Tanit Multimodal Fertility Assistant

Multimodal Gen AI Intern (VLM / LLM / Voice) – Home Assignment

Author: YOUR NAME HERE

Date: December 2025

Tanit.ai

1 Introduction

This report presents a prototype of the **Tanit Multimodal Fertility Assistant**, built for the "Multimodal Gen AI Intern (VLM / LLM / Voice)" assignment.

The assistant is designed as a **single-page chatbot** that:

- accepts text chat,
- records and transcribes voice input,
- allows image and PDF upload (hormone panels, cycle charts, ultrasound screenshots, lab reports),
- and grounds answers on a fertility knowledge layer (RAG-style design).

The goal is to provide *educational explanations* about fertility (AMH, ovarian reserve, PCOS, age), not to replace medical decisions.

Medical disclaimer. The assistant is strictly educational. It does not provide diagnosis, does not prescribe treatments, and must not replace consultation with a qualified healthcare professional.

2 User Experience and UI

The application runs as a **Gradio web app** with a simple layout:

- **Left:** chat conversation between user and assistant.
- **Right:** input panel with:
  - a text box,
  - a microphone recorder,
  - image upload (PNG/JPEG),
  - PDF upload,
  - *Send* and *Clear conversation* buttons.

A custom `style.css` applies Tanit.ai branding: logo in the header, purple/teal color palette and card-like panels with soft shadows.

Every session starts with a visible medical disclaimer, and all responses keep an empathetic, cautious tone ("many patients with PCOS...", "guidelines typically consider age, ultrasound and hormones together").

3 Architecture Overview

Figure 1 summarises the end-to-end pipeline.

The main modules are:

- `app.py`: Gradio UI + `chat_pipeline()` orchestration.
- `voice/stt.py`: speech-to-text wrapper around `faster-whisper`.
- `vlm.py`: stubs for image and PDF analysis (VLM-ready).
- `rag.py`: keyword-based RAG stub returning short fertility explanations.
- `assets/style.css`: branding and layout.

High-level architecture:

User Inputs (Text / Voice / Image / PDF)
↓
Pre-processing (STT, basic image/PDF handling)
↓
Context Building (RAG stub + multimodal summaries)
↓
Answer Generation (templated assistant logic)
↓
Gradio Chat UI

Figure 1: High-level architecture of the prototype.

Processing Steps

1. **Inputs:** text, voice, image and/or PDF are collected from the UI.
2. **Pre-processing:**
   - voice is transcribed,
   - image/PDF pass through stub functions that simulate a short medical summary.
3. **Context building:**
   - text + transcription are concatenated into one user message,
   - multimodal summaries are concatenated into a context string,
   - `rag.py` returns a topic-aware paragraph (AMH, PCOS, age, ovarian reserve).
4. **Answer generation:** a helper function combines:
   - medical disclaimer,
   - RAG context,
   - information about which inputs were used (voice, image, PDF).
5. **Output:** updated conversation is displayed in the chat.

4 Models, Quantization and Latency

Even though the submission uses stubs for heavy models, the design is explicitly aligned with the assignment requirements.

Speech-to-Text (faster-whisper)

- **Chosen library:** `faster-whisper`.
- **Typical model:** `medium` (configurable).
- **Quantization:** FP16 or int8 via `ctranslate2`, good fit for CPU/GPU.
- **Expected latency:**
  - short questions ($\sim$ 5–10s speech): 1 s on GPU, around 1–2 s on CPU.

LLM for Reasoning (Target)

In this prototype, final answers are generated by a templated function to keep things lightweight. The architecture, however, assumes a small instruction-tuned LLM:

- **Candidates:** Qwen3-4B-Instruct (primary), DeepSeek-R1-Distill-Qwen-7B, Llama-3.1-8B-Instruct.
- **Quantization strategy:** 4-bit GGUF (e.g. Q4 variants) for GPU- or CPU-friendly deployment.
- **Latency target:** about 1.5–2.5 s for a 128–256-token answer, with overall end-to-end latency (STT + RAG + LLM) kept under **4 seconds**.

Vision-Language Model (Target)

The `vlm.py` file defines clear entrypoints for future VLM integration:

- **Primary target:** Qwen3-VL-4B-Instruct for document and table understanding.
- **Alternative:** Phi-4-Vision-Instruct for fast table/chart reasoning.
- **Quantization:** 4-bit quantization to allow single-GPU deployment.
- **Latency target:** ~ 1 s per image/page, with caching for multi-page PDFs.

RAG / GraphRAG

Currently, `rag.py` is a **keyword-based stub**. For a production version:

- 100–200 fertility PDFs (guidelines, educational documents) would be ingested.
- A **GraphRAG** index or **FAISS** + **LlamaIndex** vector store would be built.
- Retrieval + re-ranking should stay under ~ 500 ms per query.

5 Safety Guardrails

Given the sensitivity of fertility topics, several guardrails are included from the beginning:

- **Medical disclaimers in every answer:** the assistant states it is educational only and does not replace a doctor.
- **Non-prescriptive language:** no explicit treatment instructions, drug names, or dosages; answers use cautious wording and population-level statements.
- **No diagnosis or triage decisions:** the system never decides if a user "needs IVF" or "will not conceive"; instead it encourages medical consultation.
- **Transparency about stubs:** where VLM or GraphRAG are still simulated, this is clearly indicated to avoid over-trust.
- **Handling incomplete inputs:** if only an image/PDF is sent without text, the assistant returns a generic explanation and recommends showing the report to a doctor.

6 Example Interaction

User (voice)

> *"I'm 34, my AMH is 1.1 ng/mL and I have PCOS. Should I be worried?"*

Assistant (text)

> *I am an educational fertility assistant and I do not replace a doctor. For any medical decision or treatment, please consult a fertility specialist.*
>
> *Based on your description, an AMH of 1.1 ng/mL at age 34 is slightly below average but still compatible with natural conception for many people, including those with PCOS. AMH mainly reflects ovarian reserve, not egg quality. In PCOS, it is common to see many small follicles on ultrasound, and AMH values can be difficult to interpret in isolation.*
>
> *Current recommendations suggest looking at the full picture: your age, ultrasound findings, hormone levels over time and your clinical history. A fertility specialist can put all of this together and discuss your options.*

This illustrates the intended behaviour: warm tone, topic-aware explanation, explicit uncertainty and a clear recommendation to see a specialist.

7 Conclusion

The submitted prototype implements a **multimodal chatbot** aligned with the assignment requirements:

- single-page Gradio UI with Tanit branding,
- support for text, voice, image and PDF inputs,
- clear architecture separating UI, STT, VLM stubs and RAG stub,
- defined model choices, quantization strategies and latency targets,
- strong safety guardrails.

While VLM and GraphRAG are currently stubbed, the code structure makes it straightforward to plug in:

- a quantized Qwen3-VL-4B-Instruct for document understanding,
- a small instruction-tuned LLM (e.g. Qwen3-4B-Instruct) for reasoning,
- and a GraphRAG or FAISS-based fertility knowledge index.

This prototype is therefore both a working demo and a solid starting point for a future patient-facing fertility companion at Tanit.ai.