MedGemma: Conversational Medical AI

**Version:** 1.0 (Stable Terminal Application)

**Date:** June 8, 2025

**Author:** Prashant Saxena

**AI Model:** MedGemma-4b-it model (Google)

# 1. Project Overview

This document outlines the architecture, setup, and usage of the **MedGemma Conversational AI**, a powerful command-line tool designed for multimodal medical analysis. The application leverages Google's state-of-the-art MedGemma-4b-it model to provide expert-level analysis of both text-based medical queries and medical images.

The application operates in two distinct modes:

* **Conversational Q&A:** Users can ask general medical questions and receive detailed explanations from the AI.
* **Interactive Image Analysis:** Users can trigger an analysis of a local folder of medical images, selecting from a comprehensive list of 25 medical specialties to guide the AI's perspective.

This tool is built for stability, robustness, and a clean user experience, running entirely within a local terminal environment to ensure privacy and full control over system resources.

# 2. System Requirements & Dependencies

To run this application, the following are required:

## 2.1. Hardware

* **GPU:** An NVIDIA GPU with **CUDA support** is mandatory. A minimum of **8 GB of VRAM** is strongly recommended for stable performance.
* **RAM:** 16 GB of system RAM or more.
* **Storage:** Approximately 15-20 GB of free disk space for the model and Python environment.

## 2.2. Software & Libraries

This project uses a specific, stable set of Python libraries. It is **essential** to use a Python virtual environment to prevent conflicts with other projects.

* **Python:** Version 3.10.x
* **Core Libraries:**
  + torch & torchvision (CUDA-enabled version)
  + transformers & accelerate (from Hugging Face)
  + pillow (for image processing)
  + pygame (for audio cues)
  + pywin32 (for the file dialog)
  + tqdm (for progress bars)

# 3. Installation and Setup Guide

Follow these steps precisely to create a clean, working environment.

**Step 1: Create a Python Virtual Environment**

This is the most critical step for ensuring stability.

1. Open a new **PowerShell** or **Command Prompt** window.
2. Navigate to your desired project directory (e.g., C:\Projects\).
3. Create the project folder and enter it:

mkdir MedGemma\_AI

cd MedGemma\_AI

1. Create the virtual environment. This will create a .venv folder inside your project.

# For PowerShell

.venv\Scripts\Activate.ps1

Your terminal prompt should now start with (.venv).

**Step 2: Install PyTorch with CUDA**

It is essential to install the version of PyTorch that matches your system's CUDA drivers.

1. **Uninstall any existing versions** to prevent conflicts:

pip uninstall torch torchvision torchaudio

1. **Install the correct version.** The following command is for a system with CUDA 11.8. If you have a different version, generate the correct command from the [official PyTorch website](https://www.google.com/url?sa=E&q=https%3A%2F%2Fpytorch.org%2Fget-started%2Flocally%2F).

pip install torch torchvision torchaudio --index-url <https://download.pytorch.org/whl/cu118>

1. **Verify CUDA availability.** Run the following command. The output **must** be True.

python -c "import torch; print(f'CUDA available: {torch.cuda.is\_available()}')"

**Step 3: Install Remaining Project Dependencies**

With the virtual environment still active, run this single command to install all other necessary libraries.

pip install transformers accelerate pillow pygame pywin32 tqdm

**Step 4: Hugging Face Login & Model Access**

The MedGemma-4b-it model is a gated model and requires authentication.

1. **Create a Hugging Face Account:** If you don't have one, sign up at [huggingface.co](https://www.google.com/url?sa=E&q=https%3A%2F%2Fhuggingface.co).
2. **Request Model Access:** Go to the [MedGemma 4B model page](https://huggingface.co/google/medgemma-4b-it) and agree to the terms to gain access.
3. **Generate an Access Token:** In your Hugging Face account settings, go to "Access Tokens" and create a new token with at least "read" permissions.
4. **Login via Terminal:** In your activated terminal, run the following command and paste your token when prompted.

huggingface-cli login

**Step 5: Final Project Setup**

1. **Download the Code:** Save the final, working code from the previous response as medgemma\_terminal.py inside your MedGemma\_AI project folder.
2. **Create Image Folder:** In the same folder, create a new directory named Image. Place any medical images (.png, .jpg, etc.) you want to analyze inside this folder.
3. **Add Audio Files:** Place your start.ogg and stop.ogg audio files in the MedGemma\_AI folder.

# 4. How to Use the Application

## 4.1. Starting the Application

1. Open a terminal.
2. Navigate to your MedGemma\_AI project folder.
3. Activate the virtual environment: .venv\Scripts\Activate.ps1
4. Run the script:
   * **To analyze the default Image folder:**

python medgemma\_terminal.py

* **To analyze a custom folder:**

python medgemma\_terminal.py -d "C:\Path\To\Custom\Images"

## 4.2. Use Cases & Test Cases

### Use Case 1: General Medical Question

* **Action:** At the You: prompt, type a question about a medical condition, treatment, or concept.
* **Example:**

You: What is the difference between an MRI and a CT scan?

* **Expected Output:** The AI will respond with a detailed, text-based explanation from its medical knowledge base.

### Use Case 2: Interactive Image Analysis

* **Action:** At the You: prompt, type analyze images.
* **Expected Output:** The application will display a numbered list of 25 medical specialties.
* **Action:** Enter the number corresponding to the desired specialist (e.g., 22 for Radiologist) and press Enter.
* **Expected Output:**
  + The application will state how many images it found in the target directory.
  + A progress bar (tqdm) will appear.
  + For each image, the terminal will print the filename followed by a detailed analysis from the perspective of the chosen specialist. The analysis text will be in bright blue.
  + After all images are processed, a completion message will appear, and the application will return to the conversational You: prompt.

### Use Case 3: Invalid Specialist Selection

* **Action:** After typing analyze images, enter a number outside the valid range (e.g., 30) or non-numeric text.
* **Expected Output:** The application will print an error message in red (e.g., "Invalid input...") and immediately return to the conversational You: prompt without crashing.

### Use Case 4: Directory Not Found

* **Action:** Start the application with the -d flag pointing to a folder that does not exist.

python medgemma\_terminal.py -d "C:\Non\Existent\Folder"

* **Expected Output:** The application will print a red error message stating the directory was not found and then return to conversational mode.

## 4.3. Exiting the Application

* **Action:** At the You: prompt, type exit or quit.
* **Expected Output:** The application will print a shutdown message, play the stop.ogg sound, and exit cleanly.
* **Alternative:** Press CTRL+C at any time. The same graceful exit process will be triggered.