**MATH AGENT**

**Math Agent: System Overview and Implementation Report**

**1. Input & Output Guardrails Used for Privacy**

**Approach Taken:**  
To maintain privacy and ensure content safety, input and output guardrails were implemented in the AIGateway class. The guardrails work by:

* **Input Filtering:** Using regular expressions to detect and block unsafe content such as sensitive personal data (e.g., passwords, credit card information, SSNs), and inappropriate or harmful queries (e.g., hacking-related, pornographic).
* **Domain Validation:** Ensuring the user’s question is math-related using keyword pattern matching. If a query does not match a predefined set of math terms and patterns, it is rejected.
* **Output Filtering:** Similar patterns are checked in the response. If potentially unsafe or irrelevant output is detected (e.g., unauthorized URLs, unsafe content), the response is blocked.

**Why This Approach:**  
This ensures responsible AI usage by preventing harmful or irrelevant content from being processed or returned to the user. It enforces scope limitation to math-related topics and avoids security or ethical violations.

**2. Knowledge Base**

**Dataset Used:**  
A custom JSON dataset named math\_dataset.json containing simple math queries and their respective solutions was used to populate the Qdrant vector database. Sentence embeddings are generated using the all-MiniLM-L6-v2 model from Sentence Transformers.

**Sample Questions:**

1. **What is 2 + 2?**  
   *Answer: The answer is 4.*
2. **Solve for x: 2x + 5 = 11**  
   *Answer: Subtract 5 from both sides: 2x = 6. Divide by 2: x = 3.*
3. **What is the value of x: x - 3 = 7**  
   *Answer: Add 3 to both sides: x = 10.*

**Details:**  
Each question-answer pair is vectorized and stored with Qdrant using cosine similarity for semantic search. The dataset is minimal but can be extended.

**3. Web Search Capabilities**

**Strategy Used:**

* Integrated the Tavily API to perform real-time web searches.
* The system queries external APIs using POST requests and retrieves JSON responses.
* If a question isn't matched in the knowledge base, the system attempts a web search.
* Extracted answers are filtered to prevent unsafe URL content unless labeled appropriately.

**Sample Questions:**

1. **What is the quadratic formula?**  
   *(Not in the dataset; fetched via web search)*
2. **What is Euler's formula in complex numbers?**  
   *(Uses Tavily to retrieve up-to-date academic explanations)*
3. **What is the latest mathematical breakthrough in number theory?**  
   *(Real-time search needed to retrieve up-to-date responses)*

**4. Human-in-the-Loop (HITL) Routing in Agentic Workflow**

**Workflow Description:**  
The Math Agent architecture includes a human-in-the-loop mechanism via the FeedbackSystem class:

* After receiving the response, users are prompted to submit textual feedback.
* The system collects both manual (user) and automated (DSPy) feedback.
* Responses and feedback are stored for post-processing and continuous improvement.

**Routing Components:**

* **Router Class:** Decides the execution path.
  + First checks the KnowledgeBase.
  + Then tries WebSearch.
  + Finally, falls back to MathProcessor to evaluate raw mathematical expressions.
* **Human Feedback:** Serves as a loop to monitor quality and potentially retrain or adjust system behavior.

**Purpose:**  
This method ensures that even if the system provides suboptimal answers, there is a mechanism to learn and adapt, helping developers improve agent performance and reliability.

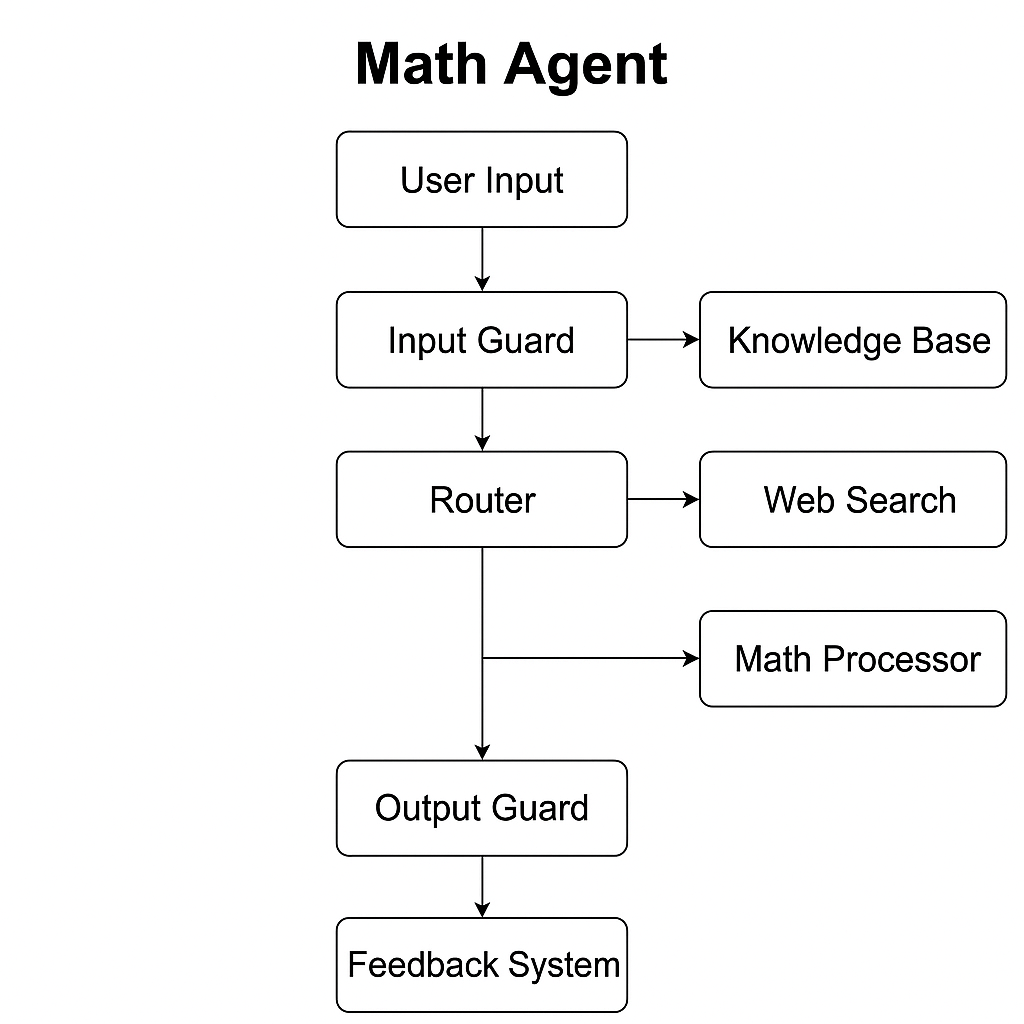
**Source Code Submission**

The complete source code for the Math Agent project, including all modules, configuration files, and datasets, is available at the following GitHub repository:

**GitHub Repository:**  
<https://github.com/ANILKUMARDUBALA/math-agent>

This repository contains:

* The main Streamlit app and all supporting Python modules (AIGateway, KnowledgeBase, WebSearch, FeedbackSystem, MathProcessor, Router).
* The sample math question dataset (math\_dataset.json).
* Configuration files and instructions for setting up the environment and dependencies.



**STREAMLIT INTERFACE**

