Agentic Tagging

Goal

From 40 labeled descriptions, build a local prototype that:

1. Tags each proposal with a small, well-reasoned taxonomy, and
2. Decides whether to publish tags or place the proposal on hold when evidence is thin.

Your final deliverable will be a GitHub repository containing your code, outputs, and documentation.

We prefer that you use LangGraph, but other agentic frameworks are acceptable if you briefly articulate your choice.

Inputs

You'll receive a CSV with the following columns: proposalId, description.
Each row is one proposal.

Task

1. Compact taxonomy derived from the data
    - Propose a concise set of categories needed to describe these projects.
    - Include short definitions.
2. Tagging pipeline
    - Build an agentic workflow that, given the input CSV, does the following:
        - Ingests proposals from the CSV
        - Proposes tags for each proposal grounded in your taxonomy
        - Produces, for each proposal:
            - Suggested tags
            - Some notion of confidence
            - Short evidence excerpts from the description that support the tags
        - Enforces your taxonomy
        - Makes a publish vs hold decision per proposal
        - Encodes your thresholds in a way that's easy to inspect
    - You are free to design the graph and node responsibilities as you wish. We're interested in how you structure the workflow and decisions.

Outputs

Your GitHub repo should include

1. Tagged results artifact
    - A machine-readable file (CSV or JSON), one record per proposal.

   - Must include: proposal ID, final tags, publish/hold, and a brief indication of *why* (e.g., confidence, evidence snippet, or short rationale).
   - It should be easy to scan and see how your system behaved across proposals.
2. Taxonomy definition
   - A clear list of categories with short definitions.
   - The mapping between these categories and the tags in your outputs should be obvious.
   - Can be a small data file, module, or clearly documented section of code.
3. Agentic workflow implementation
   - Code that defines and runs your workflow.
   - A brief README explaining how to run it and outlining your thresholds and known limitations.

Guidance

- Derive the taxonomy from the sample data; don't overfit or over-engineer.
- Prefer fewer, clearer tags over noisy complete coverage.
- Aim for 2–3 hours of effort.
- Depth over breadth:
  - Tight taxonomy
  - Clean, inspectable decisions
  - Crisp, human-readable outputs
- Submission
  - Push your solution to a GitHub repository.
  - Share the repository URL with us in your reply to this exercise.
  - If the repo is private, grant access to @ktorttila and @ajmiti.
  - Do not commit any API keys or other secrets; document how to configure them in the README.