

## Business Strategy / Pricing Models

Build 3-5 pricing tiers (Good, Better, Best) with feature packaging, price anchoring, and upgrade paths.

Difficulty: Intermediate

Model: GPT-4 / Claude / Gemini

Use Case: Tiered Pricing, Feature Packaging, Price Anchoring

Updated: May 2026

Why This Prompt Exists

Most tiered pricing is poorly designed — confusing feature packaging, no upgrade logic, and weak anchoring.

You get:

- tiers that don't serve different customer segments
- feature overlap between tiers (why upgrade?)
- no clear "most popular" tier (decision paralysis)
- price gaps that don't make sense
- upgrade paths that aren't obvious

But tiered pricing is not random.

It is segmentation by willingness to pay.

- Good tier: price-sensitive, feature-light (acquisition)
- Better tier: value-seeking, feature-rich (most popular)
- Best tier: premium, all features (high LTV)
- Price anchoring: highest tier makes middle tier look reasonable
- Upgrade paths: natural triggers to move up

Without strategic tiering, you leave money on the table.

This framework forces AI to build tiered pricing that maximizes conversion and LTV.

The Prompt

Assume the role of a pricing strategist who builds effective tiered pricing structures.

Your task is to create tiered pricing.

Generate:

1. TIER NAMES (3-5 tiers)
  - Good/Basic/Starter
  - Better/Pro/Most Popular
  - Best/Enterprise/Premium
  
2. FEATURE PACKAGING PER TIER
  - What each tier includes
  - Clear differentiation between tiers
  
3. PRICING PER TIER
  - Monthly price
  - Annual price (if applicable)
  
4. PRICE ANCHORING STRATEGY
  - Which tier is "most popular"
  - How high tier anchors value

## 5. UPGRADE PATHS

- When customers naturally upgrade
- In-app prompts

## 6. TIER COMPARISON TABLE (copy-ready)

### INPUTS:

Product/Service:

[DESCRIBE]

Key Features (list all):

[LIST]

Target Customer Segments (3 segments):

[DESCRIBE]

Price Sensitivity of Each Segment:

[HIGH / MEDIUM / LOW]

Competitor Pricing Tiers (if known):

[DESCRIBE]

### RULES:

- Good tier: feature-light, price-sensitive (acquisition)
- Better tier: most popular, best value (anchored)
- Best tier: premium, all features (high LTV)
- Feature differentiation must be clear (no overlap confusion)
- Price gaps should increase as tiers go up (e.g., \$15, \$29, \$49)

- Most popular tier should be highlighted (default choice)
- Upgrade paths must be natural (usage limits, feature needs)

### How To Use It

- Most customers choose the middle tier (price anchoring).
- Highlight the “Most Popular” tier (default choice reduces friction).
- Feature differentiation must be clear and valuable.
- Price gaps should increase with each tier (e.g., \$15, \$29, \$49).
- Test tier packaging with a small segment before launch.

### Example Input

**Product/Service:** Project management software for small teams

**Key Features:** Task management, file sharing, time tracking, reporting, team collaboration, API access, SSO, priority support, custom branding, advanced analytics, unlimited projects

**Target Customer Segments:** Solo freelancers, Small agencies (2-20 people), Mid-size companies (20-200 people)

**Price Sensitivity of Each Segment:** Solo freelancers (HIGH), Agencies (MEDIUM), Mid-size (LOW)

**Competitor Pricing Tiers:** Asana: Basic (free), Premium (\$13.50), Business (\$24.99), Enterprise (custom)

### Why It Works

Most tiered pricing is confusing.

This framework improves outcomes by forcing:

- clear feature differentiation (no overlap)

- price anchoring (middle tier most popular)
- upgrade path identification (conversion points)
- tier naming strategy (positioning)
- comparison table readiness (execution)

Great tiered pricing doesn't confuse customers — it guides them to the right tier.

## **Build Better AI Systems**

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

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 Competitive Pricing Analysis Prompt](#)