

Skillbook Activation Library

One-tap control prompt, quick index, and copy/paste skill templates. | February 2, 2026

One-Tap Master Prompt (paste once at the top of a new chat)

You are my Skillbook assistant.

Rules:

- Default to Audit Mode unless I say Execution Mode.
- Prefer minimal diffs and reversible steps.
- Always include: Plan, Verification, Rollback.
- If information is missing, make best assumptions and label them.
- Output in clean sections, not rambly paragraphs.

Now activate: <SKILL_NAME(S)>

Activation Syntax (copy/paste patterns)

Activate: SKILL_ROUTER

Goal: <what you want>

Constraints: <security / time / tools / budget>

Context: <links, repo, stack, audience>

Output: <what you want me to deliver>

Activate: SKILL_RAG

Goal: Build RAG for <use case>

Constraints: Audit Mode, minimal diffs, production-grade

Context: <data types, size, latency target>

Output: Architecture + eval plan + implementation steps

Activate: SKILL_SPEC + SKILL_ARCHITECT + SKILL_DEPLOY

Goal: Ship <thing>

Constraints: Audit Mode first, then Execution Mode

Output: PRD + architecture + CI/CD + rollback

Quick Skill Reference Index

Build + Engineering	
SKILL_SPEC	SKILL_ARCHITECT
SKILL_LANGGRAPH	SKILL_RAG
SKILL_API DESIGN	SKILL_DB_SCHEMA
SKILL_DEBUG	SKILL_TEST
SKILL_REFATOR	SKILL_DOCS

Ops + Security

SKILL_AUDIT_MODE SKILL_SEC_HARDEN

SKILL_SUPPLY_CHAIN SKILL_DEPLOY

SKILL_OBSERVABILITY SKILL INCIDENT

Business + Growth

SKILL_POSITIONING SKILL_PRICING

SKILL_SALES_SCRIPT SKILL_OUTREACH

SKILL_MARKETING SKILL_PITCH_DECK

SKILL_GRANTS SKILL_CUSTOMER_DISCOVERY

Meta / Control

SKILL_ROUTER SKILL_CHAIN

Skill Card Template (for new skills)

SKILL_NAME:
Purpose:
When to use:
Inputs required:
Assumptions:
Process (steps):
Outputs (format):
Verification:
Rollback:
Failure modes:

Templates

Each template below is designed to be pasted as-is. Fill the blanks, keep Audit Mode default, and stack skills when needed.

SKILL_AUDIT_MODE

Activate: SKILL_AUDIT_MODE
Goal:
Context:
Constraints (default):
- No logins
- No billing/resource creation
- No destructive changes
- Prefer minimal diffs
Output format:
Mode: Audit
Plan:
Findings:
Remediation:
Verification:
Rollback:

SKILL_ROUTER

Activate: SKILL_ROUTER
Goal:
Constraints:
Context:
Output:
Outputs:
- Selected skills + why
- Step plan with checkpoints
Verification: Acceptance criteria
Rollback: Escape hatch

SKILL_SPEC

Activate: SKILL_SPEC
Idea (raw):

Target user:
User pain:
Primary outcome:
Constraints:
Must-have features:
Nice-to-have:
Non-goals:
Success metrics:
Deliverable: PRD (Problem, Users, Use-cases, Requirements, Acceptance Criteria, Risks, Metrics, Milestones)

SKILL_ARCHITECT

Activate: SKILL_ARCHITECT
Inputs:
- PRD or idea summary
- Stack preferences
Output:
1) Component diagram (text)
2) Data flow
3) Interfaces (APIs/events)
4) Storage choices
5) Failure handling
6) Security model
7) Tradeoffs + alternatives
Verification + Rollback included

SKILL_LANGGRAPH

Activate: SKILL_LANGGRAPH
Goal:
Agents (roles):
Tools available:
State shape:
Nodes needed:
Edges / routing logic:
Memory strategy:
Guardrails:
Outputs: Graph plan + minimal implementation outline
Verification: Example runs
Rollback: Disable tool nodes / safe mode

SKILL_RAG

Activate: SKILL_RAG
Goal:
Data sources:
Update frequency:
Latency target:
Quality target:
Outputs:
- Ingestion (parsing, chunking, metadata)
- Index (embeddings + store)

- Retrieval (hybrid, filters, rerank)
- Generation (citations + refusal rules)
- Evaluation (golden set + metrics)

Verification: 10-question eval thresholds
Rollback: keyword-only or disable gen

SKILL_SEC_HARDEN

Activate: SKILL_SEC_HARDEN
System summary:
Assets:
Adversaries:
Entry points:
Outputs:

- Threat model (top risks)
- Severity/likelihood/blast radius
- Remediation checklist (least privilege)
- Secret handling policy

Verification tests + Rollback steps

SKILL_SUPPLY_CHAIN

Activate: SKILL_SUPPLY_CHAIN
Repo type:
Language/runtime:
Package manager:
Outputs:

- Pinning strategy (locks, hashes)
- Integrity checks (SLSA-ish basics)
- CI gates (lint/test/audit)
- Release provenance notes

Verification + Rollback

SKILL_DEPLOY

Activate: SKILL_DEPLOY
Target platform:
Envs (dev/stage/prod):
Secrets source:
Outputs:

- Pipeline stages
- Promotion rules
- Canary/staged rollout
- Observability hooks

Verification: Smoke checks
Rollback: One-command or equivalent

SKILL_OBSERVABILITY

Activate: SKILL_OBSERVABILITY
System:
Key user journeys:

Failure types:
Outputs:
- Logging (structured + redaction)
- Metrics (RED/USE + business)
- Traces (critical spans)
- SLOs + alerts
- Dashboard outline
Verification: Trigger test event
Rollback: Disable noisy alerts

SKILL_DEBUG

Activate: SKILL_DEBUG
Symptoms:
Error logs:
Last known good:
Changes since:
Outputs:
- Likely root causes (ranked)
- Fast checks for each
- Minimal fix (small diff)
Verification: reproduce -> fix -> confirm -> prevent
Rollback: exact revert

SKILL_TEST

Activate: SKILL_TEST
System/feature:
Risks:
Outputs:
- Test strategy (unit/integration/e2e)
- Critical cases + edge cases
- Automation plan + CI gates
Verification: tests run green + coverage targets
Rollback: disable flaky tests temporarily

SKILL_REFATOR

Activate: SKILL_REFATOR
Code area:
Goal (cleanup/perf/safety):
Constraints (minimal diffs):
Outputs:
- Refactor plan (small commits)
- Before/after risks
- Tests to protect behavior
Verification + Rollback

SKILL_DOCS

Activate: SKILL_DOCS
Repo/project:

Audience:
Outputs:
- README (quickstart)
- Runbook (ops)
- Troubleshooting FAQ
Verification: new user can run in <=15 min
Rollback: revert doc changes

SKILL_API DESIGN

Activate: SKILL_API DESIGN
Product/system:
Clients:
Auth constraints:
Outputs:
- Endpoints + schemas
- Auth (keys/OAuth/JWT)
- Versioning + pagination
- Error model + rate limits
Verification: example requests/responses
Rollback: keep backward compatibility

SKILL_DB_SCHEMA

Activate: SKILL_DB_SCHEMA
Entities:
Access patterns:
Scale assumptions:
Outputs:
- Tables + keys + indices
- Migration plan
- Constraints + data integrity
Verification: sample queries meet patterns
Rollback: down-migrations / restore

SKILL INCIDENT

Activate: SKILL INCIDENT
Service:
Severity definition:
Stakeholders:
Outputs:
- SEV playbook (triage -> mitigation -> comms)
- On-call checklist
- Postmortem template
Verification drills + Rollback steps

SKILL POSITIONING

Activate: SKILL POSITIONING
Product:
Audience:

Competitors:
Proof:
Outputs:
- One-liner UVP
- 3 differentiators
- 3 use cases
- Why now
- Anti-positioning
Verification: 5-second repeat test
Rollback: narrow niche

SKILL_PRICING

Activate: SKILL_PRICING
Service/product:
Target budget:
Delivery cost:
Outputs:
- 3-tier offer (Starter/Pro/Elite)
- Setup vs subscription logic
- Risk reversal
- Objection bullets
Verification checklist + Rollback

SKILL_SALES_SCRIPT

Activate: SKILL_SALES_SCRIPT
Offer:
Audience:
Sales context (call/DM):
Outputs:
- Discovery questions
- Qualification + close
- Objection handling
Verification: role-play 2 scenarios
Rollback: simplify pitch

SKILL_OUTREACH

Activate: SKILL_OUTREACH
Offer:
Audience:
Channel:
Tone:
Outputs:
- 3 first messages
- 2 follow-ups
- 1 breakup
- Qualifying questions
Verification: reply/meeting rate targets
Rollback: shorten + tighten niche

SKILL_MARKETING

Activate: SKILL_MARKETING
Brand:
Platform:
Goal:
Outputs:
- 10 hooks
- 5 pillars
- 2-week schedule
- CTA ladder
Verification: KPI targets
Rollback: reduce frequency, reuse winners

SKILL_PITCH_DECK

Activate: SKILL_PITCH_DECK
Company:
Product:
Traction:
Ask:
Outputs:
- Slide-by-slide outline
- Voiceover notes
- Proof checklist
Verification: 3-min + 10-min versions
Rollback: cut fluff

SKILL_GRANTS

Activate: SKILL_GRANTS
Project:
Eligibility:
Region:
Outputs:
- Target list
- Application kit checklist
- Draft narratives + budget outline
Verification: submission readiness checklist
Rollback: pivot to alternate funding

SKILL_CUSTOMER_DISCOVERY

Activate: SKILL_CUSTOMER_DISCOVERY
Target customer:
Hypothesis:
Outputs:
- 10 interview questions
- 5 survey questions
- Recruiting script
- Analysis rubric
Verification: 10 conversations + themes
Rollback: narrower segment

SKILL_CHAIN

Activate: SKILL_CHAIN

Goal:

Constraints:

Skills to chain (optional):

Outputs:

- Ordered skill sequence
- Checkpoints between skills
- Deliverables per checkpoint

Verification gates + Rollback triggers