SaaS Growth Pack

32 prompts for onboarding, feature launches, churn reduction, and growth experiments. Copy, customize the `[BRACKETS]`, paste into any AI.

User Onboarding (8 prompts)

Prompt 1: Welcome Email That Drives First Action

```
Write a welcome email for [PRODUCT NAME], a [ONE-LINE PRODUCT DESCRIPTION].

The user just signed up with [SIGNUP METHOD — e.g., Google SSO, email/password, invite link]. Their likely first goal is [PRIMARY USER GOAL — e.g., "create their first project," "connect their data source," "invite a teammate"].

Email structure:
1. Warm, confident subject line (not generic — reference what they can accomplish)
2. One sentence of welcome (no fluff)
3. A single, specific CTA that gets them to [FIRST ACTIVATION MILESTONE]
4. One "PS" line that mentions a quick-win feature they probably don't know about

Tone: [BRAND VOICE — e.g., "casual and direct like Basecamp," "polished and professional like Linear"]. Keep it under 150 words. No "We're thrilled to have you!" or other filler phrases.
```

Example Output: A 120-word email with subject line "Your first [project] is 2 minutes away" that opens with a direct statement of value, links to the specific first action, and closes with a PS about keyboard shortcuts.

Prompt 2: Activation Sequence — Days 1, 3, 5

```
Create a 3-email activation sequence for [PRODUCT NAME] targeting new users who signed up but have NOT yet completed [ACTIVATION MILESTONE — e.g., "created their first workflow," "uploaded a file," "sent their first campaign"].

Context:
- Product: [ONE-LINE PRODUCT DESCRIPTION]
- Activation milestone: [SPECIFIC ACTION]
- Average time-to-activate for successful users: [TIMEFRAME — e.g., "within 48 hours"]
- Top 3 reasons users stall: [REASON 1], [REASON 2], [REASON 3]

For each email provide:
- Send timing (Day 1, 3, 5 after signup)
- Subject line (test two variants: curiosity-driven and value-driven)
- Body (under 100 words each)
- CTA button text

Day 1: Address the most common stall reason with a direct fix.
Day 3: Show social proof — brief example of what an activated user accomplished.
Day 5: Create gentle urgency — reference trial expiration or missed value, not pressure.

Tone: [BRAND VOICE]. Never guilt-trip. Every email should feel like a helpful nudge from a smart friend, not a marketing funnel.
```

Example Output: Three emails with paired subject lines, each under 100 words, progressively shifting from "here's how" to "here's proof" to "here's what you're missing."

Prompt 3: Feature Adoption Nudge — In-App Tooltip Sequence

```
Write a 5-step in-app tooltip sequence that guides a [USER ROLE — e.g., "new project
manager," "first-time creator"] through [FEATURE NAME] in [PRODUCT NAME].

Feature details:
- What it does: [FEATURE DESCRIPTION]
- Why users care: [CORE BENEFIT]
- Common mistake new users make: [MISTAKE]
- Power-user tip most people miss: [ADVANCED TIP]

For each tooltip:
1. Tooltip text (max 25 words — must be scannable while the user is mid-task)
2. Placement hint (what UI element it points to)
3. Trigger condition (what user action or state triggers it)
4. Dismiss behavior (auto-advance, click-to-dismiss, or complete-action-to-advance)

Start with orientation ("Here's where X lives"), move through basic use, address the
common mistake proactively, and end with the power-user tip as a reward.
```

Example Output: Five tooltips like "Click here to add your first data source. We'll auto-detect the schema." with placement on the "+ Add" button, triggered on first dashboard visit.

Prompt 4: Trial-to-Paid Conversion Email

```
Write a conversion email for [PRODUCT NAME] sent [DAYS BEFORE TRIAL ENDS — e.g., "3 days"]
before a user's free trial expires.

User context:
- They HAVE completed: [ACTIONS TAKEN — e.g., "created 2 projects, invited 1 teammate,
used the API once"]
- They have NOT yet: [ACTIONS NOT TAKEN — e.g., "set up integrations, used reporting"]
- Plan they'd upgrade to: [PLAN NAME] at [PRICE]/mo
- Key feature they'd lose access to: [FEATURE]

Structure:
1. Subject line that references THEIR specific usage (not generic "trial ending" language)
2. Open by reflecting back what they've built/accomplished (use the actions above)
3. Name the one thing they'd lose that matters most given their usage pattern
4. Present the upgrade as protecting their work, not buying a product
5. CTA: specific upgrade button text
6. Objection handler: one line addressing the most common hesitation for [PRICE POINT]
SaaS

Tone: confident, not desperate. Under 150 words.
```

Example Output: Email with subject "Your 2 projects and 14 automations are ready for launch" that frames upgrading as "keep what you've built running" rather than "buy our software."

Prompt 5: Personalized Onboarding Checklist

```
Generate an onboarding checklist for [PRODUCT NAME] customized to a [USER PERSONA — e.g., "solo freelancer," "marketing team lead," "developer integrating via API"].

Product: [ONE-LINE PRODUCT DESCRIPTION]
This persona's primary goal: [GOAL]
Features most relevant to them: [FEATURE 1], [FEATURE 2], [FEATURE 3]
Features they should SKIP for now: [IRRELEVANT FEATURE]

Create a checklist of 5-7 steps that:
- Starts with a quick win achievable in under 2 minutes
- Orders steps by value-delivered, not by product architecture
- Includes estimated time for each step (be honest, round up)
- Marks one step as "optional but powerful" for curious users
- Ends with an action that naturally leads to inviting a teammate or sharing output

Format as a clean numbered list with step name, time estimate, and one-sentence description.
```

Example Output: A 6-step checklist starting with "Import your first contact list (1 min)" and ending with "Share your first report with your team (2 min)."

Prompt 6: Onboarding Email for Invited Team Members

```
Write a team-invite email for [PRODUCT NAME]. The recipient was invited by [INVITER ROLE — e.g., "their manager," "a teammate," "the account admin"].

Context:
- Inviter's name: [INVITER NAME]
- Team/workspace name: [WORKSPACE NAME]
- Recipient's expected role: [ROLE — e.g., "editor," "viewer," "contributor"]
- What the team is already using the product for: [CURRENT USE CASE]
- First thing this person should do: [FIRST ACTION]

The email should:
1. Lead with WHO invited them (social proof and context)
2. Explain what the team is already doing in one sentence (they're joining something active, not starting from scratch)
3. Give them ONE specific action to take — not "explore the dashboard"
4. Set expectations for their role permissions in plain language
5. Include a quick-reference link to [RELEVANT HELP DOC OR GUIDE]

Under 120 words. Tone: welcoming but efficient — they're busy, respect their time.
```

Example Output: An email that opens with "Sarah invited you to the Acme Marketing workspace" and directs them to review the Q3 campaign brief that's already in progress.

Prompt 7: Interactive Product Tour Script

```
Write the script for a self-serve interactive product tour in [PRODUCT NAME] for a [USER TYPE — e.g., "first-time user," "user upgrading from free to pro"].

Tour scope: Guide them through [WORKFLOW — e.g., "creating and publishing their first post," "setting up their first automation"].

Product context:
- [PRODUCT NAME] is [ONE-LINE DESCRIPTION]
- This workflow has [NUMBER] steps in the UI
- The most confusing part for new users is: [CONFUSION POINT]
- The "aha moment" happens when: [AHA MOMENT]

For each tour step (aim for 4-6 steps), provide:
1. Step title (3-5 words)
2. Instructional copy (max 30 words — conversational, not robotic)
3. What the user does at this step (click, type, select, etc.)
4. Transition trigger to next step
5. Optional: a "Did you know?" micro-tip if the step has a non-obvious shortcut

Build toward the aha moment. The last step should deliver visible output so the user feels accomplishment.
```

Example Output: A 5-step tour that walks through creating an automation, with step 3 addressing the common confusion around trigger vs. action, and step 5 showing the automation running live.

Prompt 8: Signup-to-Value Time Audit

```
I run [PRODUCT NAME], a [ONE-LINE PRODUCT DESCRIPTION]. Analyze my current onboarding flow and suggest improvements to reduce time-to-value.

Current onboarding steps (in order):
1. [STEP 1 — e.g., "User creates account with email + password"]
2. [STEP 2 — e.g., "Email verification required"]
3. [STEP 3 — e.g., "Profile setup wizard — 4 fields"]
4. [STEP 4 — e.g., "Dashboard loads with empty state"]
5. [STEP 5 — e.g., "User must manually create first project"]
[ADD MORE STEPS AS NEEDED]

Current average time from signup to first value moment: [TIME — e.g., "8 minutes"]
Target: [TARGET TIME — e.g., "under 3 minutes"]
Our activation metric: [METRIC — e.g., "user creates first project"]

For each step, evaluate:
- Is this step necessary before the user gets value? (Cut or defer if not)
- Can this step be automated, pre-filled, or skipped?
- What's the drop-off risk at this step?

Then provide a revised onboarding flow with estimated time savings. Be specific — don't just say "simplify step 3," tell me exactly what to cut.
```

Example Output: A revised 3-step flow that defers email verification, pre-fills profile from SSO data, and replaces the empty dashboard with a template selector, cutting time-to-value from 8 to 2.5 minutes.

Churn Reduction (8 prompts)

Prompt 9: Cancellation Survey That Captures Real Reasons

```
Design a cancellation survey for [PRODUCT NAME] that captures actionable reasons without annoying the user.

Product context:
- [ONE-LINE PRODUCT DESCRIPTION]
- Price: [PRICE]/mo
- Plan being cancelled: [PLAN NAME]
- Top 5 suspected churn reasons: [REASON 1], [REASON 2], [REASON 3], [REASON 4], [REASON 5]
- One thing competitors do that we don't: [COMPETITOR ADVANTAGE]

Survey requirements:
1. Maximum 3 steps (users are already leaving — respect that)
2. Step 1: Single-select reason with 5-6 options (map to your suspected reasons above, plus "Other")
3. Step 2: One open-text follow-up question that changes based on the selected reason (write all variants)
4. Step 3: A conditional save offer based on their reason:
   - Price concern → show discounted rate or downgrade option
   - Missing feature → show roadmap ETA if available
   - Not using it → offer a pause instead of cancel
   - Switched to competitor → ask which one (for competitive intel)
   - Other → thank them and offer a call

Write all copy including button labels, headers, and the final confirmation message. Tone: respectful, zero guilt, zero friction.
```

Example Output: A 3-step modal flow where selecting "Too expensive" triggers a follow-up asking which price would work, then offers a 30% discount for 3 months or a downgrade to the free tier.

Prompt 10: Win-Back Email Campaign — 3-Part Sequence

```
Write a 3-email win-back sequence for [PRODUCT NAME] targeting users who cancelled
[TIMEFRAME — e.g., "30-60 days ago"].

Context:
- Product: [ONE-LINE PRODUCT DESCRIPTION]
- Their most common cancellation reason: [REASON]
- What's changed since they left: [NEW FEATURE OR IMPROVEMENT]
- Win-back offer (if any): [OFFER — e.g., "50% off for 2 months," "free month," "none"]
- Competitor they likely switched to: [COMPETITOR OR "unknown"]

Sequence:
Email 1 (Day 30 post-cancel): "Here's what's new" — lead with the improvement most
relevant to their cancellation reason. No hard sell. Just inform.
Email 2 (Day 45): Social proof — share a specific result from a user who had the same
complaint and came back. Include a metric or quote.
Email 3 (Day 60): Final offer — present the win-back deal (or, if no offer, frame
reactivation as low-risk with a reminder of their existing data/history still saved).

Each email: subject line (2 variants), body under 120 words, single CTA. Never say "We
miss you." Never beg.
```

Example Output: Three emails where Email 1 announces a new Zapier integration (their top-requested feature), Email 2 shares a case study of a returning user who saved 4 hours/week, and Email 3 offers a free month to "pick up where you left off."

Prompt 11: Customer Health Score Framework

```
Design a customer health score framework for [PRODUCT NAME].

Product details:
- [ONE-LINE PRODUCT DESCRIPTION]
- Pricing model: [MODEL — e.g., "per-seat monthly," "usage-based," "flat rate"]
- Key features: [FEATURE 1], [FEATURE 2], [FEATURE 3], [FEATURE 4]
- Current MRR: [MRR RANGE — e.g., "$5K-$20K"]
- Average customer lifetime: [DURATION]

Define health score components across these dimensions:
1. **Usage depth** — which features indicate a sticky user vs. a surface-level one?
2. **Usage frequency** — what's healthy login/activity cadence for this product?
3. **Expansion signals** — what behaviors predict upgrading or adding seats?
4. **Risk signals** — what behaviors predict churn within 30 days?
5. **Support sentiment** — how to weight ticket volume and tone

For each dimension:
- List 2-3 specific, measurable metrics
- Assign a weight (all weights must sum to 100)
- Define thresholds for Green (healthy), Yellow (monitor), Red (intervene)

Then describe the automated actions that should trigger for each color band (e.g., Red + declining usage = CSM outreach + in-app prompt).
```

Example Output: A weighted scoring model where Usage Depth (35%) tracks "features used per week" with Green at 4+, Yellow at 2-3, Red at 0-1, with Red triggering an automated "Need help?" email and CSM Slack alert.

Prompt 12: At-Risk User Outreach Email

```
Write a personal outreach email from a [SENDER ROLE — e.g., "Customer Success Manager," "founder," "Head of Product"] to a user showing churn risk signals.

User context:
- Name: [USER NAME]
- Company: [COMPANY NAME]
- Account age: [DURATION]
- Risk signals detected: [SIGNAL 1 — e.g., "Login frequency dropped from daily to once per week"], [SIGNAL 2 — e.g., "Stopped using reporting feature"], [SIGNAL 3 — e.g., "Removed 2 team members"]
- Their original use case when they signed up: [USE CASE]
- Last support interaction: [SUMMARY OR "none"]

The email must:
1. NOT reveal that you're tracking their specific behavior metrics (that's creepy)
2. Instead, frame it as a natural "checking in" from someone who cares about their success
3. Reference their original use case to show you remember why they joined
4. Ask ONE specific question that invites a real conversation (not "How's everything going?")
5. Offer something concrete: a call, a resource, a configuration tweak
6. Come from a real person with a real reply-to address

Under 100 words. Must feel handwritten, not templated.
```

Example Output: A 90-word email from "Jamie, Head of CS" that says "I remember you set up Acme's reporting pipeline back in March — curious if it's still doing the heavy lifting for your monthly board decks, or if your reporting needs have shifted. Happy to hop on a 15-min call to tune anything."

Prompt 13: Churn Post-Mortem Analysis Prompt

```
Analyze the following batch of cancellation survey responses and extract actionable
patterns for [PRODUCT NAME].

Product: [ONE-LINE PRODUCT DESCRIPTION]
Time period: [DATE RANGE]
Total cancellations: [NUMBER]

Survey responses (paste raw data or summarize):
[PASTE CANCELLATION REASONS — e.g., a list of free-text responses, or categorized tallies
like "Price: 23, Missing features: 18, Switched competitor: 12, Not using: 9, Other: 6"]

Analyze and deliver:
1. **Top 3 churn drivers** ranked by frequency, with a one-paragraph deep-dive on each
(what's really behind this reason?)
2. **Controllable vs. uncontrollable** — which reasons can you actually fix, and which are
outside your control?
3. **Quick wins** — 2-3 changes you could ship in under 2 weeks that would directly
address a top reason
4. **Strategic bets** — 1-2 larger initiatives (1-3 months) that would structurally reduce
churn
5. **Competitive intel** — what are churned users telling you about alternatives?
6. **Segments at risk** — are certain user types (by company size, use case, plan) over-
represented?

Be specific. Don't say "improve the product." Say exactly what to build, change, or cut.
```

Example Output: Analysis showing that 40% of churn is price-related but concentrated in solo users on the Team plan, suggesting a new Solo tier at $9/mo would retain an estimated 15 accounts/month.

Prompt 14: Downgrade Path Design

```
Design a downgrade path for [PRODUCT NAME] that retains revenue instead of losing the customer entirely.

Current pricing:
- [PLAN 1]: [PRICE] — [KEY FEATURES]
- [PLAN 2]: [PRICE] — [KEY FEATURES]
- [PLAN 3]: [PRICE] — [KEY FEATURES]
- Free tier: [YES/NO, and what's included]

When a user on [CURRENT PLAN] clicks "Cancel," design the downgrade flow:

1. **Downgrade offer screen**: Copy that presents the next-lower plan as the smart choice (not a consolation prize). Highlight what they KEEP, not what they lose.
2. **Feature transition plan**: For features they'd lose in the downgrade, explain exactly what happens to their data/settings (archived? deleted? read-only?).
3. **Usage-based suggestion**: If their usage data shows they're under-using their current plan, write the copy that frames the downgrade as "right-sizing" rather than cutting back.
4. **Pause option**: For users who need a break, design a 1-3 month pause flow with reactivation details.
5. **Confirmation page copy**: For users who proceed with cancellation despite offers.

All copy should be empathetic and practical. Never say "Are you sure?" — that's a guilt trip.
```

Example Output: A flow where the downgrade screen says "Your usage fits our Starter plan perfectly — you'd keep projects, team access, and reporting, and save $30/mo" with a clear side-by-side comparison.

Prompt 15: Re-Engagement Campaign for Dormant Users

```
Write a re-engagement campaign for [PRODUCT NAME] users who are still paying but haven't logged in for [TIMEFRAME — e.g., "14+ days"].

Product: [ONE-LINE PRODUCT DESCRIPTION]
These users originally signed up for: [USE CASE]
Features they used most (before going dormant): [FEATURE 1], [FEATURE 2]
New capability they haven't tried: [NEW FEATURE]

Create:
1. **In-app notification** (shown on next login, if they return): 30 words max, welcoming them back with a pointer to something new or relevant.
2. **Email 1 (Day 14 dormant)**: "Your [PRODUCT] is on autopilot" — reassure that everything is still running, then surface one insight or result from their account that might pull them back.
3. **Email 2 (Day 21)**: Quick-win prompt — suggest one 5-minute action they can take right now that delivers visible value.
4. **Email 3 (Day 30)**: Honest check-in — ask if their needs changed and offer to adjust their plan/setup. Include a "pause subscription" option.

Each email: subject line, body under 100 words, single CTA. Never shame them for not logging in.
```

Example Output: Email 1 with subject "Your 3 automations ran 142 times this week" that shows their account is actively delivering value even while they're away, with a CTA to "See your results."

Prompt 16: Annual Plan Upgrade Pitch

```
Write the pitch for upgrading from monthly to annual billing in [PRODUCT NAME].

Context:
- Monthly price: [MONTHLY PRICE]
- Annual price: [ANNUAL PRICE] (saves [PERCENTAGE]%)
- User has been on monthly for: [DURATION]
- Their usage pattern: [PATTERN — e.g., "consistent daily usage," "growing team adding seats"]

Create three versions for different channels:
1. **In-app banner** (max 20 words + CTA button text)
2. **Email** (under 120 words, subject line + body + CTA)
3. **Checkout intercept** (shown when they're about to pay monthly — 40 words max + CTA)

All versions must:
- Lead with the dollar amount saved, not the percentage
- Frame the annual plan as a commitment to their own goals, not to your company
- Include a risk reversal (e.g., prorated refund policy, cancel-anytime guarantee)
- Avoid "limited time" urgency unless [IS THERE A DEADLINE? YES/NO]

Make the math effortless — show exact savings without making them calculate.
```

Example Output: In-app banner: "Switch to annual and keep $108 in your budget this year. Full refund if you change your mind." CTA: "Save $108."

Feature Launches (8 prompts)

Prompt 17: Feature Announcement Email

```
Write a feature announcement email for [PRODUCT NAME].

Feature details:
- Feature name: [FEATURE NAME]
- What it does: [DESCRIPTION — 2-3 sentences]
- Problem it solves: [USER PROBLEM]
- Who benefits most: [USER SEGMENT — e.g., "teams with 5+ members," "users on the Pro plan"]
- How to access it: [LOCATION — e.g., "Settings > Integrations," "new tab in dashboard"]
- Limitations or prerequisites: [ANY CAVEATS — e.g., "Pro plan only," "requires Chrome extension"]

Email structure:
1. Subject line: feature-name-forward, not clickbait (2 variants)
2. Opening line: state the problem this solves (1 sentence, no preamble)
3. What's new: describe the feature in plain language with one concrete example of it in use
4. Screenshot/GIF placeholder: write the alt text and caption for a visual
5. CTA: direct link to try it, with specific button text (not "Learn More")
6. What's next: one sentence teasing the next related improvement (builds anticipation)

Tone: [BRAND VOICE]. Under 180 words. No "We're excited to announce" — just show the value.
```

Example Output: Email with subject "New: Auto-assign tasks based on workload" that opens with "No more manually balancing who does what" and includes a concrete scenario of a 5-person team using the feature.

Prompt 18: Changelog Entry

```
Write a changelog entry for [PRODUCT NAME]'s public changelog.

Update details:
- Type: [FEATURE / IMPROVEMENT / FIX / BREAKING CHANGE]
- Name: [UPDATE NAME]
- Description: [WHAT CHANGED — be technical but accessible]
- Why: [USER PROBLEM OR REQUEST THIS ADDRESSES]
- Who asked for it: [e.g., "Top-requested feature on our roadmap," "Reported by 23 users,"
"Internal improvement"]
- Impact: [WHO IS AFFECTED AND HOW]
- Migration or action needed: [ANY STEPS USERS MUST TAKE, or "None — it just works"]

Format:
- Date header
- Category badge (Feature / Improvement / Fix / Breaking)
- Title (under 10 words)
- Description paragraph (under 80 words, written for end users not engineers)
- "How to use it" — one sentence with a link placeholder
- If breaking change: migration steps in a numbered list

Tone: clear, factual, slightly proud. Write for a technical audience that values brevity.
No marketing fluff.
```

Example Output: "**[Feature]** Auto-assignment by workload — Tasks now automatically route to the team member with the lightest load. Set your balancing rules in Settings > Workflow. No migration needed."

Prompt 19: In-App Feature Notification (Banner + Modal)

```
Write the in-app notification copy for launching [FEATURE NAME] in [PRODUCT NAME].

Feature: [WHAT IT DOES — 1 sentence]
User segment seeing this: [WHO — e.g., "all users," "Pro plan only," "users who have used [related feature]"]
Goal of the notification: [GOAL — e.g., "drive trial of the feature," "inform about a change," "upsell to access"]

Create two variants:

**Variant A — Dismissible top banner:**
- Banner text (max 15 words)
- CTA link text (max 4 words)
- Dismiss behavior: what happens when they close it? (gone forever, reappears once, etc.)

**Variant B — Modal (shown once on login):**
- Headline (max 8 words)
- Body copy (max 50 words)
- Primary CTA button text
- Secondary action (e.g., "Remind me later" / "Learn more" / "Dismiss")
- Screenshot/GIF placeholder with alt text

Both variants should focus on the user's benefit, not the feature's mechanics. If this is a paid-only feature being shown to free users, write a non-pushy upsell variant.
```

Example Output: Banner: "New: Auto-reports are here. Save 2 hours every Monday. [Try it now]" Modal headline: "Your reports, on autopilot" with body copy about scheduling weekly reports.

Prompt 20: Beta Invite Email — Exclusivity Without Arrogance

```
Write a beta invite email for [FEATURE NAME] in [PRODUCT NAME].

Context:
- Feature: [WHAT IT DOES]
- Why this user was selected: [REASON — e.g., "power user," "requested it," "right use case," "random selection from waitlist"]
- Beta cohort size: [NUMBER — e.g., "50 users"]
- What you need from them: [FEEDBACK TYPE — e.g., "bug reports," "workflow feedback," "UX impressions"]
- Known limitations: [WHAT'S ROUGH OR MISSING]
- Beta duration: [TIMEFRAME]
- Feedback channel: [WHERE — e.g., "Slack channel," "in-app feedback widget," "reply to this email"]

Email structure:
1. Subject line that makes them feel chosen but not manipulated (2 variants: one direct, one curiosity-driven)
2. Why they specifically were picked (be genuine — flattery without substance feels hollow)
3. What the feature does and why it matters (3 sentences max)
4. What's NOT ready yet (honesty builds trust and sets expectations)
5. What you're asking of them (be specific about the time commitment)
6. CTA to activate beta access
7. Personal sign-off from [SENDER — e.g., "the PM," "the founder"]

Under 180 words. The tone should feel like an invitation from a colleague, not a marketing campaign. Make them feel like a collaborator, not a test subject.
```

Example Output: Email from "Priya, Product Lead" with subject "You're in: early access to Smart Filters" that explains they were selected because they use filters 3x more than the average user, lists two known rough edges, and asks for 10 minutes of feedback via a Loom recording.

Prompt 21: Feature Launch Internal Brief

```
Write an internal launch brief for [FEATURE NAME] in [PRODUCT NAME] that aligns sales, support, marketing, and product teams.

Feature details:
- What it does: [DESCRIPTION]
- Target users: [SEGMENT]
- Pricing impact: [e.g., "included in all plans," "Pro only," "add-on at $X/mo"]
- Launch date: [DATE]
- Known limitations at launch: [LIMITATIONS]

Sections:
1. **One-liner** — explain this feature to a non-technical colleague in 15 words
2. **Customer problem** — the pain point in the customer's words (write as a quote)
3. **How it works** — 3-4 bullet points, plain language
4. **Competitive positioning** — how does this compare to [COMPETITOR 1] and [COMPETITOR 2]'s version?
5. **Sales talking points** — 3 sound bites reps can use in calls
6. **Support FAQ** — 5 questions support will get on day 1, with approved answers
7. **Marketing assets needed** — checklist of what Hype/marketing should produce
8. **Success metrics** — how we'll know this launch worked (specific numbers, not "good adoption")

Keep it scannable. Use headers and bullets. Total length: under 500 words.
```

Example Output: A brief with sales talking point "This replaces the manual export-to-Excel workaround your prospect mentioned" and success metric "30% of Pro users try Smart Filters within 14 days of launch."

Prompt 22: Feature Comparison Page Copy

```
Write the copy for a feature comparison page on [PRODUCT NAME]'s website, comparing your [FEATURE NAME] against [COMPETITOR 1] and [COMPETITOR 2].

Your feature:
- What it does: [DESCRIPTION]
- Key differentiators: [DIFF 1], [DIFF 2], [DIFF 3]
- Pricing: [HOW IT'S PRICED]

Competitor 1 ([NAME]):
- Their version: [WHAT THEY OFFER]
- Their weakness: [WEAKNESS]
- Their pricing: [PRICE]

Competitor 2 ([NAME]):
- Their version: [WHAT THEY OFFER]
- Their weakness: [WEAKNESS]
- Their pricing: [PRICE]

Create:
1. Page headline (value-driven, not "[PRODUCT] vs. X")
2. Subheadline (who this page is for and what they'll learn)
3. Comparison table with 6-8 rows of capabilities (be honest — mark things competitors do better with a note, not a hidden omission)
4. 2-paragraph narrative below the table explaining WHY you built it differently
5. CTA section with headline + button text
6. SEO meta title and description

Tone: confident and factual. Never trash competitors — let the comparison speak. Honesty about gaps builds more trust than fake checkmark grids.
```

Example Output: A comparison page with headline "Task automation that works without a PhD" and a table that honestly marks competitor 1 as having more integrations while highlighting your simpler setup and lower price.

Prompt 23: Feature Deprecation Notice

```
Write a feature deprecation notice for [FEATURE NAME] being removed from [PRODUCT NAME].

Context:
- Feature being removed: [FEATURE NAME AND DESCRIPTION]
- Why: [HONEST REASON — e.g., "low usage," "replaced by better approach," "technical debt," "security concern"]
- Who's affected: [NUMBER OF USERS AND SEGMENT]
- Replacement (if any): [NEW FEATURE OR WORKAROUND]
- Timeline: [DEPRECATION DATE — how much notice are you giving?]
- What happens to their data: [DATA FATE — migrated, exported, deleted]

Create:
1. **Email notice** (sent immediately): Subject line, body under 150 words. Lead with what they should DO, not with bad news. Include migration steps or export instructions.
2. **In-app banner** (persistent until deprecation date): Max 25 words + CTA to migration guide.
3. **Help doc outline** (5-7 sections): For a full deprecation guide covering why, timeline, migration steps, FAQ, and support contact.

Tone: transparent and empathetic. Acknowledge disruption without over-apologizing. If the replacement is genuinely better, say why — but respect that change is annoying regardless.
```

Example Output: Email with subject "Action needed: CSV Reports is moving to Live Dashboards by March 15" that opens with "Here's how to migrate your 3 saved reports in under 5 minutes" before explaining why.

Prompt 24: Product Hunt Launch Copy

```
Write the Product Hunt launch copy for [PRODUCT NAME].

Product details:
- What it does: [ONE-LINE DESCRIPTION]
- Target user: [WHO IT'S FOR]
- Key differentiator: [WHAT MAKES IT DIFFERENT]
- Pricing: [PRICING MODEL]
- Launch offer (if any): [SPECIAL DEAL FOR PH COMMUNITY]
- Maker story: [1-2 SENTENCES ABOUT WHY YOU BUILT IT]

Create:
1. **Tagline** (max 60 characters — this appears under the product name)
2. **First comment by maker**: 150-200 words covering the origin story, what problem it solves, what's unique, and the launch offer. End with a specific question to spark discussion (not "What do you think?").
3. **Product description** (appears on the product page): 3-4 paragraphs, under 250 words total. Cover problem, solution, key features (3 max), and who it's for.
4. **5 gallery image captions**: Alt text and descriptions for screenshots/visuals.
5. **3 planned reply templates**: Pre-written responses to common PH comments:
   - "How does this compare to [COMPETITOR]?"
   - "What's on your roadmap?"
   - "What's your pricing?"

Tone: authentic and enthusiastic but not hype-y. PH users smell marketing from miles away. Be a maker, not a marketer.
```

Example Output: Tagline "Stop wasting time on manual reports" with a maker comment that opens "I built this after spending every Monday morning copying data between 4 tabs" and asks "What's the most tedious recurring task in your workflow?"

Help & Documentation (8 prompts)

Prompt 25: Help Article — Step-by-Step Guide

```
Write a help center article for [PRODUCT NAME] explaining how to [TASK — e.g., "set up SSO with Okta," "create a custom dashboard," "configure webhook notifications"].

Context:
- User skill level: [BEGINNER / INTERMEDIATE / ADVANCED]
- Prerequisites: [WHAT THEY NEED FIRST — e.g., "Admin role," "Pro plan," "API key from Settings"]
- Number of steps: [APPROXIMATE STEPS]
- Common mistake: [WHAT GOES WRONG AND HOW TO FIX IT]
- Related help articles: [LIST ANY THAT SHOULD BE LINKED]

Article structure:
1. **Title**: "How to [action]" format
2. **Overview**: 1-2 sentences explaining what this accomplishes and why they'd want to (not just "This article shows you how to...")
3. **Prerequisites**: Bulleted list with links
4. **Steps**: Numbered, with each step containing:
   - The action (bold the clickable element names — e.g., "Click **Settings**")
   - Screenshot placeholder with alt text where the UI isn't self-explanatory
   - A "Note" or "Tip" callout if there's a gotcha at that step
5. **Troubleshooting**: 2-3 common issues with fixes
6. **Next steps**: Link to related article or logical next action

Write for scanning — users arrive frustrated and need answers fast. Front-load the action, minimize setup paragraphs.
```

Example Output: An article titled "How to set up SSO with Okta" with an overview "Connect Okta so your team can log in with their company credentials — no separate password needed" followed by 6 bolded-action steps with troubleshooting for "SAML assertion error."

Prompt 26: FAQ Page Generator

```
Generate a comprehensive FAQ page for [PRODUCT NAME] organized by user journey stage.

Product: [ONE-LINE PRODUCT DESCRIPTION]
Pricing model: [PRICING DETAILS]
Key features: [FEATURE 1], [FEATURE 2], [FEATURE 3]
Top support tickets this month: [TOPIC 1], [TOPIC 2], [TOPIC 3]
Competitor users commonly switch from: [COMPETITOR]
Data/security certifications: [e.g., "SOC 2 Type II," "GDPR compliant," "none yet"]

Generate FAQs in these categories (4-5 questions each):
1. **Getting Started** — signup, first steps, compatibility, free trial questions
2. **Features & Usage** — how specific features work, limitations, tips
3. **Billing & Plans** — pricing, upgrades, refunds, invoices, tax
4. **Security & Privacy** — data handling, compliance, permissions, SSO
5. **Troubleshooting** — common errors, performance issues, integrations

For each FAQ:
- Question: phrased the way a real user would ask (not formal or stilted)
- Answer: 2-4 sentences max. Lead with the direct answer, then add context. Link
placeholders for detailed docs.
- Tag: [BEGINNER], [COMMON], or [TECHNICAL] to help with search/filtering

Include 2-3 FAQs that proactively address objections a prospect would have before
purchasing.
```

Example Output: Getting Started FAQ: "Can I try it without entering a credit card?" Answer: "Yes. The 14-day free trial is fully functional and doesn't require payment info. You'll only be asked to add a card if you choose to upgrade. [Start free trial]"

Prompt 27: Error Message Copy — User-Friendly System Errors

```
Rewrite the following error messages for [PRODUCT NAME] to be clear, helpful, and human.

Current error messages:
1. "[ERROR CODE/MESSAGE 1 — e.g., 'Error 500: Internal Server Error']"
2. "[ERROR CODE/MESSAGE 2 — e.g., 'Invalid input: field validation failed']"
3. "[ERROR CODE/MESSAGE 3 — e.g., 'Rate limit exceeded']"
4. "[ERROR CODE/MESSAGE 4 — e.g., 'Permission denied']"
5. "[ERROR CODE/MESSAGE 5 — e.g., 'Connection timeout']"

For each error, provide:
1. **User-facing message**: What happened in plain English (max 15 words)
2. **Actionable next step**: What the user should do (max 20 words)
3. **Technical detail** (collapsible/optional): The original error code for support
reference
4. **Tone check**: Is this the user's fault, our fault, or a third-party issue? Adjust
tone accordingly:
   - Our fault: apologetic + ETA if possible
   - Their fault: instructive, never condescending
   - Third party: transparent + workaround

Format each as a UI-ready error block with headline, body, and action button text. Brand
voice: [BRAND VOICE — e.g., "friendly and competent," "professional and concise"].
```

Example Output: Error 500 rewritten as headline "Something went wrong on our end" with body "Our team has been notified. Try refreshing the page — if it persists, check our [status page]." Button: "Refresh page."

Prompt 28: Tooltip Microcopy Library

```
Write tooltip microcopy for [NUMBER — e.g., "15"] UI elements in [PRODUCT NAME].

For each tooltip, I'll provide the element. You write the copy.

Elements:
1. [UI ELEMENT 1 — e.g., "The 'Archive' button on a project card"]
2. [UI ELEMENT 2 — e.g., "The 'Team' plan badge next to a feature"]
3. [UI ELEMENT 3 — e.g., "The info icon next to 'API Rate Limit' in Settings"]
4. [UI ELEMENT 4 — e.g., "The toggle for 'Email Notifications'"]
5. [UI ELEMENT 5 — e.g., "The '?' icon on the billing page"]
6. [UI ELEMENT 6]
7. [UI ELEMENT 7]
8. [UI ELEMENT 8]
9. [UI ELEMENT 9]
10. [UI ELEMENT 10]
[ADD UP TO 15]

For each tooltip:
- Copy (max 18 words — users read tooltips in 2 seconds or not at all)
- If the tooltip needs a "Learn more" link, include the link text and destination
description
- If the feature is gated to a specific plan, include the upsell variant (same 18-word
limit)

Rules:
- Never restate the button label (if the button says "Archive," don't start with "Archives
your...")
- Start with what happens, or why they'd use it
- Use sentence case, no periods unless multi-sentence
```

Example Output: Archive button tooltip: "Moves this project out of your active list. You can restore it anytime from the Archive." (15 words)

Prompt 29: API Documentation — Endpoint Reference

```
Write user-friendly API documentation for the [ENDPOINT NAME] endpoint in [PRODUCT NAME].

Endpoint details:
- Method: [GET/POST/PUT/DELETE]
- Path: [URL PATH — e.g., "/api/v1/projects"]
- Purpose: [WHAT IT DOES]
- Authentication: [AUTH METHOD — e.g., "Bearer token," "API key in header"]
- Rate limit: [LIMIT — e.g., "100 requests/minute"]

Parameters:
- [PARAM 1]: [TYPE] — [DESCRIPTION] — [REQUIRED/OPTIONAL] — [DEFAULT VALUE]
- [PARAM 2]: [TYPE] — [DESCRIPTION] — [REQUIRED/OPTIONAL] — [DEFAULT VALUE]
[ADD ALL PARAMS]

Response:
- Success: [STATUS CODE] — [DESCRIBE RESPONSE BODY STRUCTURE]
- Error cases: [LIST COMMON ERROR CODES AND WHEN THEY OCCUR]

Generate:
1. **Endpoint summary** (1 sentence)
2. **Authentication section** with example header
3. **Parameters table** (name, type, required, description, default)
4. **Request example** (cURL + JavaScript/Node.js)
5. **Response example** (JSON with realistic sample data, not "string" placeholders)
6. **Error response examples** (2-3 common errors with JSON bodies)
7. **Usage notes** — gotchas, pagination details, rate limiting behavior

Write for developers who want to copy-paste and go. No walls of text between the code
blocks.
```

Example Output: A clean endpoint reference with a cURL example that includes real-looking data like project names and IDs, plus a JavaScript fetch example, plus error responses for 401 (bad token) and 429 (rate limited).

Prompt 30: Release Notes — Weekly Digest Format

```
Write weekly release notes for [PRODUCT NAME] covering the following changes from this week.

Changes:
- [CHANGE 1 — e.g., "Added: Bulk edit for tasks — select multiple tasks and update status, assignee, or due date at once"]
- [CHANGE 2 — e.g., "Improved: Dashboard load time reduced by 40% on accounts with 1000+ tasks"]
- [CHANGE 3 — e.g., "Fixed: CSV export was missing the 'Created Date' column"]
- [CHANGE 4 — e.g., "Fixed: Notification preferences weren't saving for team admins"]
- [CHANGE 5]
[ADD ALL CHANGES]

Format:
- Date and version number header
- Group by: New Features, Improvements, Bug Fixes (skip empty categories)
- Each item: one-line summary + one sentence of context (who it helps or why it matters)
- Highlight the top change with a brief paragraph (3-4 sentences) and a [Screenshot] placeholder
- End with a "Coming soon" teaser (1 sentence about next week's focus)

Tone: proud of the work, respectful of the reader's time. These notes serve both end users and stakeholders. Under 300 words total.
```

Example Output: Notes opening with "This week's highlight: Bulk edit is here" followed by a 3-sentence description, then grouped items, and closing with "Next week: We're finishing up the Slack integration beta."

Prompt 31: Knowledge Base Article — Conceptual Explainer

```
Write a conceptual knowledge base article for [PRODUCT NAME] explaining [CONCEPT — e.g., "how permissions work," "what webhooks are and how to use them," "understanding your usage dashboard"].

Audience: [USER TYPE — e.g., "non-technical admins," "developers," "mixed — both technical and non-technical team members"]

Concept details:
- What it is: [PLAIN LANGUAGE EXPLANATION]
- Why it matters: [WHY USERS SHOULD CARE]
- How it connects to their daily use: [PRACTICAL RELEVANCE]
- Common misconceptions: [WHAT PEOPLE GET WRONG]
- Related features: [FEATURES THAT CONNECT TO THIS CONCEPT]

Article structure:
1. **Title**: "Understanding [concept]" or "How [concept] works in [PRODUCT]"
2. **TL;DR**: 2 sentences covering the essential point (for scanners)
3. **The concept explained**: 2-3 paragraphs, use an analogy if the concept is abstract
4. **How it works in [PRODUCT]**: Specific to your implementation, with a diagram placeholder
5. **Common scenarios**: 3 real-world examples showing the concept in action
6. **FAQ**: 3-4 questions that arise after understanding the basics
7. **Related articles**: Links to how-to guides that apply this concept

Write so someone can go from "I don't understand this" to "I get it and I know what to do" in a single read. Avoid jargon without dumbing down.
```

Example Output: An article on "Understanding Permissions" that uses the analogy of office keys (some open every door, some just your office) and covers three scenarios: adding a contractor, setting up a read-only stakeholder, and rotating an admin who leaves.

Prompt 32: Support Macro Library

```
Create a library of [NUMBER — e.g., "12"] support response macros/templates for [PRODUCT NAME]'s customer support team.

Product: [ONE-LINE PRODUCT DESCRIPTION]
Support channels: [e.g., "email, live chat, in-app"]
Common ticket categories: [CATEGORY 1], [CATEGORY 2], [CATEGORY 3], [CATEGORY 4], [CATEGORY 5]
Average response time target: [TIME]
Escalation path: [WHEN AND HOW TO ESCALATE]

For each macro, provide:
1. **Macro name** (for internal reference — e.g., "billing-refund-approved")
2. **Trigger scenario** (when to use this macro)
3. **Subject line** (if email) or **opening line** (if chat)
4. **Body** (under 100 words, with [VARIABLE] placeholders for personalization)
5. **Required personalization** — what the agent MUST customize before sending (never send a macro without at least one personal touch)
6. **Tags**: category, priority level, sentiment

Create macros for these scenarios:
- First response acknowledgment
- Bug report received
- Feature request noted
- Billing inquiry
- Refund approved / denied
- Account access issue
- Integration help
- Angry customer de-escalation
- Positive feedback thank-you
- Ticket closure follow-up
- Escalation to engineering
- Out-of-scope request

Tone: warm, competent, concise. Every macro should feel like it was written by a helpful human, not pulled from a template library. Include a "Never say" note for sensitive macros (e.g., never say "Unfortunately" in a refund denial).
```

Example Output: Macro for "angry customer de-escalation" that opens with "I completely understand why this is frustrating — [SPECIFIC ISSUE] shouldn't have happened" and includes a "Never say" note to avoid "calm down" or "per our policy."