TREN STUDIO · ENGINEERING HANDBOOK · 2026 EDITION

The Ultimate Guide to Free AI Models & Local LLM Tools

Plus 100 Battle-Tested Engineering Prompts for FORGE, Ollama, Cline, MCP Servers & Zero-Subscription AI Workflows

Published by: TREN Studio (trenstudio.com)

Audience: AI Engineers, Indie Developers & Local-First Builders

Edition: November 2026 · Complete Digital Edition

TABLE OF CONTENTS

What You Will Master in This Handbook

Cloud AI subscriptions add up fast. Between IDE assistants, multi-agent runs, and repeated debugging loops, a single active developer can easily burn $100 to $300 per month on frontier API tokens. This handbook gives you the exact hardware tables, routing architecture, and 100 structured prompts we use at TREN Studio to run fast, reliable local and free-tier models every day.

Chapter 1	The 2026 Local-First AI Stack: Cut API Bills by 70%+
Chapter 2	VRAM, Quantization & Hardware Blueprint (8GB to 128GB)
Chapter 3	Zero-Friction Setup: Ollama, Windows 11, WSL2 & FORGE Routing
Chapter 4	Best Free-Tier Cloud LLM APIs & Hybrid Fallback Architecture
Chapter 5	100 Battle-Tested Engineering Prompts (7 Specialized Categories)
Chapter 6	Production Safety, Secret Hygiene & Next Steps

> **How to Use This Book:** Start with Chapters 1–4 to lock in your local runtime (Ollama + FORGE) and prevent common Windows 11/VRAM crashes. Then bookmark Chapter 5—every prompt is engineered to produce structured, hallucination-resistant output even on compact 14B–32B quantized models.

CHAPTER 1

The 2026 Local-First AI Stack: Cut API Bills by 70%+

Most developer tasks do not require a $15-per-million-token frontier reasoning model. Boilerplate generation, unit test scaffolding, log parsing, commit summaries, and single-file refactors account for roughly **75% of daily engineering prompts**—and modern 14B to 32B open-weight models handle them locally at 40–90 tokens per second with zero per-token cost.

The 3-Tier Hybrid Routing Model

Tier	Workload Type	Recommended Engine	Cost / 1M Tokens
Tier 1 (Local)	Autocomplete, unit tests, regex, docs, single-file edits	Ollama + Qwen 2.5 Coder 14B / 32B	$0.00 (local power)
Tier 2 (Fast Cloud Free/Low)	Summarization, classification, fast multi-step tool checks	Groq / DeepSeek / Cloudflare Workers AI	$0.00 – $0.27
Tier 3 (Frontier Escalation)	Cross-repo architecture, subtle concurrency bugs	Claude / Gemini / OpenAI via FORGE fallback	$2.50 – $15.00

By routing Tier 1 and Tier 2 workloads away from paid frontier endpoints, indie developers routinely reduce monthly API invoices by 65%–80% while improving privacy because proprietary source code never leaves the local workstation.

CHAPTER 2

VRAM, Quantization & Hardware Blueprint (8GB to 128GB)

Running local LLMs smoothly is a **memory bandwidth and VRAM capacity** problem first, and a compute problem second. Once a model spills even 10% of its layers onto system DDR5 RAM over PCIe, generation speed drops by 5x–10x.

GPU / System Tier	VRAM / Unified	Sweet-Spot Model & Quant	Safe Context (num_ctx)	Typical Speed
RTX 3060 / 4060	8GB – 12GB	Qwen 2.5 Coder 7B (Q5_K_M)	8,192 – 16,384	55–85 tok/s
RTX 4060 Ti 16GB / 4080	16GB	Qwen 2.5 Coder 14B (Q4_K_M)	16,384 – 32,768	40–65 tok/s
RTX 3090 / 4090	24GB	Qwen 2.5 Coder 32B (Q4_K_M)	16,384 – 32,768	35–55 tok/s
RTX 5090	32GB GDDR7	32B (Q6_K) or 70B (IQ3/Q4)	32,768+	65–110 tok/s
Mac Studio M4 Max	64GB – 128GB	70B (Q4_K_M) via MLX	32,768+	18–32 tok/s

> **Golden Rule of Quantization:** For coding tasks, never drop below `Q4_K_M` unless forced by hardware limits. Going from `Q4_K_M` to `Q8_0` doubles VRAM usage for less than a 2% gain on HumanEval benchmarks, whereas dropping to `Q2_K` severely degrades JSON tool calling.

CHAPTER 3

Zero-Friction Setup: Ollama, Windows 11, WSL2 & FORGE

On Windows 11, three silent configuration traps derail 90% of local AI setups: IPv6 loopback binding (`[::1]:11434`), Modern Standby CUDA freezes, and WSL2 firewall isolation.

Production PowerShell Initialization

```
# 1. Force explicit IPv4 binding for IDE extensions & Python agents
[System.Environment]::SetEnvironmentVariable("OLLAMA_HOST", "127.0.0.1:11434", "User")

# 2. Enable Flash Attention & quantized KV-cache to save ~35% VRAM
[System.Environment]::SetEnvironmentVariable("OLLAMA_FLASH_ATTENTION", "1", "User")
[System.Environment]::SetEnvironmentVariable("OLLAMA_KV_CACHE_TYPE", "q8_0", "User")

# 3. Verify live API health
Invoke-RestMethod -Uri "http://127.0.0.1:11434/api/tags" -TimeoutSec 5
```

Pairing Ollama with FORGE (Open-Source Multi-Provider Router)

Install `forge-agent` on Python 3.10+ and point its primary local lane at `http://127.0.0.1:11434/v1` with automatic cloud fallback. When your laptop is unplugged or GPU memory is occupied, FORGE seamlessly fails over to your configured free-tier cloud provider without breaking your terminal session.

CHAPTER 4

Best Free-Tier Cloud LLM APIs & Hybrid Fallback Architecture

Even with a strong local GPU, keeping 2–3 free-tier cloud endpoints configured in your router gives you instant redundancy when rate limits hit or when you need ultra-fast inference on the road.

Provider	Best Free-Tier Strength	Primary Use Case	Caveat to Watch
Google AI Studio (Gemini)	Massive context window + multimodal	Whole-repo analysis & PDF parsing	RPM/RPD quotas on free tier
Groq Cloud	300+ tok/s LPU latency	Real-time classification & routing	Strict tokens-per-minute cap
Cloudflare Workers AI	Edge-native serverless + Flux images	Blog imagery & lightweight edge jobs	Daily neuron allowance
GitHub Models / OpenRouter Free	Unified OpenAI-compatible testing	Rapid cross-model benchmarking	Variable queue latency

Always wrap every provider call in an exponential backoff + fallback chain. Never hardcode a single API vendor into production scripts.

Category 1: System Architecture & Code Refactoring (Prompts 1–15)

Use these structured prompts with Qwen 2.5 Coder 14B, DeepSeek Coder, or FORGE multi-provider routing to refactor legacy codebases without hallucinated imports.

PROMPT #001 Zero-Regression Function Refactor

```
Analyze the function below. Return ONLY: (1) cyclomatic complexity hotspots, (2) hidden side effects, (3) a refactored version preserving exact input/output contracts, and (4) 3 edge-case unit tests.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #002 Dependency Inversion Blueprint

```
Inspect this tightly coupled module and propose a dependency-injection interface using pure standard-library types. Avoid adding third-party packages.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #003 Async Race Condition Audit

```
Trace every shared mutable state read/write in the async code below. List exact interleaving sequences that can cause data corruption and provide a minimal lock-free or asyncio.Lock fix.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #004 REST to Typed Contract Migration

```
Convert these raw JSON dictionary handlers into strict Pydantic v2 / TypeScript Zod schemas with explicit validation error codes.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 1: SYSTEM ARCHITECTURE & CODE REFACTORING (PROMPTS 1–15) (CONTINUED)

PROMPT #005 Monolith Module Decomposition

```
Split this 600-line file into 3 cohesive modules with unidirectional imports. Output a dependency diagram and exact file boundaries before writing code.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #006 SQL N+1 Query Eliminator

```
Examine the ORM loop below, identify every N+1 query pattern, and rewrite it using a single indexed JOIN or CTE with EXPLAIN ANALYZE expectations.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #007 Error Handling Standardization

```
Replace bare try/except or swallowed promises in this module with a typed error hierarchy separating retryable network errors from fatal validation errors.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #008 Idempotent Webhook Processor

```
Design an idempotent webhook handler using a deterministic idempotency key table so duplicate retries never double-charge or double-publish.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 1: SYSTEM ARCHITECTURE & CODE REFACTORING (PROMPTS 1–15) (CONTINUED)

PROMPT #009 Memory Leak Profiler Checklist

```
Review this long-running Node.js/Python worker for unbounded caches, unclosed file descriptors, or dangling event listeners.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #010 API Backward-Compatibility Guard

```
Compare v1 and v2 JSON response payloads below and list every breaking change that would crash existing mobile or CLI clients.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #011 Config & Secret Separation

```
Audit this repository snippet for hardcoded endpoints, credentials, or environment assumptions and generate a typed `.env.example` + loader.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #012 Deterministic State Machine

```
Rewrite these nested boolean flags (`isLoading`, `isRetrying`, `hasError`) as an explicit finite state machine with exhaustive transitions.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 1: SYSTEM ARCHITECTURE & CODE REFACTORING (PROMPTS 1–15) (CONTINUED)

PROMPT #013 Pagination Cursor Conversion

```
Replace OFFSET/LIMIT pagination with keyset cursor pagination on `(created_at, id)` to keep query latency constant at 10M+ rows.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #014 Retry with Exponential Backoff & Jitter

```
Implement a reusable retry wrapper honoring HTTP `Retry-After` headers, full jitter, and a maximum total deadline.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #015 Minimal Diff Code Review

```
Act as a staff engineer. Review this diff strictly for correctness, security, and performance regressions—skip stylistic nitpicks.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

Category 2: React, TypeScript & Frontend Performance (Prompts 16–30)

Optimized for 14B–32B local models running inside Cline, Roo-Code, or VS Code with tight context windows.

PROMPT #016 **Re-Render Bottleneck Diagnosis**

```
Identify every prop reference or inline object/callback causing unnecessary React child re-renders
in this component tree.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #017 **Accessible Modal & Focus Trap**

```
Build a zero-dependency accessible React dialog component with ESC dismissal, focus trapping, and
ARIA attributes.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #018 **Strict TypeScript Discriminated Unions**

```
Replace optional boolean props in this React component with discriminated union props so invalid
UI states cannot compile.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #019 **Core Web Vitals LCP Optimizer**

```
Audit this HTML/React hero section for Largest Contentful Paint blockers (unoptimized images,
render-blocking fonts, client-side fetch waterfalls).
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 2: REACT, TYPESCRIPT & FRONTEND PERFORMANCE (PROMPTS 16–30) (CONTINUED)

PROMPT #020 Optimistic UI Rollback Pattern

```
Implement an optimistic list mutation hook that updates UI immediately and cleanly rolls back state if the server returns 4xx/5xx.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #021 Tailwind Dark-Mode Token System

```
Convert these ad-hoc hex colors into a cohesive dark-mode CSS variable palette with WCAG AA contrast ratios (>=4.5:1).
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #022 Debounced Search with AbortController

```
Write a React search hook that debounces input by 250ms and cancels stale in-flight `fetch` requests via `AbortController`.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #023 Virtual List Threshold Estimator

```
Evaluate whether this 2,000-row table needs windowing/virtualization and provide a lightweight implementation.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 2: REACT, TYPESCRIPT & FRONTEND PERFORMANCE (PROMPTS 16–30) (CONTINUED)

PROMPT #024 Form Validation Without Bloat

```
Create a typed multi-step form validator with inline field errors, dirty-state tracking, and keyboard navigation.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #025 SSR Hydration Mismatch Finder

```
Scan this Next.js/React component for browser-only APIs (`window`, `Date.now()`, `localStorage`) that trigger hydration warnings.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #026 Responsive CSS Grid Bento Layout

```
Write a pure CSS Grid Bento layout that adapts cleanly across 375px mobile, 768px tablet, and 1440px ultrawide screens.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #027 Zero-CLS Image Component

```
Refactor these `<img>` tags to guarantee zero Cumulative Layout Shift using explicit aspect ratios, `loading='lazy'`, and WebP fallbacks.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 2: REACT, TYPESCRIPT & FRONTEND PERFORMANCE (PROMPTS 16–30) (CONTINUED)

PROMPT #028 Keyboard Shortcut Manager

```
Implement a scoped keyboard shortcut hook (`Cmd/Ctrl+K`) that safely ignores keypresses while typing inside `<input>` or `<textarea>`.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #029 Client-Side CSV Exporter

```
Write a browser utility that streams an array of typed objects to a properly escaped RFC 4180 CSV download.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #030 Bundle Size Tree-Shaking Audit

```
Inspect these package imports and replace heavy barrel imports with tree-shakeable ESM equivalents.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

Category 3: Python, FastAPI & Backend Engineering (Prompts 31–45)

Designed for production Python 3.10+ APIs, CLI tools, and FORGE agent orchestration.

PROMPT #031 **FastAPI Lifespan Resource Manager**

```
Write a FastAPI `lifespan` context manager that initializes an async HTTPX pool and
SQLite/Postgres pool cleanly on startup and closes them on shutdown.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #032 **Structured JSON Logging Setup**

```
Configure Python `logging` to output single-line JSON logs with request correlation IDs while
automatically redacting Authorization headers.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #033 **Token-Bucket Rate Limiter**

```
Implement a thread-safe in-memory and Redis-backed token-bucket rate limiter per client IP and API
key.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #034 **Streaming SSE Endpoint**

```
Build a FastAPI Server-Sent Events endpoint that streams LLM tokens with proper disconnect
detection and heartbeat comments.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 3: PYTHON, FASTAPI & BACKEND ENGINEERING (PROMPTS 31–45) (CONTINUED)

PROMPT #035 Subprocess Timeout & Sandboxing

```
Wrap this Python `subprocess.run` call with strict timeouts, argument-list safety (`shell=False`), and stdout/stderr byte caps.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #036 SQLite WAL Concurrency Tuning

```
Generate optimal SQLite PRAGMA settings (`journal_mode=WAL`, `busy_timeout`, `synchronous=NORMAL`) for high-read web workloads.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #037 Pytest Parametrized Edge Suite

```
Generate a `pytest.mark.parametrize` test suite covering empty inputs, Unicode edge cases, boundary integers, and malformed JSON.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #038 Multi-Provider LLM Fallback Router

```
Design a Python router class that tries local Ollama first, falls back to Groq on timeout, and escalates to a frontier API on schema failure.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 3: PYTHON, FASTAPI & BACKEND ENGINEERING (PROMPTS 31–45) (CONTINUED)

PROMPT #039 Pydantic Structured Output Repair

```
Write a validation loop that feeds exact Pydantic `ValidationError` messages back to a local LLM for at most 2 self-correction retries.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #040 CLI UX with Argparse & Exit Codes

```
Refactor this script into a POSIX-compliant CLI with `--dry-run`, `--json`, clear stderr errors, and documented exit codes.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #041 Async Batch Concurrency Limiter

```
Use `asyncio.Semaphore` to process 500 API URLs with max concurrency 10, per-item timeout, and partial-failure reporting.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #042 ZIP64 Cross-Platform Archiver

```
Write a Python backup utility using `zipfile` with `allowZip64=True`, symlink/junction exclusion, and post-write `testzip()` verification.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 3: PYTHON, FASTAPI & BACKEND ENGINEERING (PROMPTS 31–45) (CONTINUED)

PROMPT #043 Safe File Atomic Write

```
Implement atomic file writes (write to temp file in same directory + `os.replace`) to prevent corrupted JSON state files on crash.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #044 Caching Decorator with TTL

```
Build a lightweight LRU + TTL cache decorator for synchronous and async Python functions without external dependencies.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #045 OpenAPI Client Schema Validator

```
Validate outbound HTTP payloads against an OpenAPI 3.1 schema in CI before deploying breaking changes.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

Category 4: Local LLMs, Ollama, VRAM & Quantization (Prompts 46–60)

Practical system prompts and troubleshooting workflows for running 7B–70B models locally without OOM crashes.

PROMPT #046 VRAM Budget Calculator

```
Calculate exact VRAM requirements for a given parameter count, quantization (Q4_K_M / Q8_0), and `num_ctx` KV-cache size.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #047 Ollama Modelfile Optimizer

```
Generate an Ollama `Modelfile` tuned for deterministic code generation (`temperature 0.1`, `top_p 0.9`, `num_ctx 16384`, explicit stop tokens).
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #048 Windows 11 IPv6 & WSL2 Diagnostics

```
Produce a 5-step PowerShell diagnostic script checking `netstat :11434`, `OLLAMA_HOST`, Windows Firewall profiles, and WSL2 gateway reachability.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #049 Schema-Constrained JSON Grammar

```
Convert this TypeScript interface into a strict JSON Schema / GBNF grammar so a 14B local model never emits invalid JSON.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 4: LOCAL LLMS, OLLAMA, VRAM & QUANTIZATION (PROMPTS 46–60) (CONTINUED)

PROMPT #050 RAG Chunking Strategy Designer

```
Recommend optimal chunk size, overlap, and hybrid BM25 + dense vector retrieval weights for technical API documentation.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #051 CPU Embedding Offload Architecture

```
Design a local RAG pipeline that runs `nomic-embed-text` on CPU while reserving 100% of GPU VRAM for the 14B–32B chat model.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #052 Prompt Compression for 8K Context

```
Compress this 4,000-token system prompt down to under 900 tokens without losing any hard constraints or output formatting rules.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #053 Local Model Benchmark Harness

```
Write a Python script measuring time-to-first-token (TTFT), tokens/sec, and JSON-parse pass rate across 3 local Ollama models.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 4: LOCAL LLMS, OLLAMA, VRAM & QUANTIZATION (PROMPTS 46–60) (CONTINUED)

PROMPT #054 Tool-Calling Loop Circuit Breaker

```
Implement a middleware guard that detects repeated identical tool calls from local models and injects a corrective stop hint.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #055 Multi-GPU Tensor Split Config

```
Explain optimal layer offloading and tensor-split ratios for asymmetric dual-GPU rigs (e.g., RTX 4090 24GB + RTX 4060 Ti 16GB).
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #056 Context Rot Mitigation

```
Design a rolling conversation summarizer that compacts old turns while preserving pinned file paths and error traces verbatim.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #057 Offline Air-Gapped Model Mirror

```
Document how to export, verify SHA-256 checksums, and transfer Ollama blobs to an offline engineering workstation.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 4: LOCAL LLMS, OLLAMA, VRAM & QUANTIZATION (PROMPTS 46–60) (CONTINUED)

PROMPT #058 Fine-Tuning Dataset Sanitizer

```
Write a Python filter that removes PII, duplicate instructions, and low-entropy samples from a JSONL LoRA training dataset.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #059 QLoRA Hyperparameter Baseline

```
Provide a conservative QLoRA config (`r=16`, `lora_alpha=32`, learning rate, warmup) for adapting a 14B model on 24GB VRAM.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #060 Local vs Cloud Cost Break-Even

```
Build a break-even table comparing electricity + GPU depreciation against OpenAI/Anthropic token spend at 10M, 50M, and 200M tokens/month.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

Category 5: MCP Servers, Agents & Workflow Automation (Prompts 61–75)

Build reliable Model Context Protocol (MCP) tools and autonomous pipelines with strict safety guardrails.

PROMPT #061 FastMCP Tool Schema Designer

```
Design 4 read-only MCP tools with strict Pydantic input schemas, concise docstrings, and bounded output sizes.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #062 MCP Stdio Timeout Wrapper

```
Wrap every external network call inside an MCP server with an `asyncio.wait_for` timeout so clients never hang silently.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #063 Windows Absolute Path Config Generator

```
Generate a valid `claude_desktop_config.json` / IDE MCP config using absolute Python venv paths to eliminate `spawn ENOENT`.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #064 Read-Only Database MCP Guard

```
Enforce SQL read-only execution (`PRAGMA query_only=ON` / read-only Postgres role) inside a database-querying MCP tool.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 5: MCP SERVERS, AGENTS & WORKFLOW AUTOMATION (PROMPTS 61–75) (CONTINUED)

PROMPT #065 Jaccard Duplicate Content Guard

```
Implement a shingle-based Jaccard similarity checker to block duplicate blog posts or social media captions above 0.72 similarity.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #066 RSS-to-Social Pipeline Architect

```
Design an idempotent RSS poller with local hash state, `--dry-run` verification, and clean error alerting.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #067 Human-in-the-Loop Approval Gate

```
Add a staging/draft approval boundary to an automated publishing agent so zero unreviewed content reaches production.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #068 Agent Tool-Call Telemetry Logger

```
Log tool latency, token count, and failure codes to SQLite while stripping any strings matching common API key prefixes.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 5: MCP SERVERS, AGENTS & WORKFLOW AUTOMATION (PROMPTS 61–75) (CONTINUED)

PROMPT #069 Cloudflare Worker Cron Watchdog

```
Create a Cloudflare Worker cron handler with D1 heartbeat logging and dead-man's-switch alerting on silent cron failures.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #070 Playwright Headless QA Smoke Test

```
Write a Playwright script that verifies HTTP 200, zero console errors, and visible CTA buttons across 5 landing pages.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #071 Multi-Agent Task Partitioner

```
Break a large codebase documentation audit into disjoint directory subtrees so parallel agents never edit the same file.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #072 Prompt Injection Defense Checklist

```
Audit this agent workflow that ingests untrusted web HTML and add strict data/instruction separation.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 5: MCP SERVERS, AGENTS & WORKFLOW AUTOMATION (PROMPTS 61–75) (CONTINUED)

PROMPT #073 GitHub Release Notes Automator

```
Generate clean user-facing release notes and social announcements from conventional commit messages between two Git tags.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #074 Automated Backup Integrity Verifier

```
Write a post-backup verification step that tests archive readability and logs file counts + byte sizes.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #075 Zero-Downtime Static + WP Hybrid Routing

```
Review an Apache `.htaccess` file routing SPA pages alongside WordPress posts without breaking `/wp-json/` or `/sitemap.xml`.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

Category 6: Security, DevOps & Secret Hygiene (Prompts 76–88)

Protect your API keys, Git history, and production servers before scaling traffic.

PROMPT #076 Git History Secret Scanner

```
Write a local Python scanner checking `git log --all -p` for `ghp_`, `github_pat_`, `sk-`, `AIza`, `EAA`, and private key blocks with masked output.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #077 Windows NTFS Vault ACL Hardening

```
Provide safe `icacls` commands restricting a local secrets folder exclusively to the current user, SYSTEM, and Administrators.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #078 Pre-Commit Secret Blocker

```
Create a lightweight `.git/hooks/pre-commit` script rejecting staged `.env` files or high-entropy token patterns.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #079 Least-Privilege Cloudflare API Token

```
List the minimal scoped permissions needed for separate Workers AI, D1, and DNS tokens instead of using a Global API Key.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 6: SECURITY, DEVOPS & SECRET HYGIENE (PROMPTS 76–88) (CONTINUED)

PROMPT #080 SSH Key Rotation Runbook

```
Generate a zero-lockout SSH key rotation checklist: deploy new public key, test `BatchMode=yes`, then revoke the old key.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #081 WordPress Hardening Baseline

```
Audit a WordPress installation for exposed debug logs, unused plugins, sample pages, and missing security headers.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #082 Docker Non-Root Hardening

```
Refactor this Dockerfile to use a multi-stage build, pinned base image digest, and non-root runtime user.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #083 CSP & Security Headers Config

```
Generate clean `X-Content-Type-Options`, `Referrer-Policy`, and AdSense-compatible Content-Security-Policy headers.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 6: SECURITY, DEVOPS & SECRET HYGIENE (PROMPTS 76–88) (CONTINUED)

PROMPT #084 Encrypted Vault Backup Strategy

```
Design a 3-2-1 backup plan using encrypted `.7z` archives and offline recovery keys without leaking plaintext metadata.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #085 Dependency Vulnerability Triage

```
Triage a `pip-audit` or `npm audit` report to separate reachable production CVEs from dev-only false positives.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #086 Webhook HMAC Signature Verifier

```
Implement constant-time `hmac.compare_digest` verification for Stripe, Gumroad, and GitHub webhooks.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #087 Rate-Limited Nginx / Cloudflare Rules

```
Draft Cloudflare WAF rate-limiting rules protecting `/wp-login.php` and `/api/` endpoints from credential stuffing.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #088 Incident Postmortem Template

```
Generate a blameless technical postmortem covering root cause, blast radius, verification
commands, and permanent prevention.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

Category 7: High-RPM Tech SEO, E-E-A-T & Monetization (Prompts 89–100)

Turn real engineering work into compounding US organic traffic, Amazon Associates revenue, and digital product sales.

PROMPT #089 Ultra Long-Tail US Keyword Miner

```
Given a core developer pain point, generate 15 ultra-specific troubleshooting queries with low competition and high US intent.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #090 AI-Overview Extractable Lead Writer

```
Rewrite this introduction so the first 50 words directly answer the query with concrete versions, commands, and zero fluff.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #091 Hands-On Benchmark Table Builder

```
Format raw local GPU/API timing logs into a clean Markdown/HTML comparison table highlighting price-to-performance.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #092 Buyer-Intent Hardware Review Outline

```
Create a 1,500-word review outline featuring real specs, pros/cons, who should skip it, and compliant Amazon affiliate placement.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 7: HIGH-RPM TECH SEO, E-E-A-T & MONETIZATION (PROMPTS 89–100) (CONTINUED)

PROMPT #093 FTC & AdSense Disclosure Generator

```
Write a clear, concise affiliate and advertising disclosure block compliant with FTC guidelines, Amazon Associates, and Google AdSense.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #094 TechArticle + FAQPage JSON-LD Generator

```
Generate valid Schema.org `@graph` JSON-LD combining `TechArticle` and `FAQPage` for a developer troubleshooting guide.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #095 Pinterest 1000x1500 Pin Copywriter

```
Distill a 1,500-word technical guide into a punchy 7-word headline and 3 scannable checklist bullets for a vertical Pinterest pin.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #096 Non-Spammy Reddit Troubleshooting Reply

```
Draft a 90%-value technical answer solving a developer's exact error in `r/LocalLLaMA` with an optional reference link.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CATEGORY 7: HIGH-RPM TECH SEO, E-E-A-T & MONETIZATION (PROMPTS 89–100) (CONTINUED)

PROMPT #097 Digital Product Landing Page Copy

```
Write conversion-focused, hype-free landing page copy for a technical PDF handbook aimed at senior and indie developers.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #098 Internal Topic Cluster Linker

```
Map 5 contextual internal anchor links connecting troubleshooting guides, hardware reviews, and hub pages without keyword stuffing.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #099 Core Web Vitals Blog Audit

```
Diagnose why a WordPress tech blog scores below 80 on mobile PageSpeed and list exact LiteSpeed Cache + image fixes.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

PROMPT #100 Weekly Revenue & Traffic KPI Review

```
Summarize weekly US sessions, GSC impressions, top-converting affiliate clicks, and next week's highest-ROI content priorities.
```

Recommended Target: Qwen 2.5 Coder 14B/32B · Temp 0.1–0.2 · Output: Strict Markdown/Code

CHAPTER 6 · PRODUCTION SAFETY & NEXT STEPS

Secret Hygiene, Backups & Scaling Your AI Stack

As your local and hybrid workflows grow, enforce three non-negotiable engineering habits:

1. **Never commit secrets or `.env` files:** Store API tokens in an encrypted local vault outside your project repositories and run automated Git history scans before every public push.
2. **Quarantine before deleting:** Always back up databases and workspace directories before running automated refactors or cleanup scripts.
3. **Benchmark before upgrading hardware:** Measure your actual tokens-per-second and KV-cache footprint before spending $2,000+ on a flagship GPU.

> **Explore More from TREN Studio:**
> • Open-Source FORGE Router: **https://github.com/TREN-Studio/FORGE**
> • Curated AI Hardware & Robotics Vault: **https://www.trenstudio.com/tech-vault/**
> • Deep Engineering Blog: **https://www.trenstudio.com/blog/**