

East West University Robotics Club

A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka-1212, Bangladesh

Email: roboticsclub@ewubd.edu FB: <https://www.facebook.com/ewuRoboticsClub/>

Document Name: LLM-Based AI Engine Development Question

Time: 3 Hours 30 Minutes

Date: 24th January, 2026

LLM-Based AI Engine Development: Innovation Challenge

Problem Statement

East West University maintains comprehensive policy documents covering academic rules, student conduct, and administrative procedures. These documents are extensive and not directly queryable by AI systems.

Your task is to design and implement an **Agentic Retrieval-Augmented Generation (RAG) system** that can answer policy-related queries **accurately and strictly based on the provided documents**. The system must operate as a **multi-step agentic workflow** with the following components:

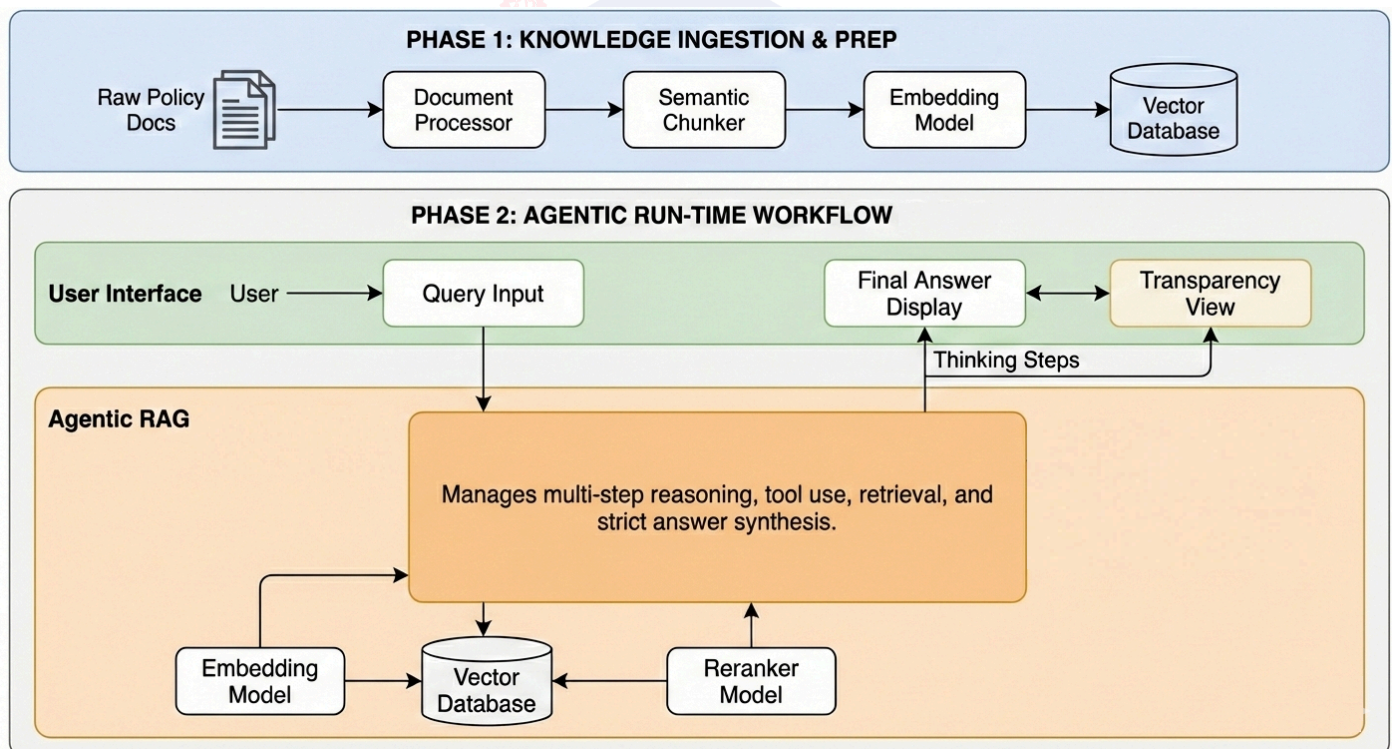
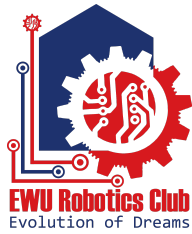


Figure : System Workflow

1. Document Processing & Knowledge Preparation
2. Retrieval-Augmented Generation (RAG)
3. Agentic Workflow & Multi-Step Reasoning
4. Backend Tool Integration
5. User Interface & Transparency

Dataset Link: [EWU Policy Documents – Google Drive](#)



East West University Robotics Club

A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka-1212, Bangladesh

Email: roboticsclub@ewubd.edu FB: <https://www.facebook.com/ewuRoboticsClub/>

Document Name: LLM-Based AI Engine Development Question

Problem 1: Document Processing and RAG Knowledge Base

Objective

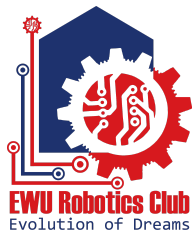
Build a pipeline that converts the raw EWU PDF policy documents into a **structured knowledge base** suitable for retrieval and reasoning.

Requirements

- Document Conversion:** Convert the three EWU policy PDFs into a structured textual format (clean text or JSON) that preserves:
 - Section boundaries
 - Headings
 - Subsections and content
- Semantic Chunking:** Split the documents into **semantically meaningful chunks**, with configurable chunk size and overlap, suitable for downstream embedding.
- Vector Embeddings & Storage:**
 - Generate vector embeddings for all chunks using a **sentence-level embedding model**.
 - Store embeddings in a **persistent vector database**, which will act as the **sole source of document knowledge** for all queries.
- RAG Query Module:** Implement a retrieval-augmented generation module that:
 - Retrieves the most relevant chunks for a given query.
 - Generates answers strictly **grounded in retrieved content**.
 - Explicitly refuses to answer if sufficient information is not found, returning: *"I don't know based on the provided documents."*
 - Avoids hallucination or reliance on external knowledge.

Evaluation Notes

- Accuracy of retrieval and grounding is critical.
- Participants must use local or limited-access models **due to API restrictions**; external models with unrestricted access are discouraged.



East West University Robotics Club

A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka-1212, Bangladesh
Email: roboticsclub@ewubd.edu FB: <https://www.facebook.com/ewuRoboticsClub/>

Document Name: LLM-Based AI Engine Development Question

Problem 2: Agent Design, Multi-Step Reasoning & Backend Tools

Objective

Implement the system as a **LangGraph-based agentic workflow** that demonstrates **multi-step reasoning and tool integration**, not a simple script.

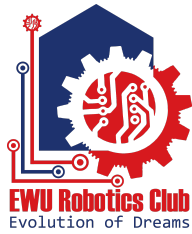
Requirements

1. **Agentic Workflow:**
 - Design a **graph of interconnected nodes** with shared state and conditional transitions.
 - Include at minimum:
 - **Planning/Router Node:** decides whether retrieval, tool usage, or both are required.
 - **Retrieval Node:** queries the vector database.
 - **Tool Execution Node:** interacts with backend services via HTTP APIs.
 - **Response Node:** generates final answer or refusal.
 - Agents must reuse **intermediate results** across steps.
2. **Backend Tool Server:**
 - Develop at least one **HTTP-accessible backend tool** (e.g., calculator, structured analysis, text statistics).
 - Queries requiring computation or structured analysis must use this API.
 - **Direct local function calls are prohibited.**
3. **Multi-Step Queries:**
 - Demonstrate queries that require **dependent steps**, e.g., retrieve content → extract numbers → call tool → generate final explanation.
4. **UI & “Thinking” Visualization:**
 - Build a **simple UI** (Gradio, Streamlit, or similar) for users to submit queries.
 - The **LangGraph agent’s reasoning steps** (“thinking”) must be **visualized in the UI**.
 - Judges will evaluate the clarity, correctness, and transparency of this reasoning chain.

Testing Queries

Test your system with this queries:

1. What actions are considered student misconduct under the EWU Disciplinary Code?
2. What are “unfair or illegal means” in examinations according to the Code?
3. What punishments can be awarded if a student is found guilty of misconduct?
4. Who are the members of the Disciplinary Committee and what is its authority?
5. What is the procedure after a complaint is lodged against a student?
6. Can a student be suspended before the disciplinary hearing? If yes, under what conditions?
7. What rights does an accused student have during a disciplinary hearing?
8. How does the university handle cases related to sexual harassment?
9. Can a student appeal a disciplinary decision, and what is the appeal process?
10. What immediate powers does an invigilator have if unfair means are detected during an examination?



East West University Robotics Club

A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka-1212, Bangladesh

Email: roboticsclub@ewubd.edu FB: <https://www.facebook.com/ewuRoboticsClub/>

Document Name: LLM-Based AI Engine Development Question

Submission Format

Generated answers must be stored in a **Python list** in the following format:

```
model_answers = [  
    "Answer generated by your model for question 1",  
    "Answer generated by your model for question 2",  
    "Answer generated by your model for question 3",  
    "Answer generated by your model for question 4",  
    "Answer generated by your model for question 5",  
    "Answer generated by your model for question 6",  
    "Answer generated by your model for question 7",  
    "Answer generated by your model for question 8",  
    "Answer generated by your model for question 9",  
    "Answer generated by your model for question 10"  
]
```

- Update **Google Sheet** ([📄 AI_ENGINE_DEVELOPMENT_OUTPUTS](#)) with a link to a **Docs file** containing your model outputs.

Constraints

- All answers must originate **solely from the provided EWU policy documents**.
- **LangGraph agent reasoning must be visible** in the UI.
- All tool interactions must occur over **HTTP APIs**.
- Internet access is allowed for boilerplate code, but answers must be reproducible and document-grounded.

Evaluation Criteria

1. Correctness and completeness of RAG retrieval.
2. Clarity and logic of the agentic workflow.
3. Effective multi-step reasoning and tool integration.
4. Transparency of intermediate reasoning steps (“thinking”). *This will add bonus marking.*
5. Proper backend API design and interaction.
6. Robustness and reproducibility of results.
7. Avoidance of hallucinations and external assumptions.