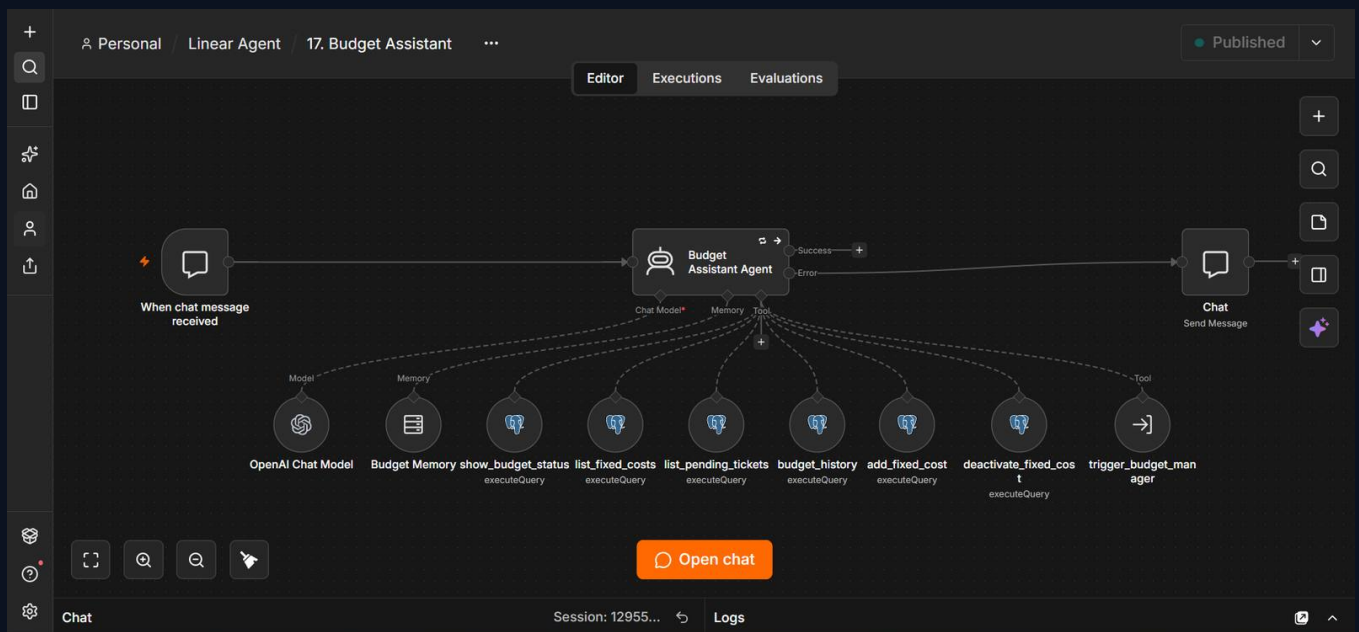
An email agent reads the mailbox, works out whether a reply needs action, pulls the thread for context and drafts a cautious response. It never sends. The draft goes to a person, and a separate workflow does the sending, triggered by a button press.



Daily review ingestion. Fetch from Google Play, dedupe against what is already stored, classify with an LLM, then insert the review and its embedding.



Inbound email. Guard, actionable check, contact lookup, thread context, then a reply proposed for a person to approve. Nothing sends from here.



Budget agent. Every tool on it is a database operation, so the model reports on the budget rather than deciding what the budget is.

04 Telegram Order Agent

Open source, MIT

Customers order a takeaway in Roman Urdu. The AI reads the order. The database decides what it costs.

n8n · Supabase · Next.js · Telegram

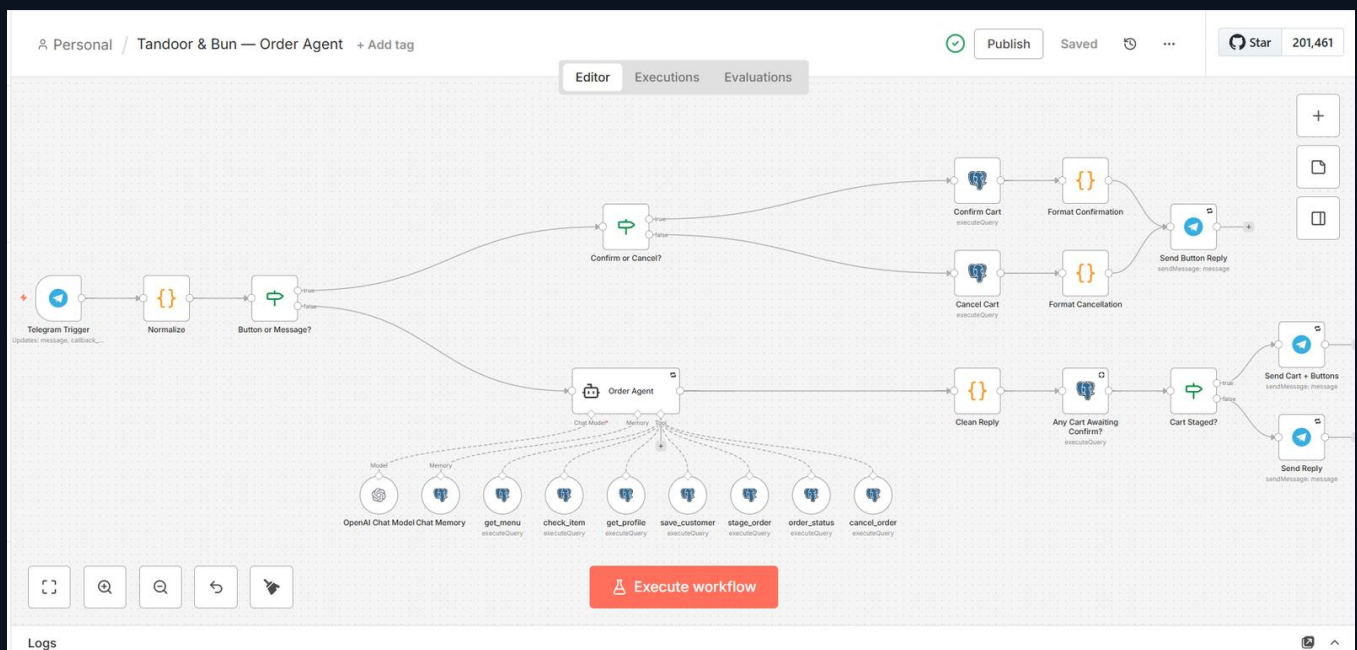
A restaurant ordering system that runs entirely through a Telegram chat. Customers type in English, Roman Urdu or a mix of both. The agent works out what they want, checks it against the real menu, collects a delivery address and phone number, and shows a priced cart to confirm. The order lands in Postgres and appears on a kitchen board, where one tap moves it along and messages the customer automatically.

The first order it ever took, it invented the customer. I was testing it and sent only my address. It needed a name and a phone number before it could place the order, so it took both from the street: the name was the road, the phone number was the house number. No error, no crash, confident and completely wrong. I had already written "always ask for a valid phone number" into the prompt. It happened anyway.

So every rule that matters moved out of the prompt and into the database. The agent has seven things it is allowed to do and it cannot write its own SQL. It sends item names and quantities; the database looks up the prices itself and ignores anything the model suggests. Phone numbers have to match the shape of a Pakistani mobile or they are refused at the door. Minimum order, delivery fee and sold-out items are all enforced the same way.

Confirming works the same way. Typing yes or haan does nothing. Only the button places an order, and the button carries the id of a cart that was already priced and stored, so what the customer confirms is exactly what they were shown. Building a new cart supersedes the old one, which means a stale button cannot be tapped into a surprise.

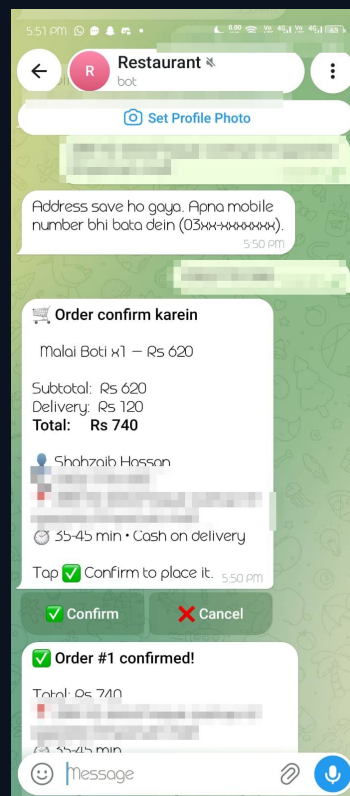
The demo restaurant is fictional and everything about it is configuration, so swapping the menu and the delivery rules runs it for a real shop. The whole thing is MIT licensed with a setup guide.



The n8n workflow. The seven Postgres functions hanging off the agent are everything it can do. There is no eighth, and it cannot write its own SQL.



Ordering in Roman Urdu. Spelling varies, nicknames work, and the agent resolves them against the real menu.



The cart, priced by the database. Only the Confirm button places the order.

Tandoor Pass

6 open

OLDEST

2h 10m

#13

6 past promise

TODAY 9

TAKEN Rs 12,220

Compact

History

Menu

NEW

1

COOKING

3

OUT

2

#10

1h 44m

2x Chicken Biryani 840

1x Raita 80

2x Garlic Naan 180

extra spicy

Rs 1,220

5 ITEMS

Start cooking

Cancel

#12

1h 57m

1x Chicken Karahi (Half) 1,450

4x Garlic Naan 360

Rs 1,930

5 ITEMS

Hand to rider

Cancel

#11

1h 32m

1x BBQ Platter 1,850

2x Sweet Lassi 440

Rs 2,290

3 ITEMS

Hand to rider

Cancel

#9

1h 21m

#14

1h 49m

2x Seekh Kabab (4 pc) 1,120

3x Roghni Naan 330

Rs 1,570

5 ITEMS

Mark delivered

Cancel

#13

2h 10m

1x Mighty Zinger 720

1x Loaded Fries 480

2x Kheer 360

Rs 1,680

4 ITEMS

Mark delivered

Cancel

Rail = time against the 45 min promise

on time

halfway

late

past promise

Board shows open orders only - finished go to History

The kitchen board. Each ticket has a rail that fills as the order ages against a 45 minute promise, green through amber to red, so a glance across the room shows what is late.

← Board

Menu

26 items

Search name, category or what the bot hears

Add item

BURGERS 4

Zinger Burger

zinger · zingr · zingar · zinger burgr · singer burger · zinger brgr

Rs 550

Sell out

Remove

Mighty Zinger

mighty · mighty zinger · bara zinger · double zinger · mighty zingr

Rs 720

Sell out

Remove

Beef Tandoori Burger

beef burger · tandoori burger · beef tandoori · tandori burger

Rs 680

Sell out

Remove

Chicken Cheese Burger

cheese burger · chicken burger · cheez burger · chicken cheese

Rs 480

Sell out

Remove

DESI 5

Chicken Karahi (Half)

karahi · chicken karahi · adha karahi · half karahi · karhai · kadhai

Rs 1,450

Sell out

Remove

Chicken Karahi (Full)

full karahi · poori karahi · bari karahi · full chicken karahi

Rs 2,700

Sell out

Remove

Chicken Biryani

biryani · chicken biryani · biriyani · briyani · chicken briyani

Rs 420

Sell out

Remove

Beef Biryani

beef biryani · beef briyani · beef biriyani

Rs 480

Sell out

Remove

Daal Makhani

Menu management, including the aliases the agent listens for.

View repo: <https://github.com/Shahzaib-Hasaan/telegram-order-agent>

05 ANESI

Lead-intelligence platform

Describe the kind of company you want. Get back a list with named decision makers and their contact details.

Next.js · Postgres · HubSpot

A lead research tool for a UK advisory firm. You describe the kind of company you are looking for and filter on headquarters, size, growth, ownership and funding. What comes back is a list of matching companies with financials, the decision makers at each one, and their contact details.

Six providers run in sequence. Inven for company search, Companies House for the registration number, Pomanda for filed financials, LinkedIn for the decision makers, Kaspr for their contact details, and Tavily for recent company activity. Finished records sync into HubSpot, deduplicated by domain for companies and by email for contacts.

Pomanda is paid and needs a company registration number before it returns anything, so the system gets that number from Companies House first and only then makes the paid call. If no number can be matched, it falls back to funding data from Inven rather than dropping the company.

The screenshot displays the ANESI lead intelligence platform interface. On the left is a dark sidebar with a list of filters: Business focus, Keywords, Headquarters, Example companies, Size, Growth, Ownership, Funding and M&A, Founded, and Number of Results. Each filter has a plus icon to its right. At the bottom of the sidebar is a circular button with the letter 'N' and a left-pointing arrow.

The main content area is titled 'History 1' and shows a list of three companies:

☐	Company Logo & Name	Location	Employees	Funding	Status	Founded	Website	Actions
☐	convedo.com Convedo is a consulting firm specializing in...	London, England, GB	9	\$ \$1.3942202435743498M	Private	2009	N/A	
☐	Prometheons prometheons.com Prometheons is a technology company...	Manchester, England, GB	1	\$ \$0.337620380534811M	Private	2024	N/A	
☐	Fliwheel.tech fliwheel.tech Fliwheel.tech is a consulting firm...	London, England, GB	3	\$ \$0.40588082626043703M	Private	2022	N/A	

Below the company list is an 'Enrichment pipeline' diagram showing a sequence of six steps connected by arrows:

- Inven** (AI Company Search) - marked with a green checkmark
- Pomanda** (Data Enrichment)
- LinkedIn** (Contact Discovery)
- Kaspr** (Contact Enrichment)
- Tavily** (Recent Activities)
- HubSpot** (CRM Integration)

Below the pipeline, it says 'Select companies to continue enrichment'.

The enrichment pipeline. Six providers in sequence, each stage feeding the next, and a company can drop out of any stage without stopping the run.

History 1

New Search

Business focus

Headquarters

Get leads

ENRICHMENT PHASE:

Inven

Pomanda

LinkedIn

Kaspr

Tavily

HubSpot

	COMPANY	LOCATION	EMPLOYEES	REVENUE	OWNERSHIP	FOUNDED	COMPANY NO.	LINKS
☐	SS&C Blue Prism blueprism.com SS&C Blue Prism is a subsidiary of SS&C...	Warrington, England, GB	1,046	\$ \$823.626583639256M	Corporate	N/A	N/A	🔗 🔍
☐	Automate Agency automateagency.co.uk Automate Agency is an automation agency tha...	Durham, England, GB	1-10	\$ \$0.262896003225416M	Private	2023	N/A	🔗 🔍
☐	Pluto AI plutoai.io Pluto AI is an AI automation agency tha...	London, England, GB	2	\$ \$0.339487023751031M	Corporate	N/A	N/A	🔗 🔍
☐	Creative Edge creativeedge.io Creative Edge is an automation agency tha...	London, England, GB	4	\$ \$0.44094796300527805M	Private	N/A	N/A	🔗 🔍

Search results after the first enrichment pass. The strip above the table tracks which provider each company has been through.

History

Manual

Auto

Get leads

🔍

500+

Search for Companies

Use filters on the left to find companies. Describe the primary function of the company (e.g. product, industry) or add examples of companies or relevant keywords.

Filters on the left build the query. Manual mode steps through each enrichment phase; auto runs the whole chain.

06 Slide Generator

AI teaching assistant

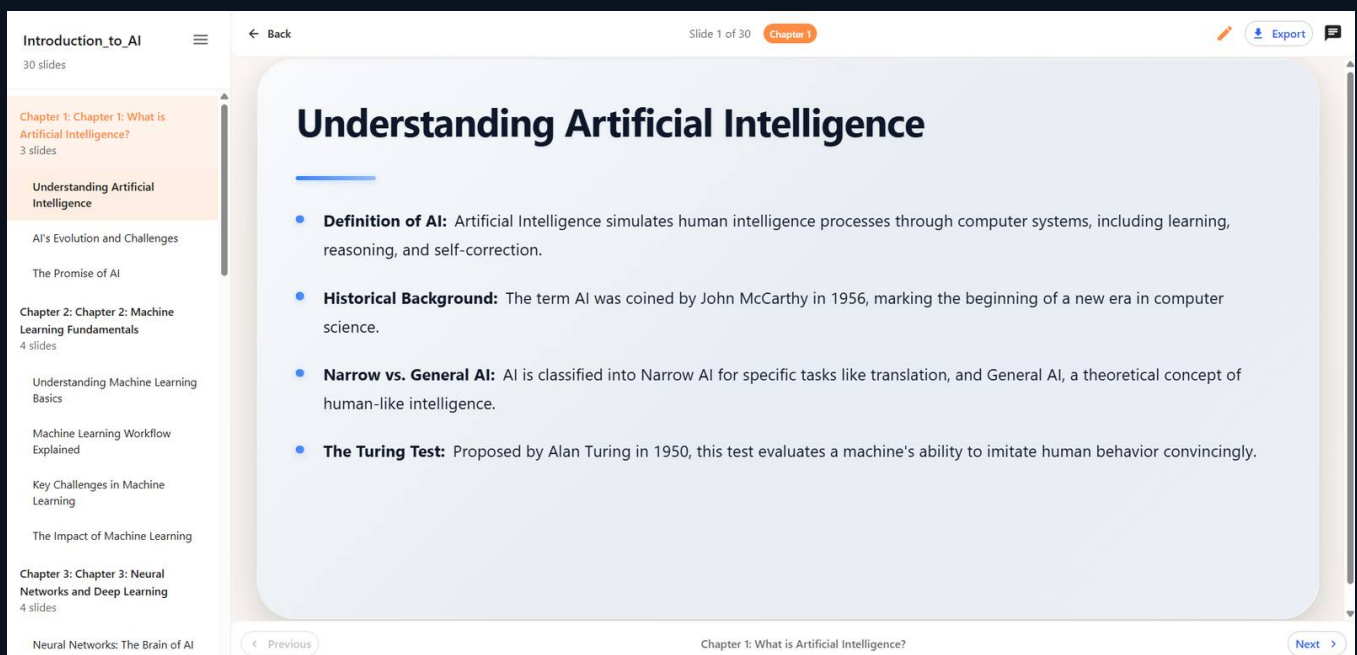
Upload a book, get a slide deck, then walk through it on a voice call with something that answers questions.

Next.js · Postgres · ElevenLabs · OpenAI

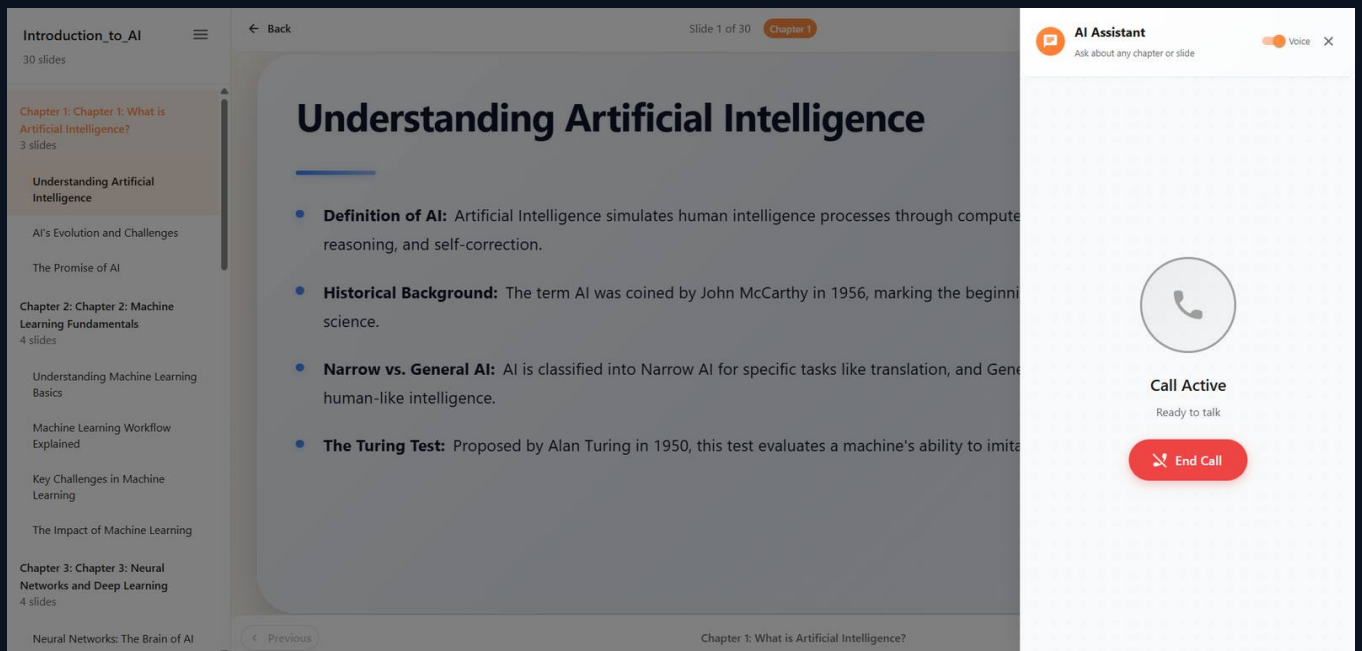
You upload a PDF, EPUB or Word file. It works out where the chapters are, summarises each one, and generates slides from the summary. You choose how detailed you want the summary before it runs, or turn on auto and let the whole chain go by itself. The book in these screenshots came out as 30 slides across its chapters.

The part I cared about most is what happens after the slides exist. An ElevenLabs voice agent sits on top of the document, so you can present the deck and interrupt it with a question on a live call. Every question is answered against the slide you are looking at, plus the summary of the chapter it belongs to. Without that the agent answers from the book as a whole and misses the point, because the same term shows up on several slides and means something slightly different each time.

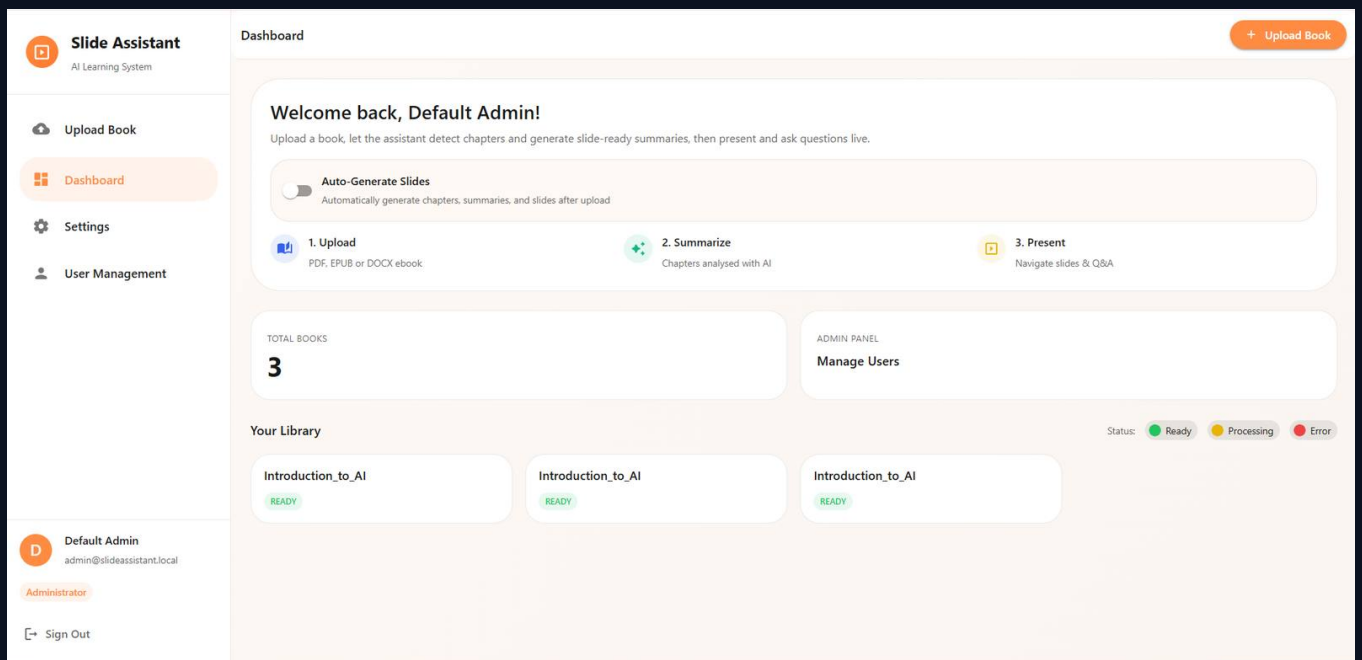
Multi-user, with accounts, roles and an admin area for managing users and their books. Decks export to PDF or PowerPoint. Every question and answer is kept against the slide it was asked on. The prompts for chapter detection, summarising, slide generation and the tutor are stored as records rather than hardcoded, so they can be changed without a deploy.



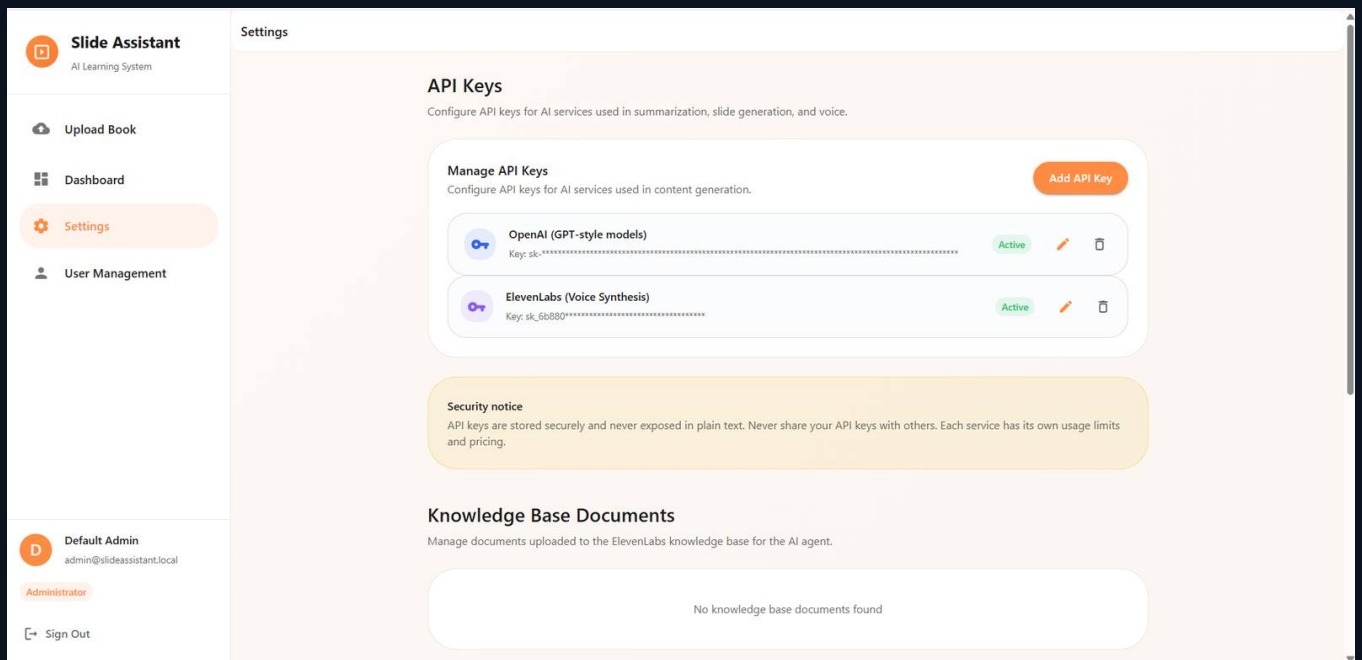
Generated deck. Chapters down the left, detected from the source document, slides built from each chapter summary.



Voice call open on a slide. The agent is answering against this slide and its chapter, not the whole book.



Library and upload. A book moves through uploaded, processing and ready, and shows an error state when parsing fails.



Keys and the ElevenLabs knowledge base are managed in the app rather than sitting in environment files.

07 Toolbelt

Browser-only utilities

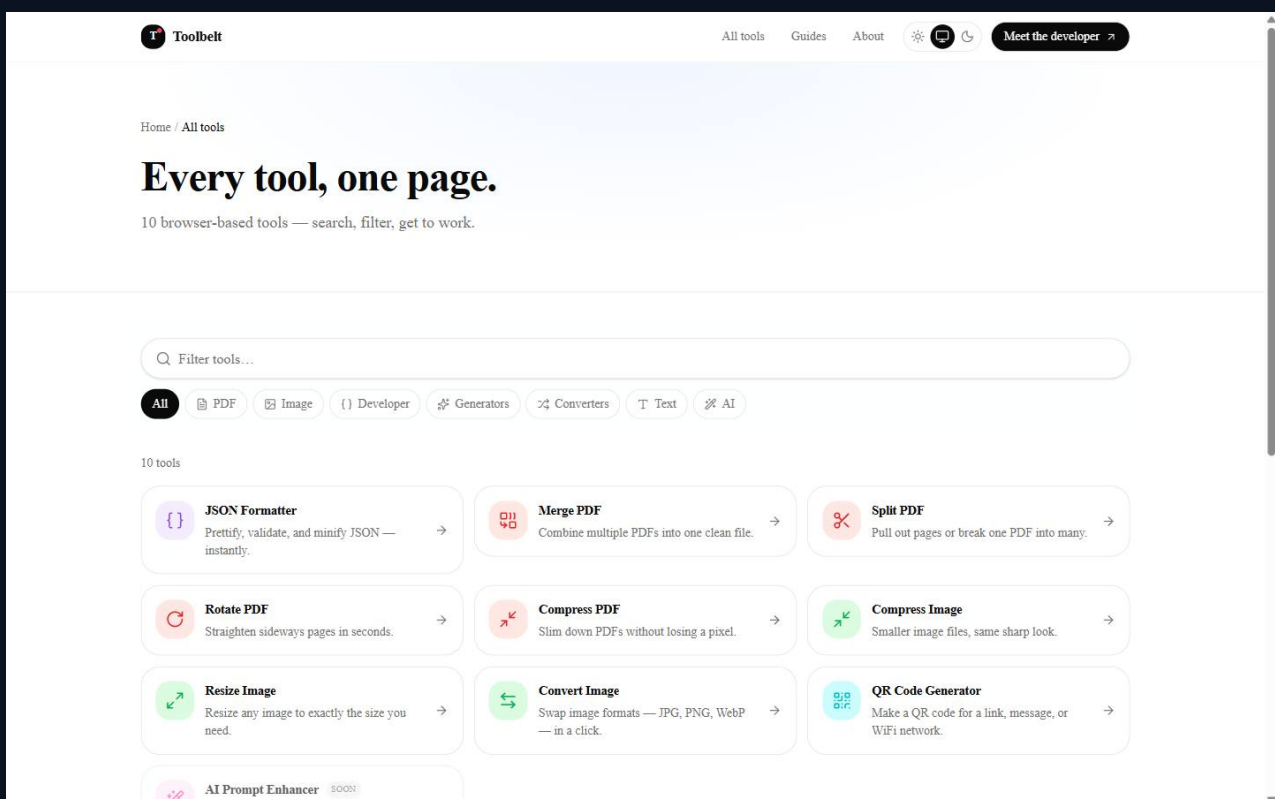
Nine PDF, image and developer tools where the file never leaves your machine.

Next.js · Tailwind · pdf-lib · Canvas API

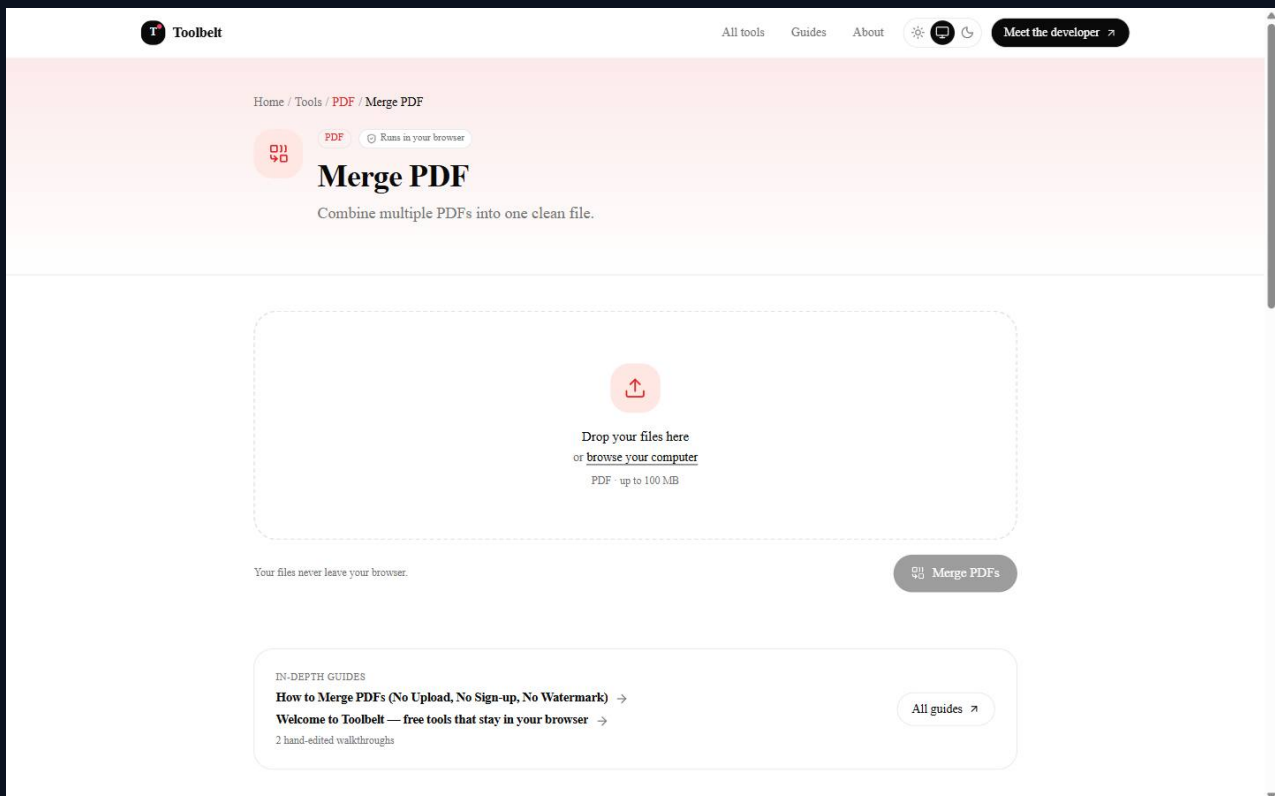
Nine tools for the jobs people normally hand to a random website: merge, split, rotate and compress PDFs, compress, resize and convert images, format JSON, generate a QR code. No upload, no sign-up, no watermark.

Everything runs client-side, which is the whole point. Open DevTools while you use it and the network tab stays empty, so you do not have to take my word for it. That constraint made it harder to build: PDF work runs on pdf-lib in the browser and image encoding runs on the Canvas API, because two WebAssembly libraries I tried first both hung the build in Next 16. The boring built-in shipped and the impressive dependency did not.

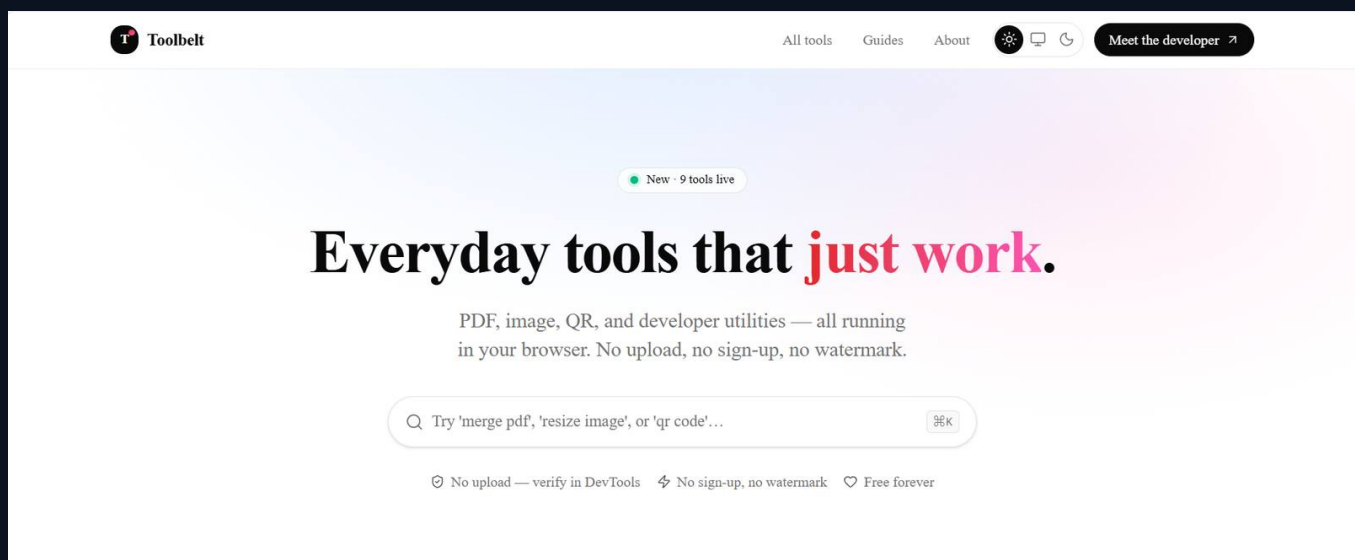
Compress PDF is deliberately honest about what it can do. It is lossless only, strips metadata and thumbnails and repacks the file, typically saving 5 to 15 percent. When a file is already optimised it says so instead of inventing a number, which is more than most online compressors manage.



All nine tools, grouped by category.



A tool page. The badge and the line under the dropzone both say the same thing, and DevTools will confirm it.



Home. Search-first, because people arrive knowing the job they need done.

Open Toolbelt: <https://tools.shahzaibbuilds.me>

08 YTDM

Open source, MIT

A desktop download manager that survives being closed mid-download.

Electron · yt-dlp · FFmpeg · Deno

A cross-platform desktop app for Windows and Linux. Queue up a playlist, pause and resume individual items, pick quality and subtitles, cap the bandwidth so the rest of the house still works.

The part that took the effort was crash recovery. A download interrupted halfway is not restarted from zero, it picks up where it stopped, which matters when the queue is a forty item course and the machine reboots. The engine also updates itself, because yt-dlp changes faster than anyone wants to ship releases for.

Builds for both platforms come out of GitHub Actions. MIT licensed.

The screenshot shows the YTDM desktop application interface. On the left, there's a sidebar with a search icon, a list of filters (All downloads, Downloading, Completed, Failed), and a status bar indicating 'Engine ready'. The main area displays a table of downloads. The table has columns for VIDEO, SIZE, PROGRESS, SPEED, ETA, and STATUS. The first item is '#1 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 40.1 MB and 100% complete. The second item is '#2 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 79.6 MB and 42% complete. The third item is '#3 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 0.0% complete. The fourth item is '#4 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 0.0% complete. The fifth item is '#5 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 0.0% complete. The sixth item is '#6 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 0.0% complete. The seventh item is '#7 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 0.0% complete. The eighth item is '#8 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 0.0% complete. The ninth item is '#9 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 0.0% complete. The tenth item is '#10 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 0.0% complete. The eleventh item is '#11 Machine Learning Specialization [Course 1, ...]' by DeepLearningAI, which is 0.0% complete. The status bar at the bottom shows '2 downloading', '38 queued', and '1 done'. The download path is 'C:\Users\shahz\Downloads' and the speed is '19.3 MB/s'.

VIDEO	SIZE	PROGRESS	SPEED	ETA	STATUS
#1 Machine Learning Specialization [Course 1, ...] DeepLearningAI	40.1 MB	100%	100%	-	Done
#2 Machine Learning Specialization [Course 1, ...] DeepLearningAI	79.6 MB	video 42%	19.3 MB/s	2s	Downloading
#3 Machine Learning Specialization [Course 1, ...] DeepLearningAI	-	0.0%	-	-	Downloading
#4 Machine Learning Specialization [Course 1, ...] DeepLearningAI	-	0.0%	-	-	Queued
#5 Machine Learning Specialization [Course 1, ...] DeepLearningAI	-	0.0%	-	-	Queued
#6 Machine Learning Specialization [Course 1, ...] DeepLearningAI	-	0.0%	-	-	Queued
#7 Machine Learning Specialization [Course 1, ...] DeepLearningAI	-	0.0%	-	-	Queued
#8 Machine Learning Specialization [Course 1, ...] DeepLearningAI	-	0.0%	-	-	Queued
#9 Machine Learning Specialization [Course 1, ...] DeepLearningAI	-	0.0%	-	-	Queued
#10 Machine Learning Specialization [Course 1, ...] DeepLearningAI	-	0.0%	-	-	Queued
#11 Machine Learning Specialization [Course 1, ...] DeepLearningAI	-	0.0%	-	-	Queued

A 41 item queue mid-run. Live speed graph, per-item progress, and the states that actually matter: downloading, queued, done, failed.

View repo: <https://github.com/Shahzaib-Hasaan/ytmd>