

ZERO TO PUBLIC

03

BUILD THE MVP



Scope one useful version.

Build the MVP

Scope one useful version.

Practical companion edition / October 2026

Copyright 2026 CodeAustral LLC. All rights reserved. Prepared with AI assistance and edited as a practical companion to the original founder field manual.

Use this book to make and record a specific product decision. Examples explicitly labeled hypothetical are teaching examples. Portfolio excerpts are taken from the original manuscript and preserve its publication context; they are not live revenue or traffic claims.

Reading path

A complete small promise

1. Write the result before the feature list
2. Trace the whole workflow
3. Decide what to buy, automate, or do manually
4. Give AI agents a bounded task
5. Treat payment and delivery as one workflow
6. Release with a reversible boundary

Source-backed field notes / Worked example / Eight reusable worksheets

A complete small promise

A first version earns trust by completing a small promise. It does not earn trust merely because it has fewer screens than an established competitor. This book turns a validated direction into a scope you can build, test, support, and release. The working output is a one-page product promise, a workflow, explicit exclusions, a verification plan, and a launch gate.

Choose a job narrow enough that you can observe whether it succeeded. A hypothetical menu builder might promise to turn an approved dish list into a readable PDF. That promise includes input, correction, generation, preview, export, and recovery when generation fails. A login screen and an attractive editor are not a useful first version if the owner cannot get the file.

Treat the boundary as a design decision. State who the product serves, what input it accepts, what result it produces, and which surrounding work remains the user's responsibility. This clarity makes tradeoffs easier and gives the first customer something honest to evaluate. It also prevents "MVP" from becoming an excuse for unreliable payments or missing delivery.

1. Write the result before the feature list

Begin with the situation you validated and the observable result. "The owner can download an accurate menu PDF from a checked dish list" is more useful than "an AI menu dashboard." Write a short acceptance example with real-shaped inputs and outputs. Include a difficult case, such as a long dish name or a missing price, so the example does not quietly assume ideal data.

For every feature, ask which part of the promise it supports. A saved project may be necessary if the work takes multiple sessions. A public gallery may be optional. An account can be necessary for repeat access, but a one-time tool may be able to serve the first job without a complex profile system. Evaluate the workflow rather than copying a conventional application shell.

Create three lists: required for a complete job, useful after the job works, and explicitly excluded. Put reasoning beside exclusions. "Multiple restaurant locations: excluded because the pilot uses one owner and one menu" is a stronger decision than "later." It gives you a condition for reconsidering the feature. Keep that condition visible when a new request arrives.

Define success without inventing a revenue target. The first release may aim to complete the workflow with a few qualified pilot users and record where assistance was required. Completion, accuracy, time to useful output, failure recovery, and willingness to return can be appropriate signals. Revenue is a separate observation when a real purchase occurs; do not relabel a successful demo as demand.