

SXSW  SYDNEY

WORKSHOP

VIBE CODE TO VIABLE COMPANY
THE ONE DAY MVP.

JOSH GRAHAM (HE/THEY)
FOUNDER / CEO, INSTANT AI

 Instant Ai

INSTANTAI.COM.AU



I WOULD LIKE TO ACKNOWLEDGE THE GADIGAL PEOPLE OF THE EORA NATION, HERE IN SYDNEY – THE TRADITIONAL CUSTODIANS OF THE LAND ON WHICH WE GATHER TODAY. I’D LIKE TO ACKNOWLEDGE THAT SOVEREIGNTY WAS NEVER CEDED AND RECOGNISE THEIR ENDURING CONNECTION TO COUNTRY, AND HONOUR THEIR ROLE IN CARING FOR AND SUSTAINING THESE LANDS FOR WELL OVER 65,000 YEARS. MY DEEP RESPECT GOES OUT TO TO ALL FIRST NATIONS PEOPLES HERE TODAY, AND ACROSS COUNTRY.

ALWAYS WAS. ALWAYS WILL BE.
ABORIGINAL LAND.

“CODING” [VERB]

The process or activity of writing computer applications using coded programming languages.

“VIBE CODING” [VERB]

*The playful art of talking at
your computer until an app
comes out.*

YOU WON'T LOSE YOUR JOB TO AI.
YOU WILL LOSE YOUR JOB TO
SOMEONE WHO KNOWS HOW TO USE AI.

USE YOUR SKILLS.
USE YOUR UNIQUE KNOWLEDGE.
TO SOLVE A PROBLEM THAT YOU FACE.

THIS IS YOUR SUPERPOWER.

DON'T TRY AND THINK LIKE A TECH BRO.
BE YOURSELF. MAKE SOMETHING FOR YOU.

USE YOUR UNIQUE LENS TO CREATE SOMETHING NO ONE ELSE WOULD.

THE *ONE DAY* MVP

FRAMEWORK

(THIS IS HOW I BUILD MY PRODUCTS)



STEP 1

CONCEPTUALISATION



STEP 2

RAPID PROTOTYPING



STEP 3

DEPLOY!

THE ONE-DAY MVP FRAMEWORK

STEP 1: CONCEPTUALISATION

WHAT PROBLEM ARE YOU SOLVING?

Ask yourself:

- Who do you help?
- What pain are they feeling?
- What's the smallest, most magical thing that could solve it?

You don't need a business plan.

You do need clarity on your users, and the problem you're solving for them.

THE ONE-DAY MVP FRAMEWORK

STEP 2: RAPID PROTOTYPE

BUILD UGLY, BUILD FAST, ITERATE ALL DAY.

- WE'RE LIVING IN THE GOLDEN AGE OF NO-CODE AI TOOLS.
- YOUR GOAL IN THIS PHASE ISN'T BEAUTY – IT'S BELIEVABILITY.
- YOU'RE NOT BUILDING THE FINAL PRODUCT... YOU'RE BUILDING THE PROOF THAT IT COULD EXIST.

THE ONE-DAY MVP FRAMEWORK

STEP 2: RAPID PROTOTYPE

THE TOOLS

Instant App

Lovable

replit

Windsurf

bolt.new

CURSOR



CLAUDE
CODE

GEMINI

Codex

BEGINNER

INTERMEDIATE

ADVANCED

THE ONE-DAY MVP FRAMEWORK

STEP 3: REALTIME DEPLOYMENT! SHIP IT!

ONCE IT WORKS LOCALLY, GET IT ONLINE IMMEDIATELY.

THEN SHARE IT WITH FIVE REAL USERS.

GIVE THEM FREE ACCESS.

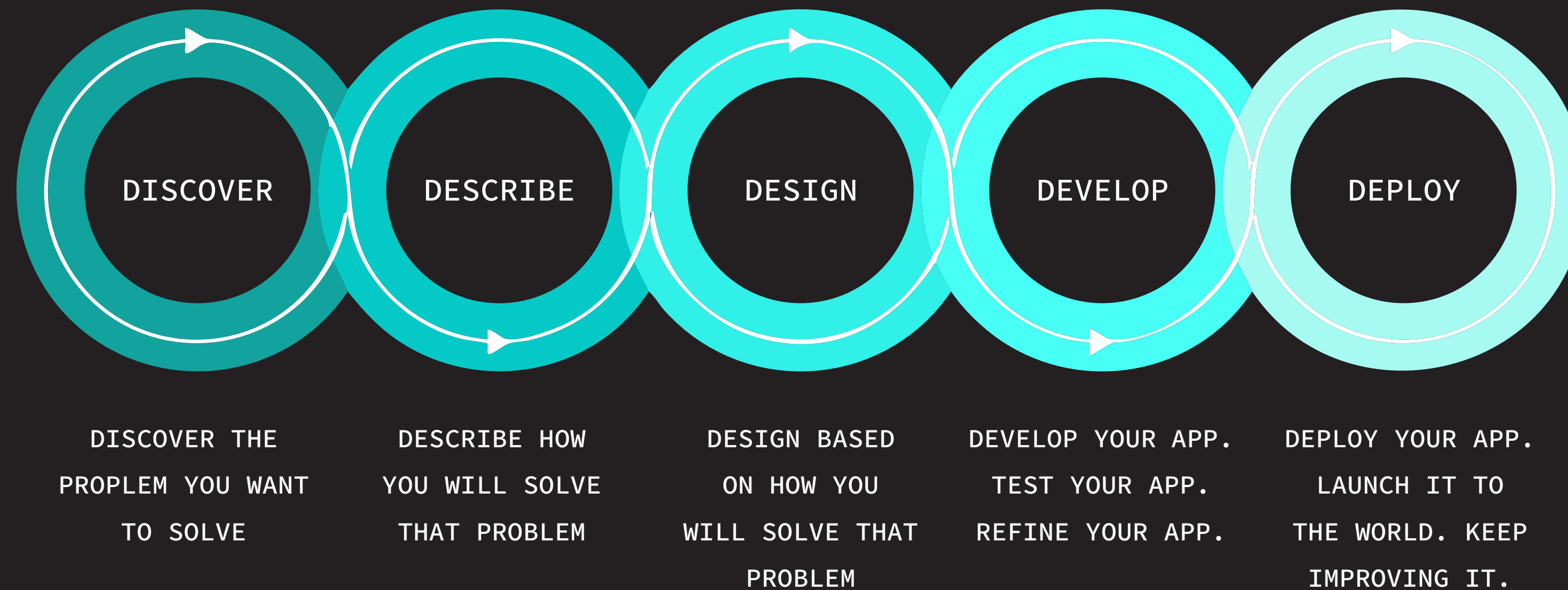
ASK THEM:

- “WOULD YOU USE THIS?”
- “WOULD YOU PAY FOR IT?”
- “WHAT WOULD MAKE IT BETTER?”

THE FEEDBACK YOU GET IS WORTH SIX MONTHS+ OF SECRET PLANNING.

CIRCULAR DEVELOPMENT

(THE 5 D'S OF DEVELOPMENT)



DEVELOPMENT IS ITERATIVE.

TRY SOMETHING. TEST IT.

TWEAK IT. TEST IT.

CHANGE THE UI. TEST IT.

CHANGE THE USERBASE, TWEAK IT. TEST IT.

YOU'RE NOT GOING TO STRIKE GOLD
ON YOUR FIRST GO, AND THAT'S OK.

DON'T BE AFRAID TO PIVOT!!!!

YOU MAY ACCIDENTALLY SOLVE A DIFFERENT PROBLEM!

VIBE CODE TO VIABLE COMPANY: THE ONE-DAY MVP

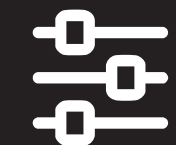
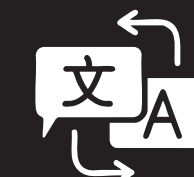
PROMPTING

- *BE REALLY SPECIFIC.*
- *One tweak at a time. Don't make too many changes at once.*
- *Define what you want simply, precisely and in detail.*
- *With each prompt, make it 1% better.*

THEN TEST, GET FEEDBACK, AND TWEAK SOME MORE.

VIBE CODE TO VIABLE COMPANY: THE ONE-DAY MVP
POWER PROMPTS:

Copy & paste these prompts directly into your vibe-code software to get fantastic usable code first time!



ADD USER ACCOUNTS

ROLE: Add secure User Accounts to this repo, adapting to its stack/conventions. DO NOT break builds/routes; keep secrets in ENV; minimal deps.

DISCOVER: Detect runtime, framework, DB/ORM, UI, tests, existing auth. Print plan; proceed.

MINIMUM SLICE: DB: users(id,email unique,email_verified_at,password_hash,created_at), sessions(token,user_id,expires_at), verify_tokens, reset_tokens. Routes: POST /auth/register, /auth/login, /auth/logout; GET /auth/me; POST /auth/request-verify, /auth/request-reset; POST /auth/reset. Sessions: cookie server-side (web) or access+refresh tokens (API). Security: Argon2id/bcrypt; rotate sessions; CSRF (cookie), CORS from ENV; rate-limit logins; validation.

EMAIL: SMTP via ENV; dev logs links. UI (if exists): Login/Register/Forgot/Reset/Verify/Profile.

DX: .env.example, README; seed dev user. TESTS: unit + e2e happy path. ACCEPTANCE: register->verify->login->auth/me->logout; reset flow; no regressions. Output commands + URLs.

ADD SUBSCRIPTION BILLING

ROLE: Add Stripe subscriptions. Use Checkout (signup) + Billing Portal (manage). Don't break builds; minimal deps; secrets in ENV. DISCOVER: runtime/framework, DB/ORM, users, UI, tests. Print plan; proceed.

ENV: STRIPE_SECRET_KEY, STRIPE_PUBLISHABLE_KEY, STRIPE_WEBHOOK_SECRET, APP_BASE_URL, PRICE IDs.

DB: billing_customers(user_id,stripe_customer_id), subscriptions(user_id,stripe_subscription_id,price_id,status,current_period_end), webhook_events(idempotency).

ROUTES:
- GET /api/billing/plans
- POST /api/billing/checkout
- POST /api/billing/portal
- GET /api/billing/status
- POST /api/webhooks/stripe (verify signature; handle checkout.session.completed, customer.subscription.*, invoice.*; idempotent)

CLIENT: pricing UI -> checkout; "Manage billing" -> portal; /billing shows status.

SECURITY: no card data; verified webhooks; rate limits; pin API version.

TESTS/ACCEPTANCE: subscribe in test mode; status updates via webhooks; portal works. Ship .env.example, README, seed, commands.

ADD A USER-SETTINGS PAGE

ROLE: Add a User Settings page (profile, notification prefs, account deletion). Adapt to stack; no breaking changes; secrets in ENV.

DISCOVER: Detect runtime, auth, DB/ORM, UI/forms libs, storage for avatars, email service, tests.

DB: Extend users with profile fields (name, avatar_url), notification_prefs JSON, deleted_at; optional audit log.

ROUTES:
- GET /settings (SSR/API)
- PATCH /settings/profile
- PATCH /settings/notifications
- DELETE /settings/account (soft-delete; revoke sessions; schedule hard-delete)
- GET /auth/me

SECURITY: Server-side validation; CSRF (if cookies); rate limit; confirm via modal + re-auth or email link before delete.

UI: Accessible form sections (Profile, Notifications, Danger Zone); client+server validation; optimistic save; toasts.

ENV: STORAGE_BUCKET for avatars; MAIL*. Log in dev.

TESTS/ACCEPTANCE: Update profile persists; toggles save; delete flow works and logs out; docs + .env.example; print URLs/commands.

ADD MULTI-LANGUAGE SUPPORT

ROLE: Add multi-language support (i18n) with dynamic loading + language switcher. Adapt to stack; keep builds/routes intact.

DISCOVER: Detect SSR/SPA, router, bundler, existing i18n libs, locales, SEO needs.

CHOOSE: Prefer framework-native i18n; else i18next/web or server lib. Lazy-load JSON per namespace/page.

STATE: Store lang in user profile if logged in; else cookie (Secure,SameSite) with URL param fallback.

ROUTES/UI: Language switcher (header/menu). Optional /:locale prefix; redirect on change.

FEATURES: Pluralization, interpolation, date/number formatting, RTL, fallbacks (e.g., en -> en-AU).

IMPLEMENT: Wrap app with i18n provider; replace literals; dynamic imports per route; SSR hydration safe.

ENV: DEFAULT_LOCALE, SUPPORTED_LOCALES.

TESTS/ACCEPTANCE: Strings switch at runtime + refresh; SSR renders correct locale; cookie/profile persists; ally + SEO tags update; docs + .env.example.

ADD AN ADMIN DASHBOARD

ROLE: Add Admin Dashboard with RBAC (roles: admin, editor, viewer). Only admins: protected routes, edit content, view analytics. Adapt to stack.

DISCOVER: Detect auth, DB/ORM, router, UI system, analytics source.

DB: roles(id,name), user_roles(user_id,role_id,unique) or users.role enum. Seed an initial admin.

AUTHZ: Middleware/guards requireRole('admin'|'editor'|'viewer'); combine with resource ownership when needed.

ROUTES/UI:
- /admin (layout+nav)
- /admin/users (list, change roles)
- /admin/content (CRUD; editor+admin)
- /admin/analytics (admin only)
Show "Forbidden" for insufficient role.



SECURITY: Enforce on server; audit log role changes; CSRF if cookies; rate limits.

DX: Feature flag ADMIN_ENABLED; admin-only nav link.

TESTS/ACCEPTANCE: Non-admin blocked; editor edits content not roles; admin sees analytics + updates roles; docs, seeds, .env.example; print URLs/commands.

VIBE CODE TO VIABLE COMPANY: THE ONE-DAY MVP

PRO WORKFLOW – MODULAR.

 <p>ChatGPT</p>	  <p>Claude</p> <p>Instant App</p>	 <p>CURSOR</p>
<p>GPT 5 THINKING / PRO (OPENAI CHAT-GPT) IN CHAT GPT APP</p> <ul style="list-style-type: none"> - Planning - Brainstorm - Research - Code Scaffolds - Big Picture 	<p>SONNET 4.5 (CLAUDE CODE CLI or INSTANT APP) IN-CONTEXT COMMAND LINE</p> <ul style="list-style-type: none"> - Implementing the code that GPT 5 Scaffolds into working code - Iterating 	<p>CURSOR (GPT-5, SONNET 4.5, GEMINI) STANDALONE ADVANCED IDE</p> <ul style="list-style-type: none"> - IDE, serverside agentic. - In-line code changes

CLOSING

AI IS FROUGHT WITH MANY ETHICAL QUESTIONS.

MAKE SURE YOU'RE USING IT ETHICALLY.

BE AWARE OF COPYRIGHT.

IF YOU'RE BUILDING WITH AI, MAKE SURE TO *BUILD THINGS FOR HUMANS.*

CLOSING



AI IS A POWERFUL TECHNOLOGY, PERHAPS THE MOST POWER TECHNOLOGY EVER.

SO GO FORTH AND MAKE THINGS THAT ARE *GOOD FOR THE WORLD.*

Instant Ai

SXSW  SYDNEY



THANK YOU!

JOSH GRAHAM
FOUNDER/CEO, INSTANT AI

josh@instantaigroup.com
[linkedin.com/joshgrahamcreative](https://www.linkedin.com/company/instant-ai/)
+61 402 937 011



*FREE DOWNLOAD
INSTANT APP +
THE 1 DAY MVP PDF*