



COMPLETE REFERENCE GUIDE

The Complete Claude Code Toolkit Guide

14 professional toolkits for Claude Code power users. Skills, hooks, workflow recipes, MCP servers, CLAUDE.md templates, and project starters — everything in one place.

14

TOOLKITS

250+

INDIVIDUAL TOOLS

\$149

COMPLETE BUNDLE

30–90

MINUTES SAVED PER WORKFLOW

Contents

INTRODUCTION

- Why Claude Code needs an infrastructure layer
- How to use this guide
- Installation overview

CORE TOOLKITS (1-7)

- 01 Dev Pro Skills Pack — 10 expert skills + slash commands
- 02 Prompt Packs — 25 expert-grade AI prompts
- 03 CLAUDE.md Pro Templates — 6 stack-specific project templates
- 04 Hook Libraries — 17 pre-built safety & quality hooks
- 05 Workflow Recipes — 7 autonomous multi-step workflows
- 06 MCP Server Templates — 5 plug-and-play MCP servers
- 07 Starter Kits — 5 project starters with CLAUDE.md

SPECIALIST TOOLKITS (8-14)

- 08 AI / LLM App Developer Toolkit — LangChain, LlamaIndex, Vercel AI SDK
- 09 Security / AppSec Pro Toolkit — SAST, threat modelling, compliance hooks
- 10 Data & Analytics Toolkit — dbt, BigQuery, Jupyter, Spark
- 11 Team & Engineering Leadership Toolkit — Shared conventions, onboarding, ADRs
- 12 Agency & Freelancer Toolkit — Client workflows, proposals, SOWs
- 13 Indie Hacker / Solo Founder Toolkit — Stripe MCP, launch recipes, validation
- 14 Mobile Developer Toolkit — iOS, Android, Flutter, React Native

REFERENCE

— **Quick installation reference**

— **FAQ**

— **Getting help & updates**

INTRODUCTION

Why Claude Code needs an infrastructure layer

Most developers use Claude Code the same way they use a chat tool — one prompt, one response, back and forth. That mode captures roughly 10% of what Claude Code can actually do.

The developers shipping fastest in 2025 have built an infrastructure layer around Claude Code. Not the code itself — the scaffolding that makes autonomous, reliable, consistent work possible:

Without infrastructure:

You babysit every command. Claude doesn't know your architecture. Every session starts blank. Quality is inconsistent. You manually review every file write. Tasks that could run autonomously require 10 back-and-forth prompts.

With this toolkit:

Hooks intercept dangerous operations automatically. CLAUDE.md gives Claude full stack context from session start. Workflow recipes execute 15-step autonomous sessions. Quality gates run without supervision.

This guide documents every product in the Complete Claude Code Toolkit — what each one contains, how it works, when to use it, and how to install it.

14

TOOLKITS

250+TOOLS
INCLUDED**17**SAFETY
HOOKS**7**WORKFLOW
RECIPES**5**MCP
SERVERS

INSTALLATION OVERVIEW

How the pieces install

GLOBAL INSTALL (~/.CLAUDE/)

- **Skills & slash commands** — active on every project
- **Safety & git hooks** — protect every Claude session
- **Notification hooks** — desktop + Slack alerts

PER-PROJECT INSTALL

- **CLAUDE.md templates** — drop in project root
- **Quality enforcer hooks** — auto-detect stack
- **Starter kits** — copy entire folder, start coding

- 1 **Download** — purchase on Gumroad, receive instant download link. Single zip, all 14 toolkits organised in folders.
- 2 **Global setup** — copy skills to `~/.claude/skills/`, hooks to `~/.claude/hooks/`. Each folder has a step-by-step install guide.
- 3 **Project setup** — drop the right `CLAUDE.md` into your project root. Claude reads it at session start automatically.
- 4 **MCP servers** — fill in `.env`, run `npm install & npm run build`, add one JSON block to Claude settings. Done.

Setup time: Most products are active in under 10 minutes. Hooks and skills are copy-paste installs. CLAUDE.md templates are a single file drop. MCP servers take the longest but each ships with a complete setup guide.



PRODUCT 01

Dev Pro Skills Pack

10 expert skills + 10 slash commands. Install once, use on every project.

Skills are Claude Code's plugin system. Install a skill once to `~/.claude/skills/` and it becomes a slash command available in every project. Each skill in this pack embeds a professional engineering framework — not a generic prompt, but a structured process that outputs consistent, expert-level results.

THE 10 SKILLS

- **api-design-pro** — REST/GraphQL with OpenAPI 3.1
- **test-generator-pro** — Jest, Vitest, pytest, Go test
- **security-auditor-pro** — OWASP Top 10, severity ratings
- **devops-builder-pro** — Dockerfile, CI/CD, GitHub Actions
- **refactor-pro** — SOLID patterns, code smell detection
- **database-schema-pro** — DDL, indexes, N+1 fixes, migrations
- **performance-pro** — Bottleneck analysis, caching strategies
- **error-handling-pro** — Custom error classes, structured logging
- **pr-review-pro** — BLOCK/SUGGEST/NIT tiered review
- **docs-generator-pro** — JSDoc, README, CHANGELOG

WHAT'S INCLUDED

- 10 SKILL.md files
- 10 matching slash command files
- plugin.json manifest
- Step-by-step install guide

INSTALL LOCATION

- `~/.claude/skills/` — global, all projects

```
# Example: run a security audit
/security-audit src/api/

BLOCK users.ts:47 - SQL injection
SUGGEST session.ts:12 - JWT in env
NIT cors.ts:3 - wildcard origin
```



PRODUCT 02

Prompt Packs

25 expert-grade AI prompts across 5 domains. Works with any AI tool.

A good prompt embeds a professional framework. These 25 prompts have the role, process, and output format built in. Drop in your context and get expert-level output from Claude, ChatGPT, Gemini, or any AI chat tool.

PACK 01 — SALES & OUTREACH

- Cold Email Master (AIDA framework)
- LinkedIn Outreach + follow-up sequence
- Objection Handler (SPIN technique)
- Follow-up Sequence (5-touch cadence)
- Sales Call Prep (discovery questions)

PACK 04 — BUSINESS STRATEGY

- Startup Idea Validator
- Competitive Analysis (2x2 map)
- Pricing Strategy (value-based)
- Investor Pitch Coach (10-slide)
- Business Plan Advisor

PACK 02 — CONTENT CREATOR

- YouTube Script Writer
- Newsletter Writer (story-lead format)
- Blog Post Engine (SEO-aware)
- Viral Hooks Generator (10 formulas)
- Social Media Calendar (30-day)

PACK 05 — DEVELOPER PRODUCTIVITY

- Code Explainer (mental model)
- Architecture Advisor (trade-offs)
- Bug Detective (hypothesis-driven)
- Technical Interviewer + rubric
- System Design Coach

PACK 03 — CAREER ACCELERATOR

- Cover Letter Writer (ATS-safe)
- Performance Review (impact metrics)
- Interview Coach (STAR method)
- Salary Negotiator (market framing)
- LinkedIn Profile Optimizer

Works with any AI:

Claude, ChatGPT, Gemini, Grok — these are Markdown prompt files, not Claude Code-specific. Use them anywhere.



PRODUCT 03

CLAUDE.md Pro Templates

Teach Claude your codebase before you write the first line. Six stacks, all gotchas included.

CLAUDE.md is the file Claude Code reads at the start of every session. Drop one of these into your project root and Claude knows your architecture rules, naming conventions, and the non-obvious gotchas for your stack — no re-explaining every session.

TEMPLATES INCLUDED

- **Next.js 15 SaaS** — App Router, Prisma, Server Actions, Tailwind v4
- **FastAPI Python** — SQLAlchemy 2 async, Pydantic v2, Alembic
- **Django Monolith** — DRF, Celery, service layer architecture
- **Go API** — Chi router, dependency inversion, table-driven tests
- **React Native** — Expo Router, React Query, FlatList performance
- **Express.js** — Express 5, Zod validation, Pino logging, Prisma

EACH TEMPLATE COVERS

- Folder structure with explanations
- Architecture rules Claude must follow
- **8 non-obvious gotchas** per stack
- Exact commands to run (tests, migrations, etc.)
- What to avoid and why
- Naming and style conventions

Also works in Cursor & Windsurf: CLAUDE.md files are read by any AI coding tool that supports project context. Drop the same file in and Cursor reads it automatically.

Install: Copy [CLAUDE.md](#) to your project root. That's all. Claude reads it at session start. Customise the gotchas section to add your own stack-specific rules over time.



PRODUCT 04

Hook Libraries

17 pre-built Claude Code hooks. Safety guardrails, auto-linting, git discipline, and notifications.

Hooks are shell scripts that run automatically at key points in Claude's workflow — before a tool call, after a file write, on session stop. They intercept dangerous operations, enforce quality standards, and alert you when Claude finishes. 17 pre-built hooks. No bash scripting required.

PACK 01 · SAFETY GUARD (5 HOOKS)

- **block-dangerous-rm** — blocks `rm -rf` on source dirs
- **block-env-writes** — prevents `.env` overwrites
- **block-force-push** — blocks force push to main/master
- **secret-scanner** — scans file writes for API keys & tokens
- **prod-db-guard** — blocks destructive SQL on prod connections

PACK 02 · QUALITY ENFORCER (5 HOOKS)

- **auto-lint** — runs ESLint/Prettier/Ruff after every file write
- **auto-test** — runs companion test file after source changes
- **console-log-blocker** — warns on debug statements in commits
- **dependency-audit** — `npm audit` on lockfile changes
- **migration-safety** — blocks edits to existing migration files

PACK 03 · GIT WORKFLOW (4 HOOKS)

- **branch-protection** — blocks direct commits to main/master
- **conventional-commit** — validates Conventional Commits spec
- **pre-commit-tests** — runs full test suite before every commit
- **session-changelog** — appends file changes to CHANGELOG.md

PACK 04 · NOTIFICATIONS (3 HOOKS)

- **desktop-notification** — macOS/Linux system notification on stop
- **slack-webhook** — Slack message with session summary
- **token-cost-tracker** — logs token usage and cost to CSV

`~/claude/hooks/` - global

PreToolUse event

PostToolUse event

Stop event

10 min install



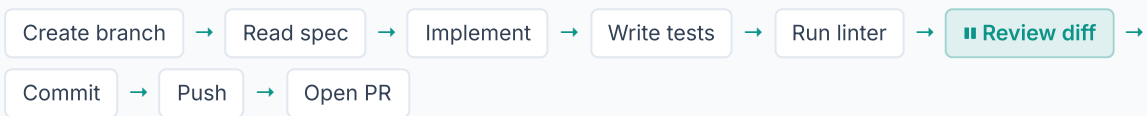
PRODUCT 05

Workflow Recipes

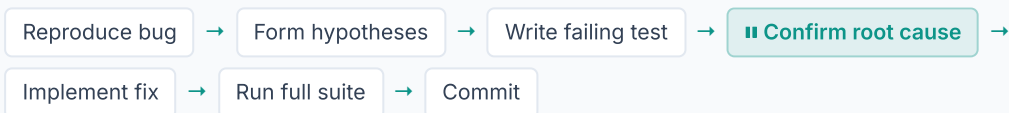
One prompt. 15 autonomous steps. One decision point. Tasks done in a single session.

Each recipe is a copy-paste starter prompt that sends Claude on a multi-step autonomous workflow — pausing only at the one moment where your judgement adds real value. Tasks that take 30–90 minutes of back-and-forth get done in one focused session.

⚡ Full PR Pipeline



🐛 Bug Fix Workflow



THE 7 RECIPES

- **Full PR Pipeline** — branch to open PR, unattended

- **Dependency Upgrade** — audit, batch upgrade, security scan

- **New Feature (TDD)** — RED → GREEN → REFACTOR

- **Bug Fix Workflow** — hypothesis-driven debugging

- **Code Review Prep** — self-review + PR description

- **DB Migration** — impact analysis + rollback script

- **Codebase Onboarding** — full project map + CODEBASE.md

EACH RECIPE INCLUDES

- Workflow table (every step documented)

- Copy-paste starter prompt

- Decision point guide (what to check)

- CLAUDE.md snippet for your project

- Expected output checklist



PRODUCT 06

MCP Server Templates

Give Claude direct access to your database, API, and tools. No boilerplate, no SDK wrangling.

MCP (Model Context Protocol) lets Claude talk directly to external systems. Set up one of these servers and Claude can query your PostgreSQL database, call your internal API, search your file system, or send Slack notifications — all from within a Claude Code session.

THE 5 TEMPLATES

- **Starter Boilerplate** — correct TypeScript foundation: error handling, stdio transport, patterns
- **PostgreSQL Server** — query, list tables, EXPLAIN ANALYZE; read-only mode built in
- **REST API Wrapper** — GET/POST/PUT/PATCH/DELETE for any REST API; configurable auth headers
- **Filesystem Pro** — tree, grep, diff, stats, find; path-sandboxed to your project
- **Webhook Notifier** — Slack (Block Kit), Discord (embeds), any webhook

EACH TEMPLATE INCLUDES

- Complete TypeScript project (src + tsconfig)
- package.json with all dependencies
- .env.example with all options documented
- Claude Code settings JSON snippet
- README with setup and usage examples

```
// Claude Code settings.json snippet
{
  "mcpServers": {
    "postgres": {
      "command": "node",
      "args": ["dist/index.js"]
    }
  }
}
```

Requirements: Node.js 20+ and npm. Fill in credentials in `.env`, run `npm install && npm run build`, add the settings snippet. Most setups take under 15 minutes.



PRODUCT 07

Terminal AI Starter Kits

Start your next project in 5 minutes, not 2 hours. Every kit includes a CLAUDE.md Claude actually understands.

The first hour of any project is the most wasteful hour in development — installing tools, configuring linting, setting up CI, writing CLAUDE.md. These kits eliminate that hour entirely. Every kit has a CLAUDE.md tuned to its stack so Claude understands the project from your very first prompt.

THE 5 KITS

- **Next.js SaaS** — Next.js 15, Prisma 6, NextAuth v5, Tailwind v4, Vitest, CI
- **FastAPI Service** — FastAPI 0.115, SQLAlchemy 2 async, Pydantic v2, Poetry, Ruff, mypy, CI
- **CLI Tool** — Commander v12, Ink v5, Chalk, TypeScript ESM, Vitest
- **Chrome Extension** — Manifest V3, Vite, React, TypeScript — 3-context architecture correct
- **Discord Bot** — discord.js v14, slash commands, deploy script, TypeScript ESM

EVERY KIT INCLUDES

- CLAUDE.md tuned to the stack
- ESLint + Prettier configured
- Conventional Commits (commitlint + husky)
- GitHub Actions CI (lint + test on PR)
- .env.example with all vars documented
- README with setup and deploy instructions



PRODUCT 08

AI/LLM App Developer Toolkit

Build LLM-powered products faster. Tools built for AI app developers, not generic backend devs.

Every company is shipping AI features in 2026. The existing toolkit was built for backend devs — this one fills the gap for LLM app developers: CLAUDE.md templates for AI stacks, skills that understand prompt engineering, workflow recipes for RAG pipelines, and MCP servers for your vector DBs.

CLAUDE.MD TEMPLATES

- **LangChain / LangGraph** — chain composition, memory, async gotchas
- **LlamaIndex** — index types, query engines, storage context
- **Vercel AI SDK** — streaming, tool definitions, edge runtime
- **CrewAI / multi-agent** — agent design, guardrails, shared memory

SKILLS PACK

- **rag-quality-auditor** — chunking, retrieval precision/recall
- **prompt-engineering-reviewer** — scores prompts on 4 dimensions
- **token-cost-optimizer** — identifies bloated context, caching gaps
- **agent-flow-debugger** — traces tool call chains, identifies loops
- **evals-generator** — test datasets + scoring rubrics

WORKFLOW RECIPES

- **RAG Pipeline Build** — ingestion to production checklist
- **Agent Debugging** — reproduce, trace, fix, document
- **Prompt Regression Test** — snapshot, change, diff, score
- **LLM Feature Ship** — build, evals, cost, safety, feature flag

MCP SERVERS

- **Pinecone Server** — upsert, query, describe stats
- **pgvector Server** — semantic search, cosine/L2/inner product
- **LangSmith Server** — runs, traces, datasets, eval results
- **Helicone / cost server** — spend by model, date, user



PRODUCT 09

Security / AppSec Pro Toolkit

Advanced security tooling for AppSec engineers. OWASP goes much deeper here.

The Dev Pro Skills Pack includes a solid OWASP Top 10 audit — this toolkit goes further. Built for security engineers who need threat models, SAST rule generation, compliance enforcement, and structured incident response. One security finding caught before prod is worth orders of magnitude more than this toolkit costs.

ADVANCED SECURITY SKILLS

- **sast-rule-generator** — CodeQL / Semgrep rules from CVE or description

- **threat-model-builder** — STRIDE, data flow, mitigations table

- **dependency-cve-scanner** — maps CVEs to actual code paths

- **security-report-writer** — executive + technical report with CVSS

- **pentest-prep** — recon checklist, attack surface mapping

SECURITY WORKFLOW RECIPES

- **Security Review** — threat model through remediation plan

- **Vulnerability Triage** — CVE intake to fix/accept decision

- **Incident Response** — scope, contain, evidence, post-mortem

- **Secure Code Review** — STRIDE per component + sign-off

COMPLIANCE HOOK LIBRARY

- **hipaa-guard** — detects PHI (SSN, DOB, MRN, diagnosis codes) in file writes

- **pci-dss-guard** — blocks PANs, CVVs in non-encrypted files

- **soc2-audit-logger** — immutable log of every file write and tool call

- **secret-pattern-blocker** — extended: AWS, GCP, Azure, Stripe, Twilio, GitHub

HARDENED CLAUDE.MD TEMPLATES

- **Next.js (hardened)** — CSP, CSRF, auth middleware, rate limiting

- **FastAPI (hardened)** — input validation, JWT, CORS, dependency pinning

- **Django (hardened)** — ALLOWED_HOSTS, HSTS, XSS filter, session security

- **Go API (hardened)** — sanitisation, TLS, error responses that don't leak



PRODUCT 10

Data & Analytics Toolkit

Claude Code for data scientists and analytics engineers. dbt, BigQuery, Jupyter, Spark — all covered.

Data scientists use entirely different stacks — dbt, BigQuery, Jupyter, Spark, Pandas, scikit-learn — and have different workflows: EDA, model training, data validation, schema migrations. This toolkit gives Claude the context and tools to work fluently in that world.

CLAUDE.MD TEMPLATES

- **dbt + BigQuery** — model naming, ref/source patterns, incremental strategy, cost-aware queries

- **Jupyter / Python ML** — notebook structure, reproducibility, venv, model serialisation

- **Spark (PySpark)** — partition strategy, broadcast joins, UDF pitfalls, caching rules

- **Snowflake + dbt** — warehouse sizing, clustering keys, zero-copy cloning, cost guardrails

WORKFLOW RECIPES

- **EDA Recipe** — load, profile, outlier detection, correlation, summary report

- **Model Evaluation** — metrics, confusion matrix, error analysis, vs baseline

- **dbt Model Build** — source to mart, tests, docs, CI, deploy

- **Data Pipeline Debug** — failure point, sample, root cause, fix, backfill

MCP SERVERS

- **BigQuery Server** — query, list datasets, cost estimates, preview

- **Snowflake Server** — query, schema inspection, credit monitoring

- **pgvector Server** — semantic search, cosine/L2 similarity

- **dbt Metadata Server** — lineage graph, test results, run history

HOOK LIBRARY

- **schema-change-guard** — blocks breaking schema changes (column drops, type changes)

- **notebook-auto-doc** — generates markdown summary of all notebooks touched in session

- **query-cost-estimator** — EXPLAIN before BigQuery/Snowflake queries execute; warns on cost

- **data-lineage-tagger** — appends lineage comment to every dbt model Claude modifies



PRODUCT 11

Team & Engineering Leadership Toolkit

Scale Claude Code across your entire engineering team. Shared conventions, onboarding, ADRs, and standards enforcement.

Every other toolkit is for a single developer. This one scales Claude Code across a team — ensuring every engineer has the same conventions, context, and guardrails without a wiki nobody reads. The workflows automate the high-value tasks engineering leads spend the most time on.

TEAM CLAUDE.MD SYSTEM

- **Team conventions template** — architecture rules, naming, PR norms
- **Sync script** — pulls latest team CLAUDE.md from shared repo
- **Per-project override system** — project CLAUDE.md extends team template
- **Onboarding installer** — single command for new hire environment setup

ENGINEERING LEADERSHIP RECIPES

- **New Hire Onboarding** — codebase map to first PR, day one
- **ADR Generator** — options, trade-offs, recommendation, formal doc
- **Sprint Review** — PRs, impact summary, changelog, retro prompts
- **Tech Debt Audit** — complexity, coverage, dependency age, violations

TEAM STANDARDS HOOKS

- **team-conventions-enforcer** — validates writes against team CLAUDE.md rules
- **pr-description-enforcer** — blocks PRs without required template sections
- **cross-team-boundary-guard** — warns on edits to files owned by other teams (CODEOWNERS)
- **session-digest** — structured session summary for async team comms

ENGINEERING PROMPT PACK

- **1:1 Prep** — surfaces work, blockers, growth signals
- **RFC Writer** — problem, solution, alternatives, risks
- **Hiring Bar Rubric** — evaluation rubric for any engineering role
- **Post-Mortem Facilitator** — timeline, blameless analysis, action items



PRODUCT 12

Agency & Freelancer Toolkit

Bill 3 hours of client work in 45 minutes. Claude Code as a business tool, not just a coding assistant.

Freelancers have the highest incentive of any developer persona to buy productivity tools — every hour saved is direct revenue. This toolkit automates the work that surrounds the code: client onboarding, proposals, scope management, deliverable generation, and handoff documentation.

CLIENT WORKFLOW RECIPES

- **Client Onboarding** — discovery notes to repo setup in one session

- **Project Scoping** — feature request to SOW section + timeline

- **Project Handoff** — CODEBASE.md, deployment guide, client summary

- **End-of-Sprint Report** — commits to client-ready status update

AGENCY HOOK LIBRARY

- **client-branch-enforcer** — validates client code prefix in branch names

- **scope-creep-detector** — flags new requirements mid-session

- **client-commit-tagger** — auto-appends client code for invoicing audit trails

- **handoff-checklist** — verifies CODEBASE.md, .env.example, README on stop

CLIENT DELIVERABLES PROMPT PACK

- **Project Proposal Writer** — problem, solution, deliverables, timeline, investment

- **Statement of Work** — scope, acceptance criteria, payment milestones, IP ownership

- **Technical Specification** — architecture, API contracts, non-functional requirements

- **Post-Project Retrospective** — learnings, testimonial prompt, case study draft

- **Change Request Handler** — impact, effort delta, revised timeline, client approval format

Stack-agnostic: The CLAUDE.md template and hooks work with any client project stack.



PRODUCT 13

Indie Hacker / Solo Founder Toolkit

Idea to launch, solo. Ship fast, validate faster. Business metrics in Claude's context.

Solo founders have a unique constraint: one person doing the work of five. This toolkit is built around that reality — ship-fast conventions, Stripe revenue metrics directly in Claude's context, launch automation, and validation frameworks that tell you whether to keep building before you've wasted six months.

STRIPE MCP SERVER

- **Revenue dashboard** — MRR, ARR, new MRR, churn MRR, NRR
- **Customer queries** — plan, payment history, LTV
- **Churn analysis** — recently churned, cancellation reasons, cohort rates
- **Failed payment alerts** — dunning queue, retry status, recoverable revenue

VALIDATION & GROWTH PROMPT PACK

- **ICP Definer** — 3 ICP candidates with pain points and WTP
- **Pricing Page Designer** — tiers, anchoring, FAQ objections
- **User Interview Analyser** — raw transcripts to JTBD framework
- **SEO Content Brief** — keyword cluster to outline and meta copy
- **Churn Exit Interview** — real cancellation reasons + winback tree

LAUNCH WORKFLOW RECIPES

- **Landing Page Writer** — ICP to hero copy, features, CTA, meta tags
- **Product Hunt Launch** — tagline, description, maker story, upvote ask, checklist
- **Cold Outreach Campaign** — ICP to 10 personalised DMs + follow-up sequence
- **Pivot or Persevere** — usage + revenue + feedback → decision framework

SHIP-FAST CLAUDE.MD + HOOKS

- **Ship-fast CLAUDE.md** — speed-over-abstraction conventions
- **revenue-context hook** — prepends MRR to session context on start
- **feature-flag-enforcer** — encourages toggleable new features
- **launch-checklist hook** — analytics, error tracking, legal, Stripe live mode



PRODUCT 14

Mobile Developer Toolkit

Claude Code for iOS, Android, and cross-platform mobile. Every major stack covered.

Every other ClaudeKit toolkit assumes a web or backend developer. Mobile development has a completely different set of conventions — App Store submissions, provisioning profiles, simulator vs device behaviour, accessibility requirements, deep links, push notifications, and platform-specific performance traps. This toolkit teaches Claude all of it.

CLAUDE.MD TEMPLATES

- **Swift / SwiftUI** — MVVM/TCA, Swift concurrency, Xcode build rules, memory management
- **Kotlin / Jetpack Compose** — recomposition traps, Coroutines + Flow, ViewModel lifecycle, ProGuard
- **Flutter / Dart** — widget tree optimisation, Riverpod/Bloc, platform channels, hot reload limits
- **React Native (Expo)** — Expo Router, native bridging, OTA updates, Metro config

MOBILE SKILLS PACK

- **accessibility-auditor** — VoiceOver/TalkBack, contrast, touch targets, dynamic type
- **app-store-metadata-generator** — title, subtitle, description, keywords; character-count aware
- **deep-link-debugger** — Universal Links, App Links, missing entitlements
- **performance-profiler** — frame drops, Instruments analysis, FlatList optimisation

MOBILE WORKFLOW RECIPES

- **App Store Submission Prep** — screenshots, privacy manifest, entitlements, metadata, TestFlight
- **Crash Log Analysis** — symbolicated crash to root cause, fix, and regression test
- **New Feature (Mobile TDD)** — UI spec, widget test, implement, snapshot, accessibility check
- **Release Candidate** — version bump, build, beta, staged rollout plan, monitoring

MOBILE HOOK LIBRARY

- **privacy-manifest-checker** — validates PrivacyInfo.xcprivacy for new API usage
- **simulator-guard** — detects simulator-only code without


```
#if targetEnvironment
```
- **permission-usage-auditor** — verifies usage strings in Info.plist / AndroidManifest
- **bundle-size-monitor** — warns if binary size increases beyond 5% of baseline

REFERENCE

Quick Installation Reference

All install locations in one place. Everything is plain files — no build step except MCP servers.

Product	Install location	Scope	Setup time	Build step?
Dev Pro Skills Pack	~/.claude/skills/	Global	~5 min	No
Prompt Packs	Any folder / copy-paste	Any AI tool	<1 min	No
CLAUDE.md Templates	Project root	Per-project	<1 min	No
Hook Libraries	~/.claude/hooks/ (global) or project .claude/	Global / Per-project	~10 min	No
Workflow Recipes	Copy-paste prompt into Claude	Per-session	0 min	No
MCP Server Templates	Any directory + Claude settings	Global / project	~15 min	Yes — npm build
Starter Kits	New project folder	Per-project	~5 min	No
Specialist Toolkits (8-14)	Mix of above by component type	Varies	10-30 min	Only MCP servers

```
# Global skills install (example)
cp -r dev-pro-skills-pack/skills/ ~/.claude/skills/

# Global hooks install (Safety Guard example)
cp hooks/safety-guard/*.sh ~/.claude/hooks/
chmod +x ~/.claude/hooks/*.sh

# CLAUDE.md — drop in project root
cp nextjs-15-saas/CLAUDE.md ./CLAUDE.md

# MCP Server (PostgreSQL example)
cd mcp-postgres-server
cp .env.example .env # fill in DATABASE_URL
npm install && npm run build

# Add JSON block from README to ~/.claude/settings.json
```

FAQ

Frequently Asked Questions

What does Claude Code cost?

Claude Code requires at least a Pro subscription (\$20/month) or API credits. The free plan covers chat on web, iOS, Android, and desktop — the terminal-based Claude Code environment is not included on the free plan.

Does this work on Windows, Mac, and Linux?

Mac and Linux are fully supported. Windows is supported via WSL (Windows Subsystem for Linux) — Claude Code runs inside a Linux environment, and all hooks, skills, and recipes work normally inside WSL. Native Windows without WSL is not supported by Claude Code itself.

Can I use these across multiple projects?

Yes — and most of them are designed for it. Skills and hooks install globally to `~/.claude/` and activate on every project. CLAUDE.md templates and starter kits are per-project — copy them as many times as you want. One purchase, unlimited projects.

Do these work with Cursor, Windsurf, and other AI tools?

Skills and hooks are Claude Code-specific. CLAUDE.md templates work in any tool that reads project context files — Cursor reads CLAUDE.md automatically. Prompt packs work with any AI (ChatGPT, Gemini, etc.). Starter kits and MCP templates are AI-tool agnostic.

What do I get exactly — source files or executables?

Everything is plain source files. Hook scripts are bash (.sh). MCP templates are TypeScript source you build with npm. Starter kits are full project folders. Prompts and CLAUDE.md templates are Markdown. Nothing is minified, obfuscated, or binary. You can read and modify everything.

Can I modify the hooks and skills?

Yes — and you're encouraged to. Everything is plain source. Hooks and CLAUDE.md templates are starting points you tune to your stack and conventions. Most developers tweak at least a few settings in the first week, then leave them running indefinitely.

Will these be updated when Claude Code changes?

Yes. Purchases include all future updates. When Claude Code adds new features that improve these tools, updates are sent to buyers via Gumroad. You can re-download at any time from your Gumroad account.

What is the refund policy?

30-day no-questions-asked refund. If you install a product, try it on a real project, and it doesn't do what it says — email for a full refund.

Is there a team or agency licence?

The standard licence covers personal use and use within a single organisation — you can install it across all your own machines and projects. Agencies using it across client projects and teams sharing across multiple developers are covered too. For a formal multi-seat licence agreement, get in touch.

How do I receive the files after purchase?

Instantly, via Gumroad. After checkout you'll get a download link by email — a single zip containing all 14 toolkits, each in its own folder with an install guide. You'll also get a Gumroad account link so you can re-download anytime, including when updates are released.



ONE PURCHASE. EVERY PROJECT BENEFITS.

The Complete Claude Code Toolkit

All 14 toolkits. Every skill, hook, recipe, MCP server, CLAUDE.md template, and starter kit. Installs in 10 minutes. Works on every Claude Code session from now on.

\$149

30-day refund guarantee · Instant download · Updates included

WHAT'S INCLUDED

- Dev Pro Skills Pack (10 skills)
- Prompt Packs (25 prompts)
- CLAUDE.md Templates (6 stacks)
- Hook Libraries (17 hooks)
- Workflow Recipes (7 recipes)
- MCP Server Templates (5 servers)
- Starter Kits (5 kits)
- **All future updates included**
- AI/LLM App Developer Toolkit
- Security / AppSec Pro Toolkit
- Data & Analytics Toolkit
- Team & Engineering Leadership Toolkit
- Agency & Freelancer Toolkit
- Indie Hacker / Solo Founder Toolkit
- Mobile Developer Toolkit

claudekit.dev

Terminal Velocity AI · © 2026