

## Task: Build an Agentic Certificate Evaluation AI (Chat-Driven)

### Objective

Design and implement a fully agentic, chat-driven Certificate Evaluation AI Agent that reasons dynamically based on user interaction and evolving context.

The solution must not be implemented as a predefined workflow or pipeline. Instead, the agent must decide its next step at runtime, driven by conversation and internal state.

For this task, any free LLM may be used (e.g., Gemini free tier, open-source models, or equivalent). The evaluation focus is agentic step selection and context handling, not model accuracy.

### Core Responsibilities

- Design a single intelligent agent (not multiple chained agents)
- Use chat as the sole control plane
- Ensure the agent dynamically selects its next action
- Allow skipping, revisiting, or reordering steps
- Avoid hard-coded sequences
- Maintain persistent state across turns, including:
  - Certificate content and extracted data
  - Evaluation criteria and weights
  - Intermediate reasoning and decisions
  - User instructions, corrections, and overrides

### Agentic Step Selection (Primary Focus)

The agent must continuously reason about:

“What is the most appropriate next action based on current context?”

Possible actions include:

- Extract missing or ambiguous information
- Validate selected criteria only
- Re-score after criteria changes
- Explain or justify prior decisions
- Compare certificates
- Ask for clarification
- Pause execution awaiting confirmation

### Context & State Management

The agent must manage three evolving state spaces:

1. Certificate State: content, extracted fields, inferred facts, confidence levels
2. Evaluation State: criteria, rules, weights, partial and final scores
3. Conversation State: user intent, follow-ups, challenges, scope changes

### Explainability & Reasoning

The agent must:

- Explain why a specific step was chosen
- Justify decisions clearly
- Highlight missing or uncertain evidence
- Revise conclusions when challenged
- Treat previous outputs as living context

### Technical Constraints

- LLM: Any free LLM (Gemini free tier, open-source, etc.)
- Frameworks: LangChain + LangGraph (state & reasoning only)
- No external tools or integrations
- No fixed workflows or pipelines

### Test Data Requirement

- Use your own university certificate as test data
- Demonstrate:
  - Context ingestion
  - Incremental reasoning across turns
  - Partial re-evaluation without resetting state
  - Explainable outcomes via chat

### Deliverables

- Source code
- Short README explaining:
  - Agent state model
  - Step-selection logic
  - Context handling strategy
- Sample chat transcript showing:
  - User intervention
  - Step reordering
  - Re-evaluation after criteria change

### Expected Outcome

A minimal but correct agentic system where:

- The agent chooses actions dynamically
- The user controls evaluation through conversation
- State evolves continuously without restarting flows

This task evaluates agentic thinking and context management, not UI, integrations, or LLM sophistication.