MVP IN 3 HOURS

Enterprise App Architecture with AI-Powered Tools

From Concept to Shipped Product — A Practical Framework

The 4-Phase Blueprint

Phase 01 · Blueprint — *0–30 min*

- Define one user persona
- One problem statement
- 3-entity data model
- Define success on screen

Phase 02 · Scaffold — *30–90 min*

- AI scaffolding prompt
- Auth + DB + routes in one gen
- 3 core screens only
- Seed DB immediately

Phase 03 · Happy Path — *90–150 min*

- One user journey only
- Entry → process → result
- AI-driven debugging
- No edge cases yet

Phase 04 · Ship It — *150–180 min*

- Self-test 3 times
- One-click deploy
- Share URL immediately
- Collect structured feedback

ELITE ROBOTICS & LABS · MAKING A REVOLUTION

PHASE 01 · Blueprint the Concept

0–30 min · Highest leverage block — don't skip this

Answer these four questions before touching any code:

Who is the user?	One persona. No 'everyone.' Write their job title and primary goal.

What is the ONE problem?	If you can't state it in 10 words, it's two problems.
What does success look like?	Describe exactly what appears on screen when the app works.
What is the data model?	Max 3 entities. Sketch on paper first. Name every relationship.

***PRO TIP** The more precisely you define this phase, the better your AI scaffold prompt will perform.*

PHASE 02 · Scaffold + Core Screens + Data

30–90 min · Generate everything in one shot

AI Scaffold Prompt

Include your full stack in the initial prompt. Name your 3 entities, 3 screens, and auth method up front. Generate auth + DB + all routes in one generation — don't scaffold piecemeal.

Tools: Replit, Lovable, Bolt, v0

Core Screens Only

Build exactly three screens: (1) the landing/entry screen, (2) the main action screen, (3) the result/output view. No settings. No profile. No admin. No onboarding flows.

Rule: if it's not on the happy path, defer it.

Wire Up Data Layer

Use a managed Postgres instance — Supabase or Neon both connect in under 5 minutes on free tiers. Seed 3–5 test records immediately. Never develop against empty state.

Enable row-level security before first deploy.

PHASE 03 · Build the Happy Path End-to-End

90–150 min · One complete user journey: entry → process → result

The journey: User Entry → Form Submit → Backend Processes → DB Write → Response Returns → UI Shows Result

One journey only	Build the single most important user action from start to finish. Not two.
Outcome-oriented prompts	"When user clicks X, system does Y, and shows Z." Specificity drives quality.
Never leave a broken state	Fix every blocker immediately. Paste the error into AI. If AI fails twice, debug manually.
Defer edge cases	Empty states, error messages, rate limits — write them on a parking lot list. Build nothing from it now.

PHASE 04 · Validate & Ship

150–180 min · Zero new features. Deploy. Collect signal.

✓	Run your own happy path 3 times without shortcuts
✓	Verify the DB record is actually saved after the action
✓	Check that the result screen shows accurate data

✓	One-click deploy via Replit / Vercel / Railway
✓	Share the URL with 3 real humans immediately

Feedback Question to Ask Your First 3 Users:

Ask this:	"Did you accomplish [goal]? Yes / No / What stopped you?"
Not this:	"What features do you want?"
Then:	Iterate after feedback — not before. No new features pre-signal.

SCOPE DECISIONS · What's In vs. What's Out

Define this BEFORE you start coding or you'll build the out-of-scope items anyway

✓ Build This (In Scope)

- Core user action — end to end
- 3 screens maximum
- Managed auth (Clerk / Supabase Auth)
- Managed DB (Neon / Supabase Postgres)
- Happy path only — no error states

✗ Defer This (Out of Scope)

- User profile & settings pages
- Admin dashboard / analytics
- Mobile-first design polish
- Email / notification system
- Multi-tenant or team features

Key Takeaway

A deployed MVP with 3 real users testing it is worth more than any amount of polish added before anyone has seen it. Shipped is better than perfect. Validate first, then iterate with confidence.

4 Phases	3 Screens max	1 User journey	3 hrs total