

## Overview

This document provides an in-depth guide for setting up, understanding, and running the **Emotional Support Buddy (ESB)** chatbot, which leverages Retrieval-Augmented Generation (RAG) and multiple specialized LLMs to offer empathetic, safe, and contextually aware conversations.

### 1. Introduction

- **Purpose:** Build a warm, non-judgmental AI companion that validates user emotions and adapts its tone.
- **Audience:** Developers and stakeholders deploying or extending the ESB system.

### 2. Prerequisites

- Python 3.8 or higher
- ChromaDB library and service
- LM Studio instance with the following models loaded:
  - **mistral-7b-instruct-v0.3** (ESB responses)
  - **phi-3-mini-4k-instruct** (feedback directives)
  - **qwen1.5-1.8b-chat** (guardrail safety checks)
  - **text-embedding-nomic-embed-text-v1.5** (embeddings)

## Installation & Configuration

### a. Clone repository:

```
git clone https://github.com/DanielCujba/ESH.git && cd ESH
```

### b. Install dependencies:

```
pip install chromadb, openai
```

### 3. Core Components

#### 3.1. ChromaDB RAG

- **initialize\_and\_populate\_chroma\_collection():** Reads `requirement.txt`, computes embeddings, and stores rules with deterministic IDs to avoid duplicates.
- **retrieve\_relevant\_requirements\_from\_chroma(query):** Embeds user query and retrieves top-3 rules above a similarity threshold for prompt context.

#### 3.2. Prompt Engineering

- **ESB System Prompt:** Defines the bot's empathetic persona and general guidelines.

- **Feedback LLM Prompt:** Instructs a lightweight model to emit a single-line JSON directive (*action\_type*, *suggested\_tone*, *focus\_hint*) based on conversation context.

- **Guardrail LLM Prompt:** Prompts a safety model to classify candidate responses as *SAFE* or *UNSAFE*.

### 3.3. LLM Orchestration

- **get\_feedback\_instruction(history):** Builds a user-focused prompt from chat history, sends to Feedback LLM, parses JSON into an instruction string, with fallback.

- **generate\_esb\_response(history, instruction, requirements):** Merges static ESB prompt, retrieved rules, and dynamic directive to query the ESB LLM for a candidate message.

- **validate\_with\_guardrail(response):** Sends candidate to Guardrail LLM, repeats generation if marked *UNSAFE*.

## Running the Chatbot

Execute:

```
python main.py
```

4.

- The bot greets the user, then loops:
  1. Accept user input
  2. Generate directive via Feedback LLM
  3. Retrieve context from ChromaDB
  4. Generate candidate with ESB LLM
  5. Validate safety; retry if needed
  6. Output response

5. Future Roadmap

- **Long-term Memory:** Persist user preferences and history across sessions.
- **Expanded Scenarios:** Add specialized modules for crisis, grief, celebration, etc.
- **Analytics & Feedback:** Collect user ratings to fine-tune directive strategies.