Task 1: Prompt Reliability – Quick Mini-Harness

What to Ship First:

For the first step, the main goal should be getting the **basic test harness up and running**. This means setting up the evaluation tests with a mix of **synthetic** and **real-world** edge cases. We’re focusing on making sure the model is consistently returning reliable answers to paraphrased queries, queries with typos, and noisy input. Initially, we'll use **exact match** or **regex-based matching** as our primary evaluation metrics.

The **consistency score** is important here. For each test case, we’ll calculate how consistent the model's responses are across the different queries. This way, we can measure how well the system performs overall.

I would also ship a **basic reporting system** that shows which queries pass or fail and gives an overall consistency score for each test case. This helps us visualize which parts of the system are working well and which need improvement.

What to Ship Later:

Once we have the basic system working, there are a few things that could improve the tool in the longer term. For example, we could add more advanced metrics for evaluating **semantic similarity**, like using **BLEU score** or **cosine similarity**. These would give us deeper insights into how well the model is understanding the intent behind queries, not just whether it matches exactly.

I'd also recommend **automating** the test process and building in features to monitor performance over time. A **dashboard** for continuous testing would be really useful in a production environment. As the system scales, we can optimize the testing process and handle a larger set of queries without manual intervention.

Lastly, as we collect more feedback, we could look at improving the model’s handling of edge cases. Some queries may fail more often than others, and we should prioritize refining those areas.