AI INTEGRATION SAMPLES

Real AI features inside real workflows, not prompt-only demos

Selected real-world AI integration patterns from production work and public prototypes.

Public-safe and local case patterns showing how I wire LLMs into apps, bots, and internal tools with structured outputs, guardrails, and maintainable delivery. Best fit: teams that already have a product, admin flow, support process, or document-heavy workflow and want one useful AI feature that behaves predictably.

WHAT I BUILD

Engineering-first AI integration

- Structured extraction, scoring, and drafting flows
- Deterministic filtering and routing before LLM calls
- Slack, Telegram, Sheets, APIs, and background-job integrations
- Human-in-the-loop workflows where AI is used only where it adds value
- Timeouts, retries, dedupe, limits, and logging around the model layer

DELIVERY STANDARD

How I keep AI features from becoming expensive chaos

Schema-first
JSON-only or constrained output so downstream logic stays stable.

Rule before model
Deterministic filtering or routing before spending tokens.

Measured behavior
Caps, retries, fallbacks, and low-confidence handling.

FAST USE CASES

MVP lanes that sell well on Upwork

- Document-grounded assistant or internal Q&A tool
- Form triage and intake normalization
- Summarize -> classify -> draft -> approval workflow
- Support tagging, reply drafts, or AI-assisted ops routing

TECHNICAL DNA

The patterns visible in the case studies

Python · C# / .NET · Apps Script · SQLite · Slack · Telegram · OpenAI-style APIs · xAI / Grok · Prompt safety · Retries / backoff · Cost caps · Human approval

> Core positioning: rules first, AI second. The model is a controlled component inside an engineered system, not the whole system.

What I can deliver quickly

A focused spike, one production-minded MVP feature, or a hardened workflow with logging, fallbacks, and handoff notes.

CASE STUDIES

Two stronger AI implementation patterns

These are not "chatbot" examples. They show LLMs embedded inside a broader workflow with validation, routing, and operator-friendly delivery.

CASE 01

Public-safe repo

AIJobSearcher: structured lead triage with model sanity checks

Public GitHub proof available. The real value is the combination of deterministic filtering, controlled LLM scoring, and delivery into a reviewable workflow.

BUILT

- Multi-source collection into SQLite with normalization and dedupe
- LLM scoring used only after deterministic filtering
- GPT + Grok style "council" checks for final ranking
- Telegram/feed export for practical review and distribution

WHY IT WORKS

- JSON-only conservative outputs
- Timeout-aware calls and fallback path
- Prompt-injection defenses and host restrictions
- "Answer only if confident" behavior

CORE STACK

Python, SQLite, OpenAI-compatible APIs, Telegram

SALES ANGLE

Repeatable extraction, ranking, and controlled AI use instead of free-form prompt output.

CASE 02

Local app build

Mobile app localization pipeline with dual-model review

Useful for apps with catalog content, CMS entries, product text, help content, or multilingual assets that need more than raw translation.

BUILT

- Chunked translation flow for structured content
- Second-pass review with xAI / Grok
- Fallback review through OpenAI if reviewer output breaks
- Write-back into app resource files

WHY IT WORKS

- Strict JSON schema for every batch
- ID and tag preservation checks
- Controlled chunk sizing for token and failure control
- Final catalog verification before publish

CORE STACK

Python, OpenAI, xAI / Grok, JSON validation, mobile app resources

SALES ANGLE

AI embedded into a real shipping content pipeline, not isolated as a toy translation script.

INTEGRATION PATTERNS

Internal tools, delivery discipline, and adjacent automation

Most client projects are hybrids: one LLM feature, plus the boring engineering needed to make it usable inside a real business process.

CASE 03

GPT + Google Sheets / Docs internal tooling pattern

Older public prototypes showing a practical pattern: read settings, fetch source content, call GPT, then write results back into the operator's tool.

- Apps Script version reading prompt and model settings from a spreadsheet
- .NET version reading Google Docs content, sending it to GPT, and writing back into Sheets
- Fits admin tools, review queues, CRM helpers, and lightweight internal copilots

ADJACENT PROOF

Ops-grade automation around APIs and notifications

- Slack bot workflow with daily cost cap, retry/backoff, and SQLite dedupe
- Slash commands and operator actions instead of opaque background behavior
- Proof that reliability matters as much as the AI layer

Slack | Webhooks | Retry / backoff | Budget cap | Operator controls

WHAT CLIENTS USUALLY BUY

High-conversion offer formats

- **AI integration spike:** prove one workflow fast and define guardrails
- **AI feature MVP:** one integrated workflow with config, logging, and test scenarios
- **Hardening pass:** timeouts, retries, cost caps, and safer operator handoff

CASE 04

QA workflow AI assistant (production prototype)

Built for a real internal QA workflow.

- Converts raw QA notes into structured Jira ticket drafts via Jira API
- Human review before final submission
- Telegram alerts shorten the "found -> ticketed -> seen" cycle

WHY IT WORKS

- Structured ticket schema validation
- Human approval before submission
- AI used only to structure raw notes

CORE STACK

C# / .NET, Jira API, Telegram bot, OpenAI-style LLM APIs

SALES ANGLE

AI embedded inside a real engineering workflow while keeping human validation.

STRONGEST POSITIONING

How to position this service on Upwork

- Do not lead with "agents" or hype language
- Lead with workflow integration, structured outputs, and cost control
- Make RAG one option, not the whole identity, unless the client clearly needs it
- Show AI embedded into an existing process, not standing alone as a novelty

Best CTA for the listing

Send your stack, one target workflow, and 3 to 5 real input/output examples. I will map the fastest path to a working AI feature MVP and show where AI should and should not be used.