Project 1: Domain Q&A Chatbot

Due: Feb 23

Groups: Up to 2 students

Submit on CourseWorks: GitHub repo link + live deployed URL

Goal

Build and deploy a chatbot that answers questions in a narrow domain, and prove it works using an automated eval harness.

What you must demonstrate

Prompting

- Few-shot prompting (≥3 examples)
- Clear scope using **positive constraints** (no "don't do X")
- A way to handle uncertainty (escape hatch)

Evaluation

- At least **1 deterministic metric**
- At least **10 golden-reference MaaJ** evals (judge compares to an expected answer)
- At least **10 rubric MaaJ** evals (judge grades against a rubric)

Requirements

1) App

- Pick a niche topic narrow enough to write **20+** test questions.
- Backend: **FastAPI**
- Frontend: simple web UI
- Deploy on **GCP** and provide a live URL

2) Prompt

Your promptng strategy must include: - Role/persona (domain voice + boundaries) - Few-shot examples (≥3) - You may statically fix or dynamically load your examples - Organizaton method of your choice - Positive constraints (define what it *can* answer) - Escape hatch (what to do when unsure)

3) Out-of-scope handling

- Define **3+** out-of-scope categories using positive framing
- Add a **Python backstop** after generation (keyword/regex/simple classifier) to catch misses
    - A great example of this would be safety handling, detect distressed keywords and fallback into a different prompt.

4) Evaluation harness

Create: - A **golden dataset** with **20+** cases: - 10 in-domain (with expected answer) - 5 out-of-scope (with expected refusal behavior) - 5 adversarial/safety-trigger - A runnable **eval script** that: - runs all tests - reports pass/fail per test - prints pass rates by category - includes: - ≥1 deterministic metric (regex/keywords/refusal detection)

Repo contents

- `README.md` (topic, how to run locally, link to live URL)
- `pyproject.toml` (it must be `uv`-based)
- app code (FastAPI + frontend)
- eval (single command to run specified in `README.md`)