

The Zero-Code App Launch Playbook

Your 5-step playbook to go from idea to a live app in hours with
Claude Code and Railway — no dev team, no code.

01

Describe the App Like a Founder

The quality of your app is decided before Claude Code writes a single file — it's decided by your description. Write out what the app does, who uses it, what screens it needs, and what happens on each one. This becomes your build prompt, and specific beats clever every time.

PASTE THIS

```
Build me a web app called [NAME]. It is for [WHO] and helps them [CORE JOB]. Screens: [list each screen and what's on it]. Users need accounts with email login. Store data in a Postgres database. Use a simple, clean design. Plan the build first, then create the app step by step.
```

💡 QUICK TIPS

- One app, one core job. If your description needs the word 'also' more than twice, cut features.
- Name your users specifically ('freelance photographers'), not generically ('users').

02

Let Claude Code Build It

Open Claude Code, paste your build prompt, and let the agent plan before it builds — say yes to a plan first so you can correct the direction while it's still cheap. Claude Code then generates every file: the pages, the database schema, the login system, all of it. You never need to read the code; you need to read the plan.

💡 QUICK TIPS

- Ask 'explain what you built in plain English' after each big step.
- Commit checkpoints: tell Claude Code to save its progress with git after each working feature.

03

Test, Break, Paste the Error

Run the app locally and click through it like your pickiest user. When something breaks, don't try to understand the error — copy the whole message and paste it back into Claude Code. The agent reads its own errors and fixes its own mistakes. This paste-the-error loop is the entire debugging skill you need.

QUICK TIPS

- Test the boring paths: sign up, log out, log back in, wrong password.
- Describe bugs by behavior: 'the save button does nothing' beats guessing at causes.

04

Connect Railway and Add the Database

Ask Claude Code to create a GitHub repository and push your app to it. Then in Railway, create a new project from that repo and add a Postgres database in one click. Railway hands your app the database connection as environment variables — tell Claude Code 'connect the app to my Railway Postgres database' and it wires everything up.

QUICK TIPS

- Never paste secret keys into your code — Railway's Variables tab is where they live.
- A small app with a database runs on Railway for roughly \$5 a month.

05

Deploy and Share the Live URL

Railway detects your app, builds it, and deploys it to a public URL in a few minutes — and from now on, every push to GitHub redeploys automatically. Generate the domain in Railway's settings, open your live app, and send the link to three real people in your target audience today. An app on your laptop is a project; an app with a URL is a product.

QUICK TIPS

- If the deploy fails, paste Railway's build log into Claude Code — same fix loop as step 3.
- Ship embarrassingly small. Version two is a prompt away.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>