Assignment 1: Building an Agentic AI System

Date

02/03/26 – 02/17/26

Total Points

100

Assignment Template

Link:
https://colab.research.google.com/drive/1JHuD2VhX2CJv8aR0sKiQyI8PoCxb2Syn?authuser=1#scrollTo=WjyHer3DkQ50

---

1. Introduction

In this assignment, students will build a modern agentic AI system that integrates:

- Large Language Models (LLMs)
- Retrieval-Augmented Generation (RAG)
- Deterministic computational tools
- A router-based decision mechanism

The goal of this assignment is to understand how state-of-the-art AI systems are composed and orchestrated in practice.

---

2. Assignment Structure

The assignment consists of five main sections, implemented in a single Jupyter Notebook.

Section 1: Basic LLM Chat

Students implement a minimal LLM-based chatbot using an open-source instruction-tuned model.
This section introduces message formatting, chat templates, and text generation.

Section 2: Conversation Memory

Students extend the chatbot to support multi-turn conversations by explicitly maintaining and updating conversation history.

Section 3: Retrieval-Augmented Generation (RAG)

Students implement a RAG pipeline that:

- Loads a document corpus
- Creates vector embeddings
- Retrieves relevant documents
- Generates answers grounded in retrieved evidence

Retrieved evidence must be displayed and cited in the final answers.

Section 4: Tool-Based Climate Data Analysis

In this section, students use pre-implemented tools to analyze climate data stored in NetCDF files.
These tools perform deterministic numerical computations (e.g., averages, sums, unit conversions).
No TODOs are required in this section.

Section 5: Complete Agentic System

Students build a router-based agent that:

- Analyzes user queries
- Decides which tools or RAG components to invoke
- Executes the selected actions
- Synthesizes a final answer

The router produces structured JSON plans, ensuring stability, predictability, and grading consistency.

---

3. Provided Materials

Students are provided with:

- A starter Jupyter Notebook containing code structure and TODOs
- A climate documentation corpus (JSONL format)
- Climate datasets in NetCDF format

All required datasets and starter code are included.

---

4. Student Tasks

Students are required to:

- **Complete all TODOs in Sections 1, 2, 3, and 5. (14 in totals)**
- Ensure the notebook runs end-to-end without errors
- Display RAG retrieval evidence when answering documentation-based questions
- Use tools (not the LLM) for numerical computations
- Produce correct and grounded final answers

Students must not hard-code answers.
All information must come from RAG or tool outputs.

---

5. Submission Instructions

Students should submit:

1. Only one completed Jupyter Notebook
2. Make sure all code cells executed successfully and output results retained for key cells

The notebook should be submitted through the course submission system.