



Pre-Seed — Working Product, Pre-Incorporation — October 2026

Describe an app. Ship it to **web**, **desktop** and **mobile**.

NewAgentsHub generates full-stack apps in ten languages from 65 templates, with the backend already wired — real tables and security policies applied to a live Supabase project, MCP servers connected, AI providers configured, and generated skills, plugins, agents and teams built in. Deployed to GitHub and Vercel in one click — or built into Windows, macOS and Linux installers and Android and iPhone apps.

Ilija Dimitrov | Solo builder | newagentshub.dev

No company yet · no users yet · no revenue yet — a working product looking for its first backer

Presented by Matas Ramanauskas — on behalf of NewAgentsHub

 [Book a video call →](#)

Anyone Can Get a Front End. Almost Nobody Gets a Business.

AI app builders generate an interface in minutes, and most now generate a backend to go with it. What none of them hand you is a company: billing that takes money, auth that holds real users, an AI layer that does the work, live integrations, and ownership of all of it. Assembling those is the five weeks between a demo and a product — and the point where a non-engineer stops.

WHAT EVERY BUILDER GIVES YOU

- ✓ Pages & components
- ✓ A styled UI
- ✓ A backend — on their cloud

WHAT A REAL SAAS STILL NEEDS

- ✗ Billing that takes money
- ✗ Auth for real users
- ✗ An AI layer that works
- ✗ Live integrations
- ✗ Infrastructure you own
- ✗ The source code

- ✗

The backend they give you is theirs

The default is their own managed cloud — their billing, their infrastructure, their dashboard. Lovable's own documentation states there is no automatic migration between its built-in backend and your own Supabase project, in either direction. You end up renting the thing you built.
- ✗

Taking money is still left to you

A generated app almost never ships with billing. Stripe keys, checkout, webhooks, subscription state and the tables behind them are hand-written afterwards — and that is the only difference between an app and a business.
- ✗

The AI layer is hand-built, and it is React-only

Generated apps ship a chat box on one hard-coded model; provider choice, tool access, retrieval and reusable domain logic are still written by hand. And every builder in the category generates React — a Django, Rails, Laravel or Spring Boot team is not served at all.

A Deployed AI SaaS **In a Day.** Because the Hard Parts Are Already **Built.**

It clones a base repository that already builds — Supabase auth, Postgres with row-level security, Stripe billing and one-click Vercel deploy already wired — then writes the user's product into it. They bring their own keys. A day of prompting later: a full-stack, deployed AI SaaS they own outright.



65 ready templates, not a blank page

Six Next.js SaaS presets and 59 starters in ten languages — 23 of them AI SaaS dashboards — each a real base repository or official CLI, not generated. Supabase auth and database, Stripe billing, shadcn/ui, the AI SDK, MCP wiring and phone, tablet and desktop shells. It builds before the first prompt.



Stripe billing, already wired

Payments ship inside the template rather than arriving as a later chore: checkout, subscriptions, webhooks and the tables behind them. Stripe is also an MCP preset, so the build can create products and prices in the user's own account while it works. Lemon Squeezy and Polar are one radio button away.



Any MCP server, a few clicks

49 presets plus the Composio, Vercel, Smithery and Docker gateways — over 1,000 toolkits behind one key. Connect one and the generated app calls its tools at runtime through its own gateway token, so it needs no MCP client and no provider SDK of its own.



Skills, agents and teams — one click

The builder works out which AI resources this app needs and generates them on request — a scoring Skill, an integration Plugin, an autonomous Agent, a multi-role Team, a knowledge base. Each is saved to the user's workspace, wired into the app, and reusable in the next one.



Grounded in the real documentation

Add a Firecrawl key and the builder scrapes the docs the app depends on — Stripe's API, the auth pattern, the framework's conventions — into the user's knowledge base before it writes code, so it targets the real API surface rather than a recollection of it. Their key, their credits, always.

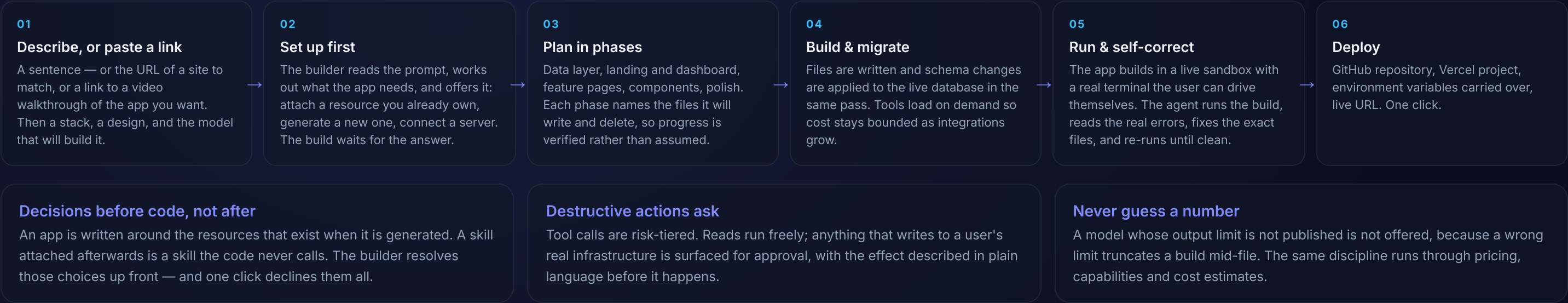


Theirs to keep, and deployed

Their Supabase project, their GitHub, their Vercel, their model keys — and the full source, downloadable as a runnable project or written straight into a folder on their machine at every build. Chat, a real terminal and a live Vercel Sandbox preview in one window; one click to production.

From Sentence to Live URL

The build is a supervised pipeline, not a single prompt. Each stage is checkable, resumable, and visible to the user while it runs.



The Generated App Has an **AI Layer**, Not a Chat Box

Four resource types are generated on demand and compiled into the app. This is the part no other app builder does, and it is why the output is a product rather than a prototype.

Skills

Domain judgement that must be identical everywhere it is applied — scoring, ranking, classification, moderation. Defined once, called from every route instead of re-prompted per call site.

Plugins

A reusable integration with one place to change its credentials and behaviour, rather than the same third-party call pasted across a dozen pages.

Agents

Autonomous workers with their own prompt, tools and provider. The generated app can call them at runtime through a provisioned gateway token — no dependency on the builder.

Teams

Multiple specialised agents with a workflow: sequential, parallel, hierarchical or democratic consensus. The cheapest reliable pattern for work that has to be right.

Why this compounds

- **Resources outlive the app.** A Skill generated for one project is owned by the user and attachable to the next. Every build makes the following build cheaper.
- **Reuse beats regeneration.** The builder checks what the user already owns and offers to attach it before it offers to create anything — so a workspace accrues assets instead of duplicates.
- **The app runs standalone.** Generated apps call AI through their own credentials. Nothing depends on the platform at runtime, which is what makes them shippable to real customers.

Bring your own everything

- **The user's provider key.** Model spend goes to their account at cost. The platform charges for the build, not the tokens.
- **The user's database.** Their Supabase project, their data, their control — the platform never becomes a data custodian.
- **The user's repository.** Code lands in their GitHub and their Vercel account. There is no lock-in to argue about, which shortens every enterprise conversation.

Next.js 15

React 19

TypeScript

Supabase

Stripe

MCP

Vercel

The Build Stops and Asks. Then Remembers.

Every other builder takes one prompt and runs to the end. This one reads the request, works out what the app will actually need — a model to call, a database, payments, a reusable skill, an agent team — and pauses to ask, one question at a time, before a line is written.

← Web Apps

MR Marketing

Draft

config/app.ts ✓

app/page.tsx ✓

components/mobile-tabbar.tsx ✓

components/app-sidebar.tsx ✓

✓ \$ npm i server-only 2>&1 | tail -3

npm warn install-scripts unrs-resolver@1.12.2 (postinstall: node postinstall.js)

npm warn install-scripts

npm warn install-scripts Run "npm install-scripts ls" to review, or "npm install-scripts approve <pkg>" to allow.

✓ Built this step

P Checkpoint

View changes

Restore

Before I build: 3 still to connect or create. I'll scaffold the app either way — sort what you need and say "go".

Worth setting up — 1 thing left to decide

Automation / multi-step logic

This is a multi-step workflow — consider creating an Agent (or Team) to orchestrate it, which the app can call.

Matched: "workflow"

An Agent — a system prompt plus tools, which the app can invoke to handle this on its own. Saved to Dashboard → Agents.

Instant generate

Generate

Create

×

Skip

Generate all — 1 model call

via Claude Opus 5.5 · \$2.08 each

Skip all and build

One question at a time, priced before it runs

Worth setting up — 2 things left to decide

Payments (Stripe)

Add STRIPE_SECRET_KEY + NEXT_PUBLIC_STRIPE_PRICE_ID in the Config tab; the template already includes the billing flow + webhook.

Matched: "plan"

Connect Stripe

Connect PayPal

×

Skip

Agent team (multi-role)

This coordinates several specialized agents — create an Agent Team the app can invoke.

Matched: "team"

A Team — several agents with roles and a coordination protocol, invoked as one. Saved to Dashboard → Agents → Teams.

Generate

opens a dialog

Create

×

Skip

Documentation worth having (2)

✓ Created · 2 documents

I can scrape these into your knowledge base first, so the code targets the real API surface rather than a recollection of it. Nothing is fetched until you approve it.

via Claude Opus 5.5 · \$2.08 each

Skip all and build

Connect, generate, or skip — each is one click

Before I build: 1 you already own and can just attach, 2 still to connect or create. I'll scaffold the app either way — sort what you need and say "go".

Worth setting up — no things left to decide

Reusable AI capability

✓ Attached · Excel Data Analysis

You already have "Excel Data Analysis", "Word Document Creation", "PowerPoint Presentation" and 5 more — attach one to this app instead of creating another.

Matched: "score"

run_command...

Attaches what you already own instead of duplicating it

Before I build: 3 still to connect or create. I'll scaffold the app either way — sort what you need and say "go".

Worth setting up — no things left to decide

Automation / multi-step logic

✓ Created · Campaign Crew Engine Builder

This is a multi-step workflow — consider creating an Agent (or Team) to orchestrate it, which the app can call.

Matched: "workflow"

An Agent — a system prompt plus tools, which the app can invoke to handle this on its own. Saved to Dashboard → Agents.

Generated, named, and saved to your dashboard

● **Asked in the order that makes them answerable**

Provider first, then the services the app talks to, then the skills and agents built on top. One answer changes the next question.

● **A refusal is remembered, not discarded**

"Skip" is stored with the card itself, so it can be reopened and accepted later. The build never asks the same thing twice.

● **Nothing is spent before you say so**

The cost of each generation is shown on the button. Stripe is never called on a first build, even when it is connected.

AI New Agents Hub .dev — WEB APP BUILDER — PRE-SEED

6 / 22

Ten Languages, 59 Starters. Web, Desktop and Mobile.

Every competitor in this category generates React for the browser. A Django team, a Rails shop or a Spring Boot backend gets nothing from any of them — and nobody hands back a desktop installer. Each starter here comes from its framework's **official CLI** or a tagged NewAgentsHub base repository, then runs in the same sandbox, with the same terminal, the same live preview and the same one-click deploy.

59

Official Starters

23

Also Ship as an AI SaaS

10

Languages

6

Platforms — Web, Windows, macOS, Linux, Android, iOS

4

Ways to Import Existing Code

Every starter, by language and platform

TypeScript	Next.js · Vite + React · Vite + Vue · Vite + Svelte · Vite + Solid · SvelteKit · Nuxt · Astro · Astro Blog · React Router · Angular · Bootstrap · Bootstrap SaaS Dashboard
Desktop	NewAgentsHub Desktop · Electron + React · Electron + Vue · Electron + Svelte · Electron + Vanilla
Mobile	Expo (iOS, Android, Web) · Expo Tabs · Expo Blank · Capacitor + Vite
Python	FastAPI · Flask · Django · Streamlit
Ruby	Rails · Sinatra
PHP	Laravel · Symfony
Go · Rust	net/http service, standard library · Axum on tokio, compiled ahead of time
Java · C#	Spring Boot, straight from start.spring.io · ASP.NET Core minimal API
Elixir · Deno	Phoenix on the BEAM, LiveView included · Deno.serve — no install, no build step
+ AI SaaS	23 of the 59 are AI SaaS variants — sign-in, dashboard, streaming AI chat, MCP tools — sitting beside their plain starter. Next slide.

Or bring code you already have

- **GitHub** — any public repository, paste the URL.
- **GitLab** — gitlab.com or your own self-hosted instance.
- **Upload** — a .zip of the project, straight from disk.
- **Local folder** — pick a folder and edit it in place.
- **No URL in mind?** Describe the app; it searches open-source starters and imports the one you pick.

The builder works out the language, framework, package manager, dev command and port for itself — an Electron repository is recognised as a desktop app and its renderer previewed live.

Why this is not just a longer list

- **Nothing rots.** CLI starters are whatever the maintainers ship today; base repositories are pinned, versioned tags upgraded in place.
- **A form built for the framework.** A Django project is asked different questions than an Expo one.
- **One runtime underneath all of it.** Same sandbox, terminal, preview and deploy — the next language is a configuration entry, not a second product.

Why it matters commercially: Lovable, Bolt and v0 sell to the React half of the market, in a browser. The other half — Django, Rails, Laravel and Spring Boot teams — and everyone who needs an installed desktop or phone app is served by none of them. Every starter here carries the same connected backend, the same generated AI layer and the same one-click shipping.

65 Templates. Every One Already a Business.

A template elsewhere is a starting UI. Here it is a starting company: sign-in, a database with row-level security, billing, an AI layer and a phone layout are in the repository before the first prompt — on eighteen frameworks, not one. Pick it from the landing page and it carries straight into the builder.

65

Templates in the Gallery

23

AI SaaS Starters

18

Frameworks With an AI SaaS Variant

6

Next.js SaaS Presets

1

Click From Landing Page to Builder

Next.js presets — cloned, not generated

- **SaaS (recommended)** — Supabase auth + Postgres with RLS, Stripe billing, every shadcn/ui component, TanStack Query.
- **SaaS (no payments)** — the same base with storage, for apps that do not charge yet.
- **AI app** — auth, database, Stripe and a full-bleed streaming AI chat with saved history.
- **Personal Agent (MCP)** — a persistent agent that uses whichever MCP servers the user connects.
- **NewAgentsHub Blank** and **Pure Blank** — the design system, device shells and theme, nothing else.

What every AI SaaS starter ships

- **A landing page, sign-in and a dashboard**, wired to the user's Supabase project.
- **Streaming AI chat** with saved history, **MCP tools** and **knowledge search**, called through the platform's AI routes with an app token — no provider key in the client.
- **Settings** and a tools page, ready to extend by conversation.
- **A real phone layout**: below 760px the sidebar becomes a top app bar and a bottom tab bar, clear of the notch and home indicator.
- **Desktop variants** keep the AI token encrypted on the user's computer, never inside the installer.

AI SaaS, across the stack

React

SvelteKit

Nuxt

Astro

React Router

Angular

FastAPI

Django

Go

Rust

Deno

Rails

Laravel

ASP.NET Core

Spring Boot

Phoenix

Expo

Capacitor

Electron + React

Electron + Vue

Electron + Svelte

Electron + Vanilla

Desktop AI SaaS

Each lives in its own pinned repository; Rails, Laravel and Phoenix keep their own CLI scaffold and the SaaS layer is laid over it.

From the landing page to a running app

- Every template card has **Start with this template**: a visitor signs up and lands in the New App dialog with it already picked.
- Stack, design, model and MCP tools are chosen there — then the build begins.
- A public page per language and platform lists the live registry, so marketing can never advertise a template that does not exist.

Why it matters commercially: the problem slide lists what a real SaaS still needs — billing, auth, an AI layer, ownership. The templates are where that list is crossed off, on the user's own infrastructure. And a Django, Laravel or Spring Boot team gets the same head start as a Next.js one, which no competitor offers.

One Project. Web, Windows, macOS, Linux, Android and iPhone.

Any app built here — or a dedicated Electron, Expo or Capacitor starter — becomes installable software in one click. The builds run on **GitHub Actions in the user's own account**, so they cost the platform nothing, the installers stay private, and the store listings are written for them.

6

Platforms From One Project

9

Desktop & Mobile Starters

24

Real Phones in the Emulator

5

App Stores Prepared, Plus Flathub

\$0

Build Cost to the Platform

Desktop — Windows, macOS, Linux

- **Electron** with React, Vue, Svelte or plain TypeScript, or the **NewAgentsHub Desktop** shell.
- **Press Desktop app**: the Electron shell is generated and pushed to the user's repo — bundled to run offline, or a window onto the live site.
- **.exe, .dmg, .ApplImage and .deb** in one run; MSIX, Snap and Mac App Store packages when the store keys are set.
- Each installer carries **the app's own name, icon and app id**.
- Attached to a **private draft release** and downloaded from NewAgentsHub — nothing public until the user says so.
- The AI token is **encrypted on the end user's computer**; no secret is ever built into an installer.

Mobile — Android and iPhone

- **Expo** (React Native, one codebase for iOS, Android and web) or **Capacitor** wrapping a Vite app — or any web app built here.
- One run gives an **Android APK and an iOS Simulator build**; with the user's keystore and Apple certificate, signed **.aab and .ipa** for the stores.
- An App Store Connect key **uploads to TestFlight**; iOS builds on Xcode 26.
- A real **Android emulator run** in Actions returns screenshots, a video and the crash log.
- The preview shows **12 iPhones and 12 Android phones** at real size — notch, Dynamic Island and safe areas — plus a QR code to open it on your own phone.

Store kit — ready to upload

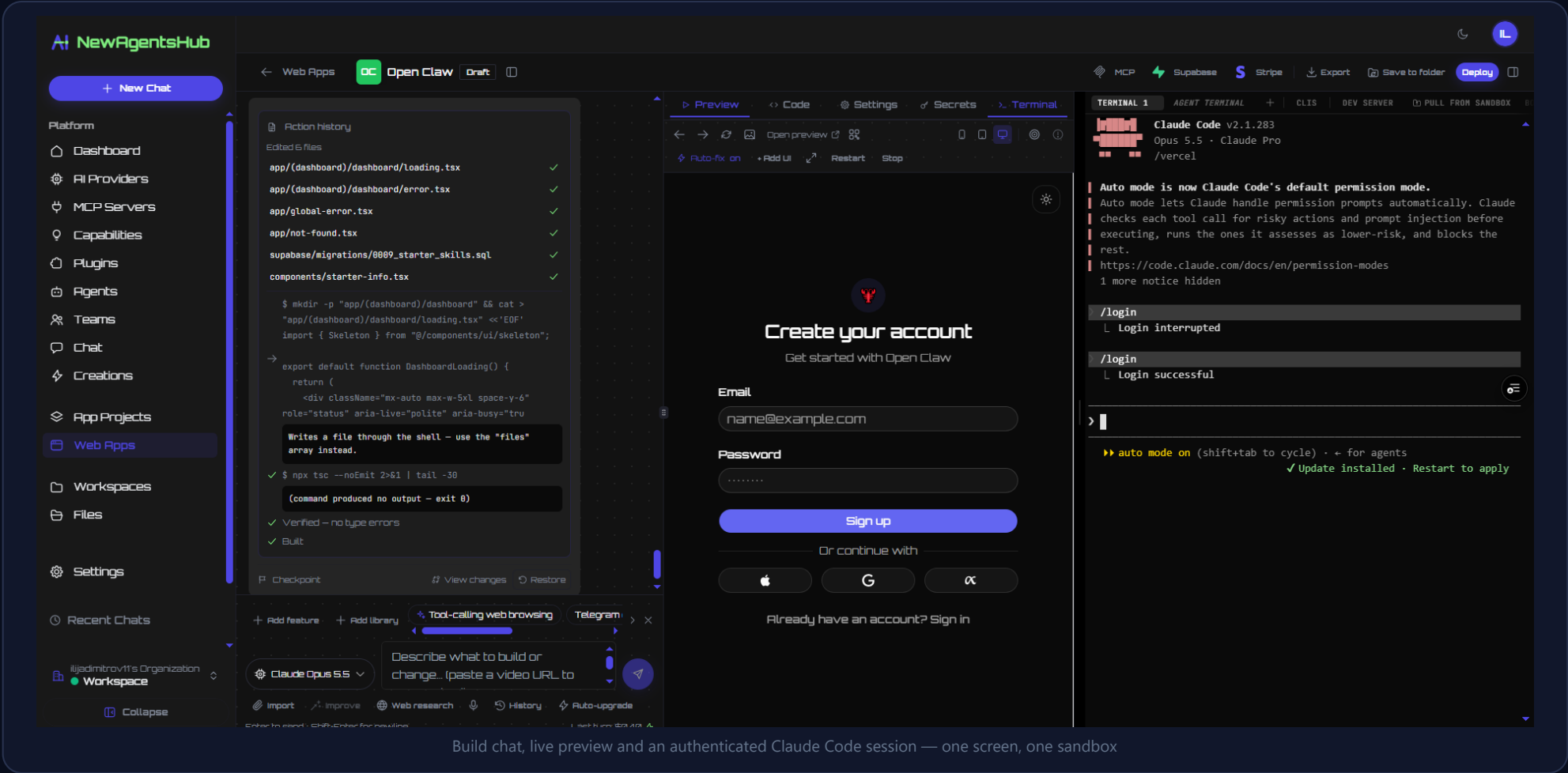
- **Prepare store listing** writes store folders for Windows, macOS, Linux, Android and iOS into the user's repo.
- fastlane supply/deliver, **EAS Metadata**, Microsoft Store listing, **Snap and Flathub** files, and a checklist per store.
- Listing copy and art come from the model chosen in the builder, on **the user's own key**.
- Playwright renders and checks **every screenshot size** each store asks for.
- An optional publish uploads listing text and images only — it **never submits for review** on the user's behalf.
- Signing keys go straight into the repo's secrets; the platform never stores them.

Why it matters commercially: Lovable and v0 stop at the browser. Bolt and Replit reach phones through Expo. None of them builds Windows, macOS and Linux installers — an independent 2026 comparison found no no-code AI builder for desktop apps at all. Here it is one project, six platforms, and builds that run on the user's own CI at no cost to us — so the feature widens the market without widening the cost base.

Sources: lovable.dev guides; support.bolt.new/integrations/expo; docs.replit.com (Native Mobile Apps); Vercel community (v0: no React Native); builderdex.dev, "Best AI App Builder for Desktop Apps", 2026.

Your Own Claude Code, Beside the Live Preview.

A terminal panel runs real agent CLIs — signed in to your own subscription, in the same sandbox that serves the preview. Chat, running app and command line are on one screen, so a fix can be made in whichever of the three is fastest and seen immediately in the other two.



- **Your subscription, not our margin**
`/login` signs the CLI in to the user's own Claude Pro or Max plan. Their agent time is not billed through us and costs us nothing.
- **More than one CLI**
A CLIs tab switches between agent command lines, plus a dev-server tab and a pull-from-sandbox tab against the same running project.
- **The sandbox is shared, not mirrored**
The CLI edits the files the preview is serving. No sync step, no second checkout, no drift between what you ran and what you see.
- **Survives the session**
Terminals reconnect after a reload and the sandbox is reaped on a timer, so an abandoned build stops costing money.

Why it matters commercially: agent CLIs are the fastest-moving part of this market and the most expensive to resell. Letting the user bring their own removes the single largest variable cost in the category — the same reason bring-your-own-key makes this company survivable at zero revenue.

Not a Prototype. A Working Product.

Live at [newagentshub.dev](#), built solo since 29 September 2025 with no prior computer-science education and no outside funding. These are engineering facts, not adoption metrics — there are no users to report yet.

88K

Lines of TypeScript

24

AI Providers Integrated

49

MCP Server Presets

6,500+

Models in Live Catalog

10 Mo

Solo Build Duration

Shipped and working

- Full-stack Next.js 15 generation with a phased, resumable build plan
- Prompt from text, a site URL, or a link to a video walkthrough
- Migrations and row-level security applied to the user's live Supabase project
- 49 MCP presets plus Composio, Smithery, Docker and Vercel gateways
- 24 AI providers with verified context, output limits and live pricing
- Generated skills, plugins, agents and teams, attached to the build
- Web research and documentation scraping into a knowledge base
- Live preview sandbox with a real terminal and an autonomous build-and-fix loop
- Shareable live previews, and one-click GitHub and Vercel deployment

Engineering discipline

- **Seven automated guards** run on every push — design system, accessibility, contrast, icons, prompt caching, stack and model validity
- **Never guess a limit.** A model with no published output cap is excluded rather than estimated, because a wrong number truncates a build
- **Nightly model catalog sync** keeps capability and pricing current without hand-maintained tables
- **Credentials encrypted at rest** (AES-256-GCM), org-scoped access, row-level security throughout

Next.js 15

React 19

TypeScript

Supabase

Vercel

MCP SDK

Sep 2025

First commit. No CS education, no funding, no team.

Nov 2025

Platform live on Vercel. Multi-provider chat shipped.

Jan 2026

Agent teams and the MCP catalog go live.

Apr 2026

App Projects — the first generated, deployable apps.

Aug 2026

Web App Builder: connected backend, generated resources, one-click deploy.

SHIPPED SINCE AUGUST — October 2026

65 templates

— 59 starters, 23 of them AI SaaS dashboards on 18 frameworks, plus 6 Next.js SaaS presets

Desktop apps

— one-click Windows, macOS and Linux installers from any project, built on the user's GitHub

Mobile apps

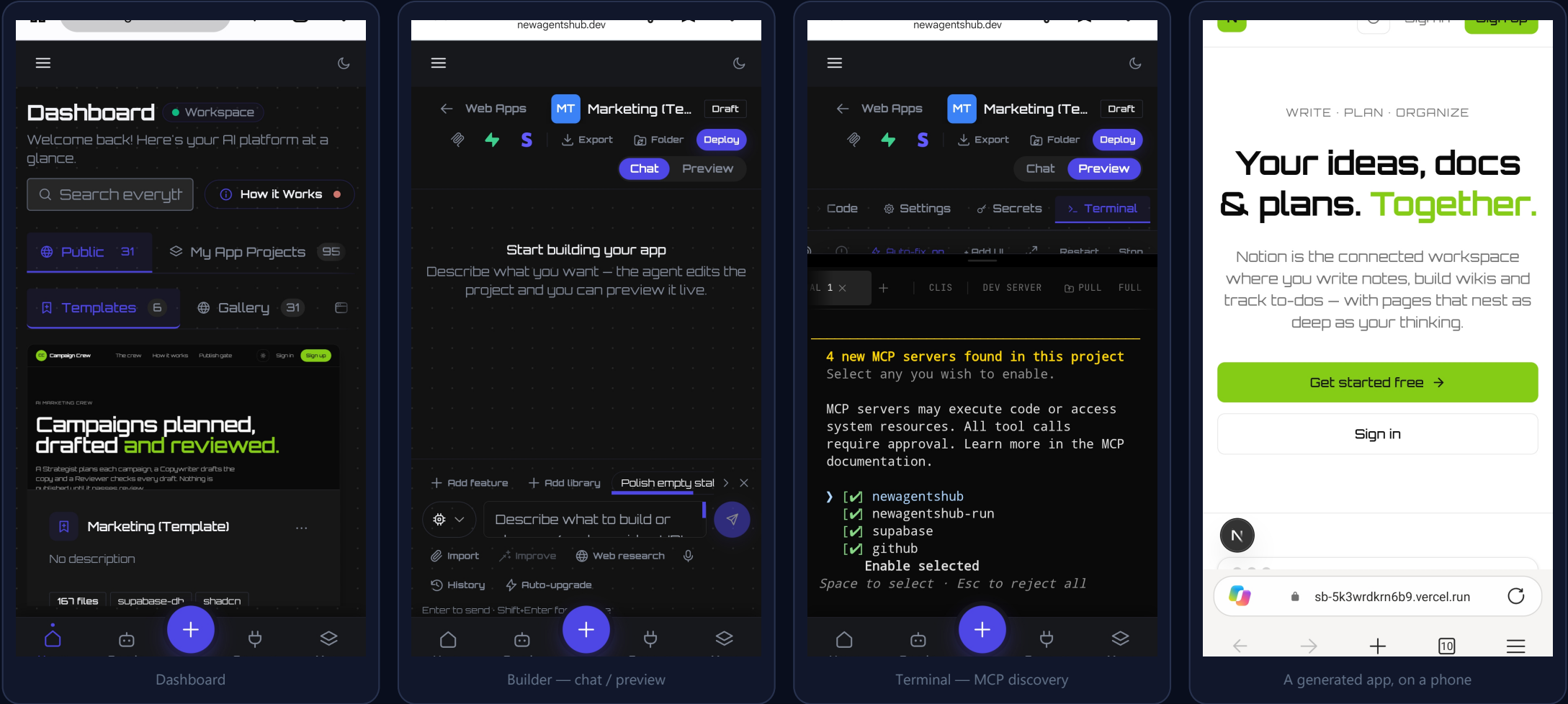
— Expo and Capacitor, Android APK/AAB and iPhone builds, TestFlight upload, 24-phone emulator

Store kit

— listings, screenshots and metadata for Microsoft, Apple, Google Play, Snap and Flathub

The Whole Builder Fits in a Pocket.

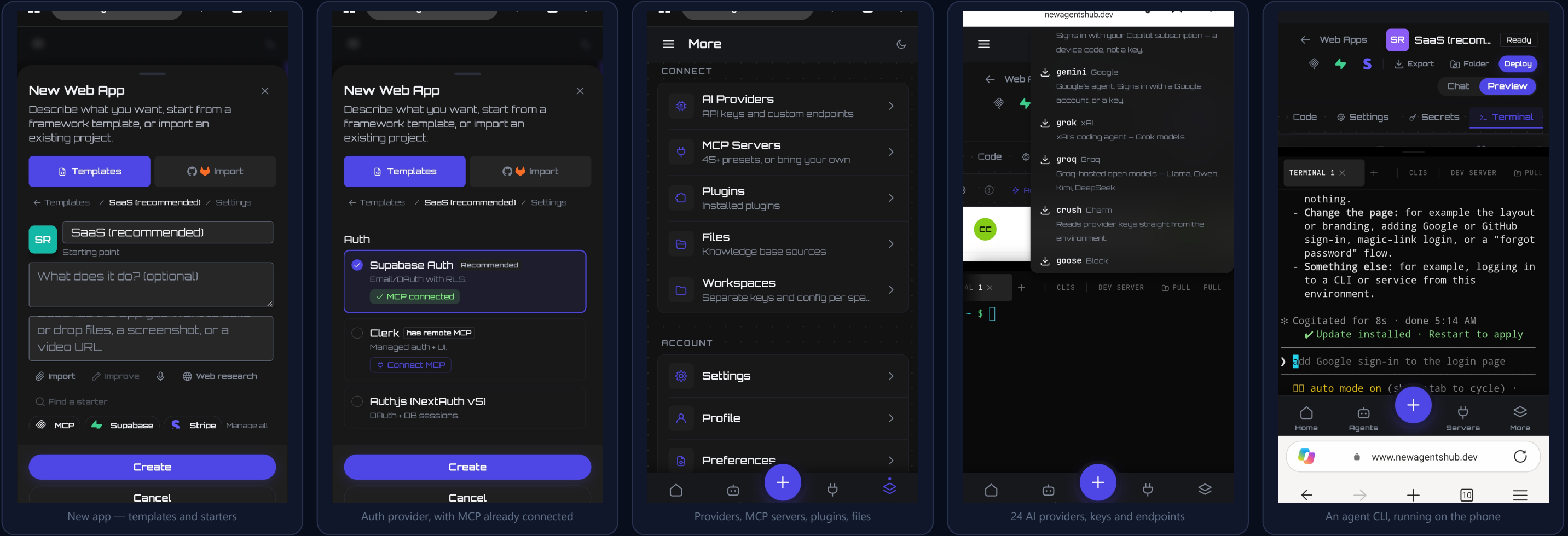
Not a viewer and not a cut-down companion app. The dashboard, the create flow, MCP server selection, provider keys, the file tree and the terminal are all reachable from a phone browser — and so is the app being built.



- **Built responsive, not scaled down**
Phone, tablet and desktop shells are separate layouts in the design system — a mobile tab bar and a sidebar, not one grid squeezed.
- **The generated apps are responsive too**
Every app this builder ships carries the same three shells, so a Next.js app made here already works on a phone the moment it deploys.
- **Where the category is desktop-first**
Replit offers a mobile app; the rest of the field assumes a laptop. A builder you can answer a card on from a train is a different product.

Create, Connect and Configure — All From the Phone.

The flows that normally force people back to a laptop: starting a project from a framework template, choosing an auth provider, enabling MCP servers found in the repository, and managing the AI provider keys the app will run on.



The point: none of this is a separate mobile build. It is the same application, the same routes and the same permissions, rendered through a design system that has had a phone shell and a tablet shell since the first commit — which is also why the apps it generates arrive with both.

Everyone Generates the App. Nobody Connects It — or Ships It Everywhere.

The category is crowded at the front end and empty at the back. Our wedge is the half of the work every competitor leaves to the user — and, since August, every platform the app has to reach.

CAPABILITY	NEWAGENTSHUB	LOVABLE	BOLT.NEW	V0	REPLIT AGENT	CURSOR
Full-stack web app generation	✓ 10 languages	✓	✓	✓	✓	~
Backend on infrastructure you own	✓	~	~	✓	—	—
Auth + billing + deploy in the template	✓	~	—	—	—	—
MCP servers connected to the app	✓ 49 presets	—	—	—	—	~ editor only
Choice of AI provider for the build	✓ 24	—	—	—	~	✓
Generated skills / plugins / agents / teams	✓	—	—	—	—	—
Docs scraped into the build context	✓	—	—	—	—	~
One-click GitHub + Vercel deploy	✓	✓	✓	✓	✓	—
Cost shown before the build runs	✓	—	—	—	—	—
Funding raised to date	\$0	\$200M+	\$105M+	Vercel	\$220M+	\$175M+
SHIPPED SINCE AUGUST — assessed from public sources, October 2026						
AI SaaS templates (auth + AI + phone layout)	✓ 65 · 18 frameworks	~ web	~ web	~ web	~ web	—
Desktop installers — Windows, macOS, Linux	✓ one click	—	—	—	—	~ by hand
Native Android + iPhone builds	✓ Expo + Capacitor	—	✓ Expo	—	✓ Expo	~ by hand
Store listings prepared (stores + screenshots)	✓ 5 stores + Flathub	—	~ guide	—	~ App Store	—
Agent CLIs in the builder, beside the preview	✓ multiple	—	~ shell	—	✓ Claude Code	✓ local
Full builder on phone and tablet	✓	~	~	~	✓ app	—
Generated app ships phone + tablet layouts	✓	~	~	~	~	—

✓ Full support ~ Partial — Not available · Bolt and Replit are credited for Expo mobile apps; Cursor can build anything a developer packages and signs by hand. v0 and Lovable have no native mobile or desktop target (their own docs and community, Oct 2026).

The wedge: the field has caught up on generating a backend, and two competitors now reach phones. What none has done is hand the whole business over — billing in the template, integrations the app itself can call, an AI layer that outlives the build, ten languages, and one project that ships to the web, three desktop systems and both phone platforms, on infrastructure the user owns. One unfunded person shipped what the funded field has not.

MARKET OPPORTUNITY

A Category Being Created in Public

\$65B

AI Platform Market by 2028

38% CAGR (Grand View Research, MarketsAndMarkets)

\$16B

AI Agent Market 2028

\$4.6B

AI Dev Tools 2027

Competitor pricing establishes willingness to pay. Lovable \$25/mo for 100 credits. Bolt \$25/mo for 10M tokens. Replit Core \$20/mo plus usage. v0 \$30 per user in credits. The market already pays a subscription for generated apps that stop at the front end.

Why Now

- **MCP became the standard.** Anthropic's Model Context Protocol is now how AI connects to tools, and every major provider has adopted it. A platform built on MCP inherits every new integration the ecosystem ships.
- **Model capability crossed the threshold.** Frontier models now hold a million tokens of context and emit a hundred thousand in one response. Generating a coherent full-stack application in a handful of passes became possible in the last twelve months, not before.
- **The front end is commoditised.** Four well-funded products generate a React UI from a sentence. Differentiation has moved to what happens after the code exists — which is where this started.
- **The buyer is no longer only a developer.** Operators and domain experts have budget and ideas but cannot run a migration. They are underserved by every tool in the category.
- **Open-weight models collapsed the cost floor.** A full application build now costs cents on an open model. That makes generous free tiers and high-volume usage economically viable in a way it was not a year ago.

Billing Is Live. Flat Plans, Priced on Sandbox Hours.

Five plans are built and running through Freemius today — the checkout works, it simply has not been sold to anyone yet. Every plan is bring-your-own-key, so a plan sells only the two things that cost the platform money — hours of live sandbox and storage. No credit balances, no token metering, no model downgrades.

FREE TRIAL

\$0 / 7 days

- No card required
- All 24 AI providers, your own key
- 3 hours of live sandbox
- 500 MB storage
- The full builder

STARTER

\$8 /mo

- All 24 providers, no markup
- 15 hours of live sandbox / month
- 1 GB storage
- 1 build at a time
- Agents, teams, MCP and deploys

PRO

\$20 /mo

- Everything in Starter
- 50 hours of live sandbox / month
- 10 GB storage
- 2 builds at a time
- Email support, 48h

BUSINESS

\$35 /mo

- Everything in Pro
- 120 hours of live sandbox / month
- 20 GB storage
- 3 builds at a time
- Priority support, 24h

TEAM

\$45 /mo

- 3 seats — \$15 per seat
- Shared workspaces & roles
- 40 hours of sandbox / month, per seat
- 25 GB storage, per seat
- 5 builds at a time · priority support

The trial is the acquisition engine

- Seven days, no card required.** The trial runs on the user's own API key; its only cost to the platform is three hours of sandbox, so it cannot be abused into a loss.
- Hours, not balances.** A plan is a number of hours a buyer can hold in their head — and the same number caps what each user can cost the platform in a month.
- Team is the agency wedge.** \$45 for three seats — \$15 a seat — with shared workspaces and roles, aimed at the segment most likely to buy first and the one that expands by adding seats.

Why the model should hold

- Inference is not the cost centre.** AI runs on the user's own key at their provider's price, with no markup, so gross margin does not move when a model reprices — and every plan gets frontier models.
- Profitable at 100% use.** At ~\$0.17 a sandbox hour, Starter costs ≈ \$2.60 to serve, Pro ≈ \$8.50, Business ≈ \$20.40 and Team ≈ \$20.40 — every plan stays above cost after the processor's cut even if every hour is used.
- Priced to be an easy yes.** \$8 to enter against \$25 at Lovable and Bolt, \$20 at Replit and \$30 per user at v0. Pro at \$20 matches what the market already pays.
- Expansion is structural.** Every generated Skill, Agent and Team is a reusable asset in the workspace; usage grows with the number of apps, and generated resources are the natural path to a marketplace take rate.

What the month-12 target actually requires

750

Pro subscribers alone

429

Business subscribers alone

334

Team accounts alone

OR A BLENDED 1,040 CUSTOMERS

600 Starter · 360 Pro · 50 Business · 30 Team = **\$15,100 MRR**

The point of this arithmetic: \$15K MRR is roughly a thousand customers in a market of millions of people who want an app and cannot build one. Entry sits well below the field — ARPU traded for the reach an unknown product needs — so the customer count is larger and each one is an easier sell. It is a distribution problem of a very ordinary size, and it is the entire thing this raise is being asked to solve.

No Company. No Users. A Product That Works.

Every investor will establish this in the first ten minutes. Better that it comes from us, on one slide, with nothing hidden behind a metric.

NOT YET	Legal entity	Not incorporated. There is no company, no cap table and no prior investors — an investor coming in now defines the structure rather than inheriting one. Incorporation is the first line item in the use of funds.
NOT YET	Users	Zero registered users and no pilot customers. The platform is live and public, but nothing has been marketed — the last ten months went entirely into building, not selling.
NOT YET	Revenue	Nothing has been sold. Not a pricing problem or a plumbing problem — the plans exist and the checkout works. Nobody has been asked to buy yet.
NOT YET	Team	One person, no employees and no co-founder. Concentration risk is real, and the first engineering hire is funded by this round.
WORKING	Billing	Live through Freemius. Five plans built and active — a 7-day free trial with no card, three individual plans from \$8 to \$35, and a \$45 Team plan with three seats and shared workspaces. Priced on hours of live sandbox, with AI on the user's own key. The revenue machinery is ready; only the customers are missing.
WORKING	The product	Deployed and public. It generates full-stack apps in ten languages against real databases, applies real migrations and policies, connects real integrations, and ships them to production URLs, desktop installers and phone builds. 88,000 lines, 24 providers, 49 MCP presets, 65 templates.
PROVEN	Execution	Built solo in ten months with no computer-science education and no outside capital, while the funded competition shipped a narrower product. Whatever else is unproven, the ability to build is not.

The shape of the bet: the technical risk in this business has already been retired — at a cost of zero to the investor. What remains is commercial risk, which is the cheaper and more familiar kind to underwrite. The usual pre-seed question is whether the founder can build the thing. Here it is already built, and the question is only whether anyone will buy it.

One Person Today. Six Roles, in the Order They Get Hired.

The build is finished. What is missing is everyone who sells it, and everything that stops this company depending on one person. The order is not seniority — it is what unblocks the next thing. Capital and a legal entity come first, because every role beneath them is paid for by the round they close.

01 · FIRST HIRE

Investment Manager

Fundraising, VC and angel relations. Capital strategy and round sizing, the investor pipeline from warm intro to term sheet, the data room and due diligence, cap table and dilution modelling, SAFEs and convertibles alongside counsel, investor updates after close. **Every role below is paid for by the round this one closes.** The job description is already written.

02 · MONTH 1–2 · FRACTIONAL

Corporate operations

Incorporation, legal and finance — the executing half of the role beside it. Jurisdiction and entity setup, founder and ESOP structure, cross-border billing and tax on USD subscription revenue, GDPR and the compliance a first enterprise buyer asks about. Nothing can be invested into a company that does not exist yet. Scales into operations as the team grows.

03 · MONTH ~6

Senior full-stack engineer

TypeScript, Next.js, Postgres and LLM tooling — ideally a technical co-founder rather than an employee. The first *engineering* hire and the only one the funding plan names outright. 88,000 lines with a single author is a real concentration risk, and every hour spent fixing a build is an hour not spent selling.

04 · FOUNDER-LED FIRST

Growth & go-to-market

Developer and SaaS marketing. Acquisition is the largest line in the plan, but the first months are founder-run: thirty-plus customer interviews, the first launch channels, a funnel instrumented end to end. Then someone fractional who has taken a developer tool from zero to its first thousand users.

05 · PART-TIME

Sales & customer discovery

Aimed at agencies. The \$45 Team plan — three seats with a shared workspace — is the wedge: agencies buy first and expand by adding seats. Someone who can close the first paid accounts this product has ever had.

06 · FRACTIONAL

Design, community, support

A product designer — the weakest area against competitors funded at \$100M+ — plus community and support once there are users to support. Neither is urgent until the first two roles have done their work.

What this list deliberately does not contain: an AI researcher, or a large engineering team. The building is done — 88,000 lines, 24 providers, 49 MCP presets, 59 starters across ten languages, live and deployable today. Every role above exists to turn that into a company that sells it. That is also why the concentration risk on the previous slide is a hiring problem rather than a technical one.

When each role lands, against the 12-month plan



Raising \$250K–\$500K Pre-Seed

The product is built, billing is live, and both cost nothing to build. The capital buys a company, the first users, the first hire, and the first evidence that anyone will pay. This is the full-size plan — smaller entry points, including grants and single angel cheques, are on the next page.

Use of Funds



All four are forward targets, not results. There is nothing to report against them yet.

12-Month Execution Plan



Want to walk through the plan?
30 minutes on video — the product live, the numbers, and any questions.

📺 Book a video call →

The core unlock — and the first step toward Zero-Configuration AI Access, which the next slide sets out in full: the 7-day trial already removes the credit card — at almost no cost, because it runs on the user's own key. What it does not remove is the key itself, and that is the barrier that stops a curious visitor at the door. Funding a first build with no key at all is now affordable, because a complete application generated on a frontier open model costs cents. That is what converts a product nobody has tried into a product with users.

Today You Bring the Keys. Eventually You Bring **Nothing**.

Bring-your-own-key is what makes this company survivable at zero revenue — model spend never touches our books. It is also the last barrier at the front door: a curious visitor with no API key cannot try the product at all. Removing it, in stages, is the growth strategy. We call the destination **Zero-Configuration AI Access**.

HORIZON 1 · TODAY

Bring your own everything

The user supplies a provider key, a Supabase project, a GitHub and a Vercel account. Cost to serve is near zero, gross margin does not move when a model reprices, and a 7-day trial costs the platform almost nothing. This is the only reason \$8 entry pricing is viable at all — and the reason ten months of building cost no investor anything.

HORIZON 2 · FUNDED BY THIS ROUND

The no-key first build

35% of the raise — the largest single line. A visitor with no key at all gets one complete application, generated and deployed on the platform's account. Affordable only because a full build on a frontier open-weight model now costs cents, which was not true twelve months ago. This is what converts a product nobody has tried into a product with users.

HORIZON 3 · NEXT

Zero configuration, end to end

Managed inference on the paid tiers, and a managed database and deployment for users who would rather not run their own. Describe an app, get a URL — no keys, no dashboards, no accounts to create anywhere else. That is the mass-market version of this product, and the version an operator with a budget and no engineering can actually buy.

Why this is a plan, not a wish

- **The cost floor already moved.** A complete application generated on a frontier open-weight model costs cents, not dollars. Funding a stranger's first build stopped being reckless and became arithmetic.
- **Margin is a dial, not a cliff.** Each tier moves from bring-your-own-key to managed independently, priced against measured cost. Nothing is bet all at once, and any step that does not pay for itself is reversible.

What does not change

- **Ownership.** Even on managed inference, the code, the repository and the deployment stay the user's, and the source stays downloadable. Zero configuration removes setup, never ownership — that distinction is the brand.
- **The moat compounds.** Every layer we manage is one more reason a non-technical buyer picks this over a builder that hands back a repository and a to-do list.

The strategic point: bring-your-own-key is the right model for a company with no revenue and no funding. It is the wrong model for the buyer this product is ultimately for — the operator who has an idea and a budget and will never have an API key. This round pays for the first step across that gap, and the market on the far side is an order of magnitude larger.

Who supplies what, at each horizon

	HORIZON 1 — TODAY	HORIZON 2 — THIS ROUND	HORIZON 3 — NEXT
Model API key	The user	Platform, for a first build	Platform, on paid tiers
Database	The user's Supabase	The user's Supabase	Managed, or their own
Repository & deployment	The user's GitHub & Vercel	The user's GitHub & Vercel	Managed, or their own
The source code	The user	The user	The user
Cost to serve one user	Effectively zero	Cents, once	Priced into the tier

Four Entry Points. Each One a Complete Plan.

The round does not depend on a single large cheque. Because the product is already built and costs almost nothing to run, small capital goes a long way — each tier below funds a coherent set of milestones rather than a fraction of a bigger plan.

NON-DILUTIVE

€10K–€50K

Grants & accelerator stipends

- Incorporation and legal setup
- A year of infrastructure
- Free builds for a first cohort

Milestone: a company exists, first few hundred trials, no equity given up.

FIRST CHEQUE

\$25K–\$50K

A single angel

- Incorporation on a SAFE
- Six months selling, not building
- First paid acquisition tests

Milestone: first 100 paying users and the first revenue this product has ever seen.

PARTY ROUND

\$75K–\$150K

Three to six angels, or a syndicate

- Twelve months of solo runway
- Resource marketplace launch
- Part-time growth support

Milestone: \$5K MRR and a retention curve worth showing a fund.

LEAD INVESTOR — THE ASK

\$250K–\$500K

A fund or a lead angel

- 18–24 months of runway
- First engineering hire
- Platform-funded free builds at scale

Milestone: \$15K+ MRR, 5,000+ users, a seed round on real numbers.

Why small money still works here

- **The build is already paid for.** Ten months and 88,000 lines were funded out of pocket. No investor is being asked to pay for the product — only for its distribution.
- **The cost base is near zero.** Users bring their own model keys and their own databases, so the platform carries hosting and little else. Runway per dollar is unusually long.
- **Revenue can start immediately.** Billing is live across five plans. There is no build phase between the cheque clearing and the first customer being charged.

Terms & structure

- **SAFE on incorporation.** The company is formed in the first weeks after the first cheque clears; the instrument converts at the following round.
- **Tiers combine.** A grant plus an angel cheque reaches the same place as a small round. Non-dilutive money is pursued in parallel regardless of who else comes in.
- **A lead sets the terms.** No valuation is being defended here. An investor who wants to price the round is welcome to.
- **Open to a technical co-founder** in place of, or alongside, capital.

What the money is **not** needed for

Building the product

It exists, it is deployed, and it generates and ships real applications today. The usual first eighteen months of a pre-seed have already happened, at no cost to anyone but the founder.

Paying for inference

Users bring their own model keys, so token spend never touches the platform's books. The single largest variable cost in most AI companies is simply absent from this one.

A team, immediately

One person shipped this in ten months and can keep shipping. The first hire accelerates the plan; it is not a precondition for the plan to start working.

Every entry point starts with one conversation.

Grant partner, first angel, syndicate or lead — pick a time and see the product running live.

 **Book a video call →**



The Front End Is Solved. One Person Built the **Other Half**.

A live product that turns a sentence into a deployed web, desktop or mobile app with a real database, live integrations and a generated AI layer. No company yet, no users yet, no revenue yet — raising \$250K–\$500K to build all three around a product that already works.

A category with a hole in it

\$65B market at 38% CAGR. Four funded competitors generate front ends. None connects the database, applies the policies, wires the integrations, or generates the AI layer. The hard half is unclaimed.

Execution already demonstrated

88,000 lines of TypeScript. 24 AI providers. 49 MCP presets. 65 templates in ten languages, shipping to web, desktop and mobile. Built solo in ten months, with no CS degree and \$0 raised.

The ask

\$250K–\$500K pre-seed, on a SAFE at incorporation — and grants, single angel cheques from \$25K, or a small syndicate all work as entry points. Billing is live, so revenue can start the week after the cheque clears.



Ilija Dimitrov
Creator, NewAgentsHub



iljadimitrov@newagentshub.dev



newagentshub.dev



Ilija Dimitrov
linkedin.com/in/ilija-dimitrov

Book a video call →