internStudio

GENERATIVE AI INTERNSHIP

TASK GUIDE

Build with large language models. Prompting, retrieval, agents and real GenAI apps.

SIX ADVANCED PROJECTS — CHOOSE ANY TWO

01 **Retrieval-Augmented Knowledge Assistant**

02 **AI Content Generation & Brand Voice Studio**

03 **Multi-Agent Research & Report Assistant**

04 **AI-Powered Code Review & Documentation Assistant**

05 **Text-to-Image Prompt Studio & Gallery**

06 **Custom Chatbot Builder with Persona Management**

INTERNSHIP
Generative AI

REQUIRED TASKS
Complete any 2 projects

FOCUS
LLMs · Prompt Engineering · RAG · Agents · Evaluation

INTERNSHIP INSTRUCTIONS

Your Internship Tasks

Select and complete **any 2 projects** from the task list. Each should be a real application built around a generative model — not a single one-off prompt in a notebook.

Design	Prompt	Build	Evaluate	Guard	Deploy

EACH PROJECT SHOULD DEMONSTRATE

- [ ] A well-scoped use case a generative model genuinely improves
- [ ] Thoughtful prompt design, not just a raw pass-through
- [ ] Handling of hallucination or low-quality output
- [ ] Grounding in real data where the use case needs it
- [ ] Guardrails against misuse or unsafe output
- [ ] Cost/latency awareness (caching, model choice)
- [ ] A working interface to use the application
- [ ] A real evaluation of output quality

EVERY SUBMISSION INCLUDES

- [ ] GitHub repository
- [ ] Source code
- [ ] README explaining prompts and architecture
- [ ] Example inputs and outputs
- [ ] Evaluation notes (what worked, what didn't)
- [ ] Requirements / dependency file
- [ ] Demo link or screenshots

> **Suggested technologies:** Python · an LLM API (OpenAI, Anthropic, or open-source via Ollama) · a vector database (Chroma, FAISS, pgvector). Optional: LangChain/LlamaIndex, Streamlit.

Submit your completed work here

Once both projects are evaluated, documented and pushed to GitHub, submit your project details through the official submission form.

https://forms.gle/vxL5JE8cvRXXfDBv8

> **A note on quality.** Calling an LLM API is the easy part. The work is in prompt design, grounding, guardrails and knowing when the model is wrong.

01 Retrieval-Augmented Knowledge Assistant

Build a RAG assistant that answers questions grounded in a specific document set, with citations.

REQUIREMENTS

- Chunk and embed a real document collection
- Generate answers grounded in retrieved content only
- Detect and handle out-of-scope questions
- Measure retrieval quality separately from generation quality
- Retrieve relevant chunks per query
- Cite the specific source for each claim
- Evaluate answer accuracy against a test question set
- Provide a simple chat interface

Focus: Retrieval-augmented generation, grounding and citation, and separated retrieval/generation evaluation.

02 AI Content Generation & Brand Voice Studio

Build a tool that generates on-brand marketing copy across formats, adapting to a defined brand voice.

REQUIREMENTS

- Define and encode a brand voice / style guide as prompt context
- Allow tone and length controls
- Version and compare multiple generations
- Evaluate on-brand consistency across outputs
- Generate copy across multiple formats (social, email, blurb)
- Detect and flag off-brand or inconsistent output
- Support human editing before finalizing
- Export finished copy in usable formats

Focus: Prompt engineering for style control, generation variety, and consistency evaluation.

03 Multi-Agent Research & Report Assistant

Build a multi-agent system where specialized agents (researcher, summarizer, writer) collaborate to produce a report.

REQUIREMENTS

- Define distinct agent roles with clear responsibilities
- Ground the researcher agent in real retrieved sources
- Produce a structured final report, not raw agent chatter
- Evaluate the final report against a manual baseline
- Coordinate agents in a defined workflow
- Handle disagreement or low-confidence findings
- Log the intermediate steps for auditability
- Handle failure of one agent without crashing the pipeline

Focus: Multi-agent orchestration, role design and end-to-end pipeline evaluation.

04 AI-Powered Code Review & Documentation Assistant

Build a tool that reviews code diffs for issues and generates accurate documentation from source code.

REQUIREMENTS

- Parse a real diff or codebase as input
- Generate accurate docstrings or README sections
- Rank or prioritize flagged issues by severity
- Support multiple languages or be explicit about scope
- Flag genuine issues (bugs, style, missing tests), not noise
- Avoid hallucinating behavior the code doesn't have
- Evaluate flagged issues against known real issues
- Integrate into a usable workflow (CLI or CI-style)

Focus: Code-aware prompting, hallucination control and evaluation against real code issues.

05 Text-to-Image Prompt Studio & Gallery

Build a studio for crafting, versioning and curating prompts against an image-generation API.

REQUIREMENTS

- Integrate with an image-generation API
- Version prompts and compare resulting images
- Support iterative refinement of a prompt
- Handle API failures and rate limits gracefully
- Support prompt templates and reusable style presets
- Organize generated images into a searchable gallery
- Track cost/usage per generation
- Allow tagging and filtering of saved images

Focus: Prompt iteration workflow, API integration and asset management for generative output.

06 Custom Chatbot Builder with Persona Management

Build a tool that lets a non-technical user configure a domain-specific chatbot — persona, knowledge, guardrails — without writing prompts by hand.

REQUIREMENTS

- A configuration UI/flow for defining a persona and scope
- Enforce guardrails (topics to avoid, escalation triggers)
- Support multiple independently configured bots
- Evaluate whether the bot stays within its configured scope
- Attach domain knowledge the bot can be grounded in
- Preview and test the configured bot before publishing
- Log conversations for review
- Provide an embeddable or shareable chat interface

Focus: Configuration-driven prompt generation, guardrail enforcement and scope-adherence evaluation.

FINAL STEP

Ready to Submit?

Verify everything below before you send your work in.

SUBMISSION

- [ ] GitHub repository
- [ ] README explaining the approach
- [ ] Example inputs and outputs
- [ ] Requirements / dependency file
- [ ] Submission form completed
- [ ] LinkedIn post where required

EVALUATION

- [ ] Output quality actually evaluated
- [ ] Failure cases identified
- [ ] Cost / latency considered

DESIGN

- [ ] Use case is well-scoped
- [ ] Prompts are deliberately designed and iterated
- [ ] Grounded in real data where needed

SAFETY

- [ ] Guardrails against misuse in place
- [ ] Hallucination / low-quality output handled
- [ ] Sensitive data handled responsibly

Important. Complete any 2 projects. You should be able to explain your prompt design choices and show real examples of where your system succeeds and where it fails.

Submission Form

Submit your repository, live link and screenshots here.

https://forms.gle/vxL5JE8cvRXXfDBv8

internStudio

LEARN · BUILD · GROW

www.internstudio.in

team.internstudio@gmail.com