

Business Strategy / Startup Planning

Help founders define the minimum viable product scope by distinguishing must-haves from nice-to-haves and identifying core features only.

Difficulty: Intermediate

Model: GPT-4 / Claude / Gemini

Use Case: Product Definition, Scope Management, Lean Development

Updated: May 2026

Why This Prompt Exists

Most MVPs fail because they're not minimal — founders build too much before validating.

You get:

- 6 months of development on features nobody wants
- bloated products with 50 features instead of 5 core ones
- no clear distinction between must-haves and nice-to-haves
- wasted time on polish before validation
- running out of money before launch

But an MVP is not a prototype.

It is the smallest thing that solves the core problem.

- Core features: must be present to solve the problem
- Nice-to-haves: can be added later (post-validation)
- Out of scope: explicitly excluded (prevents scope creep)
- Validation criteria: what proves the MVP worked

Without MVP discipline, you build too much, too soon.

This framework forces AI to ruthlessly prioritize MVP features.

The Prompt

Assume the role of a lean product strategist who ruthlessly prioritizes MVP scope.

Your task is to define the MVP feature set.

Generate:

1. CORE PROBLEM STATEMENT

- What problem does the MVP solve?

2. CORE FEATURES (MUST-HAVE)

- Features essential to solving the problem
- Why each is essential

3. NICE-TO-HAVE FEATURES (post-MVP)

- Features to add after validation
- Why they can wait

4. OUT OF SCOPE (explicitly excluded)

- Features you will NOT build for MVP
- Why they're excluded

5. MVP SUCCESS CRITERIA

- What proves the MVP works?
- Quantitative metrics

6. TIME TO BUILD ESTIMATE

- Realistic timeline for core features only

INPUTS:

Product Idea:

[DESCRIBE]

Core Problem Solved:

[WHAT PAIN POINT?]

Target User:

[WHO WILL USE IT?]

Desired Features (full wish list):

[LIST ALL FEATURES YOU'D LIKE EVENTUALLY]

Development Resources:

[SOLO / SMALL TEAM / AGENCY]

Time to Launch Target:

[WEEKS OR MONTHS]

RULES:

- Core features must be essential to solving the problem
- If a feature isn't essential, move it to nice-to-have or out of scope
- Out of scope features must be explicitly excluded (prevents creep)
- MVP success criteria must be measurable (not "users like it")
- Time estimate must be for core features only

- Be ruthless: cut everything that isn't essential

How To Use It

- Start with the core problem (not the solution).
- Ask “can the user solve their problem without this feature?”
- If yes, cut it or move to nice-to-have.
- Explicitly list what you’re NOT building (prevents scope creep).
- Launch with core features only, then iterate based on feedback.

Example Input

Product Idea: Mobile app for freelancers to track time and invoice clients

Core Problem Solved: Freelancers waste 5+ hours/week manually tracking time and creating invoices

Target User: Freelance designers, writers, developers with 1-5 years experience

Desired Features: Time tracking (timer), manual time entry, invoice generation, expense tracking, client portal, payment processing, project management, team collaboration, reporting, mobile app, web dashboard, integrations with accounting software

Development Resources: SOLO FOUNDER (basic coding skills)

Time to Launch Target: 3 months

Why It Works

Most MVPs are too big.

This framework improves outcomes by forcing:

- core problem clarity (what you’re solving)
- must-have vs. nice-to-have distinction (prioritization)

- explicit out-of-scope list (scope creep prevention)
- measurable success criteria (validation)
- realistic timeline (feasibility)

Great MVPs don't build everything — they build the smallest thing that solves the core problem.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI business strategy tools, startup frameworks, and practical strategies for founders and entrepreneurs.

Carefully engineered prompts for people doing real work.

Share this:

- [Share on Facebook \(Opens in new window\) Facebook](#)
- [Share on X \(Opens in new window\) X](#)

See also [The Startup Financial Projection Prompt](#)