PromptTriage Case Study

RAG-powered prompt engineering platform with benchmarked prompt strategy, retrieval, model-selection, and format-efficiency evidence.

28K+ Prompt corpus	1,080 Study E evaluations	+13.9% Best gain vs generic	86.7% Fewer prompt words

Executive Summary

PromptTriage turns rough user intent into production-grade prompts across text, image, video, and system-prompt workflows. The platform combines a Next.js interface, Supabase Auth, Stripe subscription logic, FastAPI retrieval, Pinecone vectors, modality-specific metaprompts, and Azure deployment history. Its strongest signal is the research trail: benchmark outputs, judge-bias corrections, model/runtime tradeoffs, and prompt-format experiments.

The results are intentionally framed as task-dependent. PromptTriage showed the strongest gains where prompts needed explicit structure, exact constraints, edge-case handling, boundary setting, or resistance to default model behavior.

Benchmark Impact

Metric	Scope	Result	Impact
PromptTriage vs generic prompt	Study D IFEval, Gemini, 50 tasks	82% vs 72%	+13.9% relative
PromptTriage vs no system prompt	Study C v2, Nemotron 120B, 13 tasks	30.92 vs 27.54	+12.3% relative
Naive RAG vs judge RAG	Study A, 180 outputs, /50 rubric	42.27 vs 38.97	+8.5% quality, 57.3% faster
Naive RAG vs agentic RAG	Study A RAG ablation	42.27 vs 39.40	+7.3% quality, 42.5% faster
Short prompts vs long prompts	Study E v2, 1,080 evals	32.03 vs 26.76	+19.7% quality, 86.7% fewer words
Qwen3 14B vs tested MoE path	Study B prompt-generation benchmark	41.77 vs 41.47	Slightly higher score, about 64x faster

Edge Cases With The Largest Gains

Edge case	Best observed win	Why it mattered
Anti-default UI/code generation	Nemotron: 35 vs 5 bare (+600%)	Helped avoid AI UI defaults such as dark gradients, rounded corners, and framework assumptions.
Strict constrained formatting	Nemotron: 25 vs 7 simple (+257%)	Exact line/syllable constraints punished generic prompting.
Timezone scheduling reasoning	Nemotron: 30 vs 13 simple (+131%)	Constraint ordering and assumption control mattered.
Boolean logic trap	Gemini: 31 vs 18 simple (+72%)	Systematic hypothesis testing beat intuitive guessing.
Security code review	Gemini: 48 vs 43 simple (+11.6%)	Improved role expertise, boundaries, edge cases, specificity, and format.
DevOps incident response	Claude: 45 vs 41 bare (+9.8%)	Improved edge cases, specificity, and operational formatting.
Medical/financial boundary handling	Up to +3 / 50 (+7.0% to +7.3%)	Better boundary setting where confident generic advice can be risky.

Architecture And Implementation

- Frontend: Next.js interface with modality selection, prompt analysis, and refinement flows.
- Auth and billing: Supabase Auth plus Stripe checkout, billing portal, and webhook-backed subscription state.
- Retrieval: FastAPI RAG service with Pinecone prompt vectors and corpus-derived examples.
- Generation: model-provider calls wrapped by modality-specific metaprompts.
- Deployment history: GitHub Actions, Azure Container Registry, Azure Container Apps, and Google Cloud Run history.

Format And Token-Order Lessons

Study E supports compact, structured prompts over long instruction walls. Short prompts scored 32.03 versus 26.76 for long prompts while using 86.7% fewer words on average. For multimodal and agentic systems, the practical design rule is to place the most important details first: task, role, constraints, input priority, output format, and safety boundaries should be ordered from highest to lowest importance because models react differently depending on token order.

Case Study And Repository Format Status

- The public case study follows the required professional sequence: summary, problem, constraints, requirements, architecture, security, deployment, operations, cost, results, tradeoffs, lessons, next steps, links, interview explanation, and resume bullets.
- The PromptTriage repository has a README, deployment documentation, security policy, research notes, benchmark output JSON, Azure ML logs, release charts, and architecture evidence.
- Remaining gaps are evidence polish: public-safe screenshots, job-level cost mapping, runbooks, monitoring/alert screenshots, and downloadable benchmark CSV/JSON summaries.

Resume-Ready Value Claims

- Built and benchmarked PromptTriage across prompt strategy, RAG complexity, prompt format, model selection, and judge-bias studies.
- Demonstrated up to +13.9% relative instruction-following improvement versus generic prompts and up to +12.3% downstream quality lift versus no system prompt.
- Identified a compact-prompt strategy that scored +19.7% higher than long prompts while using 86.7% fewer words.
- Selected a Qwen3 14B production candidate that was about 64x faster than the tested MoE inference path while maintaining comparable quality.

Verification Sources

Primary evidence lives in the PromptTriage repository: backend/research/RESEARCH_PROGRESS.md, backend/research/notebooks/named-outputs, Study A/B/C/D/E benchmark JSON, release charts, README, deployment notes, and the public case study page.