



CANDIDATE ASSESSMENT

Jetlink.

2025

TASK

Version 1.3

PREPARED BY
Jetlink Team

TASK

Implementing a Conversational AI Chatbot with Memory

Objective:

The goal of this case study is to develop a Conversational AI Chatbot that implements a memory system similar to OpenAI's memory feature. The chatbot should have:

1. Short-term memory (context persistence within a session).
2. Long-term memory (persistent user details across sessions).
3. A simple UI to interact with the chatbot.
4. Python-based backend for memory storage and retrieval

1. System Overview

The chatbot should function as follows:

- A user can chat with the bot in a UI.
- Within a session, the bot retains short-term context.
- Once a session ends, short-term memory is reset.
- If relevant, the bot retrieves long-term memory to personalize responses.
- Users can delete memory when requested.

2. Technical Constraints

- The backend must be implemented in Python.
- The candidate is free to choose the rest of the tech stack (database, frontend framework, hosting, etc.).

3. Expected Features & Deliverables

Features:

- Short-term memory within a session.
- Long-term memory stored persistently across sessions.
- Ability to delete long-term memory.
- A simple UI for interaction.

Deliverables:

- Fully functional chatbot system.
- Well-documented Python code.
- Instructions for running locally.

4. Evaluation Criteria

The candidate's solution will be evaluated based on:

- **Correctness:** Does the implementation correctly handle short-term and long-term memory?
- **Code Quality:** Is the code well-structured and documented?
- **Efficiency:** Is memory retrieval optimized?
- **UI Usability:** Does the chatbot UI work smoothly?

5. Bonus Points

- Implementing vector search for efficient memory retrieval.
- Providing an option in the UI to choose between cloud-based LLMs (e.g., OpenAI, Anthropic) and a small local LLM model for response generation.
- Deploying the chatbot using Docker + Cloud (AWS/GCP/Azure).

Final Notes

This case study is meant to assess both AI engineering skills and software development abilities. The focus should be on writing clean, scalable Python code and ensuring the chatbot effectively handles memory like OpenAI's system.