🚀 Panic/Anxiety RAG Chatbot - Setup Guide (with RAG-Anything)

📋 Overview

This setup uses **RAG-Anything for preprocessing** + **custom chatbot for conversation**:

```
Your PDFs → RAG-Anything (ingest.py) → FAISS Vector Store
                                    ↓
                            Custom Chatbot
                            (safety layer + prompt)
```

🔧 STEP-BY-STEP SETUP

PART A: RAG-Anything Setup (Preprocessing)

Step A1: Clone RAG-Anything Repo

```bash
# In a working directory
git clone https://github.com/HKUDS/RAG-Anything.git
cd RAG-Anything

# Install dependencies
pip install -r requirements.txt
```

⏱ **Expected time:** 2-3 minutes

Step A2: Prepare Your PDFs

```bash
```

```
# Create docs directory
mkdir -p data/docs

# Copy your 5 trusted mental health PDFs here:
# - panic_attack_coping.pdf
# - grounding_exercises.pdf
# - deep_breathing_manual.pdf
# - CBT_panic_guide.pdf
# - WHO_anxiety_tips.pdf

ls data/docs/  # Verify all 5 PDFs are there
```

Step A3: Run RAG-Anything Preprocessing

```bash
# This preprocesses PDFs, creates embeddings, and builds FAISS vector store
python ingest.py \
  --data_dir ./data/docs \
  --vectorstore faiss \
  --chunk_size 500 \
  --chunk_overlap 50 \
  --embedding_model sentence-transformers/all-MiniLM-L6-v2
```

What happens:

1. ✅ Loads all 5 PDFs from `data/docs/`
2. ✅ Splits into 500-character chunks with 50-char overlap
3. ✅ Creates embeddings using `all-MiniLM-L6-v2`
4. ✅ Builds FAISS index for fast similarity search
5. ✅ Saves to `./vectorstore/` directory

Output files:

```
RAG-Anything/vectorstore/
├── index.faiss      # Vector index
└── index.pkl        # Metadata
```

⏱️ **Expected time:** 3-5 minutes (first run downloads embedding model)

✅ **You'll see:**

```
Loading documents...
✓ Loaded 5 PDFs
✓ Created 150 chunks (approx)
✓ Generated embeddings...
✓ Built FAISS index
✓ Saved to ./vectorstore/
```

PART B: Your Custom Chatbot Setup

Step B1: Create Project Structure

```bash
# Go back to parent directory
cd ..

# Create anxiety chatbot project
mkdir anxiety-rag-chatbot
cd anxiety-rag-chatbot

# Create directories
mkdir logs
```

Step B2: Copy Vector Store

```bash
# Copy the preprocessed vector store from RAG-Anything
cp -r ../RAG-Anything/vectorstore ./

# Verify it's there
ls vectorstore/
# Should show: index.faiss  index.pkl
```

Step B3: Create Virtual Environment

```bash
```

```
# Create venv
python -m venv venv

# Activate it
# On Linux/Mac:
source venv/bin/activate

# On Windows:
venv\Scripts\activate
```

Step B4: Install Dependencies

Create `requirements.txt`:

```
langchain==0.1.0
sentence-transformers==2.2.2
faiss-cpu==1.7.4
torch==2.0.0
transformers==4.35.0
huggingface-hub==0.19.0
```

Then install:

```bash
pip install -r requirements.txt
```

⏱️ **Expected time:** 5-10 minutes

Step B5: Place Chatbot Script

Save the main chatbot code as `chatbot.py` in your project root

Step B6: Verify Setup

```bash
# Check that everything is in place
ls -la
# Should show:
# - chatbot.py
# - requirements.txt
# - vectorstore/
# - logs/
# - venv/
```

🚀 RUNNING THE CHATBOT

First Launch

```bash
# Make sure venv is activated
source venv/bin/activate  # or on Windows: venv\Scripts\activate


# Run chatbot
python chatbot.py
```

What happens:

1. ✅ Validates vector store exists
2. ✅ Loads embeddings model (cached)
3. ✅ Initializes FAISS vector store
4. ✅ Loads LLM
5. ✅ Starts interactive chat

⏱️ **Startup time:** 30-60 seconds

You'll see:

```
🚀 PANIC & ANXIETY RAG CHATBOT - STARTUP
✅ Logs directory ready
✅ Vector store validated at ./vectorstore
📂 Loading vector store...
✅ Vector store loaded successfully
   Embedding model: sentence-transformers/all-MiniLM-L6-v2
   Total vectors indexed: 150 (approx)

🤖 Initializing LLM...
   Using HuggingFace (gpt2)
✅ LLM initialized

🎯 Creating RAG chain...
✅ RAG chain created

╔══════════════════════════════════════════════════════════
║    🧠 PANIC & ANXIETY SUPPORT CHATBOT
║    Powered by Trusted Mental Health Resources
╚══════════════════════════════════════════════════════════

You: _
```

Example Interaction

```
You: I'm having a panic attack right now, help me

⏳ Thinking...

🤖 Assistant:
I'm glad you reached out. Let's work through this together using a proven technique.

Here's what we'll do - Box Breathing:
1. Breathe in slowly for 4 counts
2. Hold your breath for 4 counts
3. Breathe out slowly for 4 counts
4. Hold for 4 counts
Repeat 5-10 times

This activates your parasympathetic nervous system, which calms your body...

📚 Sources:
  • deep_breathing_manual.pdf
  • panic_attack_coping.pdf

💡 Remember: For persistent symptoms, please consult a mental health professional.
```

```
You: _
```

⚙️ CONFIGURATION

Switch to Ollama (Recommended for Better Quality)

Install Ollama:

```bash
# Visit: https://ollama.ai
# Download and install for your OS
```

In your terminal (separate window):

```bash
# Pull and run Mistral (best quality)
ollama pull mistral
ollama serve
```

In `chatbot.py`, change:

```python
class Config:
    USE_OLLAMA = True  # Change from False to True
    OLLAMA_MODEL = "mistral"  # or "neural-chat", "orca-mini"
```

Restart chatbot:

```bash
python chatbot.py
```

Quality comparison:

Model	**Speed**	**Quality**	**Use Case**
GPT-2	⚡ Fast (2-3s)	Basic	Testing, CPU-limited
Mistral (Ollama)	⏱️ Medium (5-10s)	Excellent	Production, recommended
Neural-chat (Ollama)	⏱️ Fast (3-5s)	Very Good	Balance of speed & quality

---

🆘 TROUBLESHOOTING

"Vector store not found"

```
❌ Vector store not found at ./vectorstore
```

Fix:

```bash
# Make sure you copied vectorstore from RAG-Anything
cp -r ../RAG-Anything/vectorstore ./

# Verify
ls vectorstore/index.faiss  # Should exist
```

"No module named 'langchain'"

```bash
pip install --upgrade pip
pip install -r requirements.txt
```

"Slow startup"

- **First run:** Normal (downloads models)
- **Subsequent runs:** Should be 30-60 seconds
- **If still slow:** Close other apps, check disk space

"Out of memory"

```python
# In chatbot.py, reduce TOP_K:
TOP_K = 2  # Changed from 3
```

Or use smaller embedding model:

```python
EMBEDDING_MODEL = "sentence-transformers/all-MiniLM-L6-v2"  # Already minimal
```

Ollama connection error

```
Error connecting to Ollama
```

Fix:

```bash
# In separate terminal, make sure Ollama is running:
ollama serve

# Then run chatbot in another terminal
python chatbot.py
```

📊 PERFORMANCE METRICS

First Run (RAG-Anything preprocessing)

- **Time:** 3-5 minutes
- **What:** PDF loading, chunking, embedding, FAISS indexing
- **Output:** 150+ chunks indexed

Chatbot Startup

- **Time:** 30-60 seconds
- **What:** Load embeddings, vector store, LLM
- **One-time:** These are cached after first load

Per Query

- **Time:** 2-10 seconds (depends on LLM)
  - FAISS search: ~0.1s
  - LLM generation: 2-9s (GPT-2) or 5-10s (Mistral)
- **Memory:** ~1-2GB

📁 FINAL PROJECT STRUCTURE

```
anxiety-rag-chatbot/
├── venv/                   # Virtual environment
├── vectorstore/              # Pre-built FAISS index (from RAG-Anything)
```

```
│   ├── index.faiss          # Vector embeddings
│   └── index.pkl            # Metadata
├── logs/                    # Conversation logs (optional)
├── chatbot.py               # Main chatbot script ← RUN THIS
├── requirements.txt         # Python dependencies
└── README.md                # Documentation (optional)
```

🎯 QUICK START CHECKLIST

- [ ] Cloned RAG-Anything
- [ ] Placed 5 PDFs in `RAG-Anything/data/docs/`
- [ ] Ran `python ingest.py` successfully
- [ ] Copied `vectorstore/` to `anxiety-rag-chatbot/`
- [ ] Created Python venv
- [ ] Installed dependencies from `requirements.txt`
- [ ] Placed `chatbot.py` in project root
- [ ] Verified `vectorstore/index.faiss` exists
- [ ] First run: `python chatbot.py`
- [ ] Asked test question: "How do I breathe calmly?"
- [ ] Got response with source documents

🚀 YOU'RE READY!

```bash
# Activate venv
source venv/bin/activate

# Run chatbot
python chatbot.py

# Start chatting!
You: I feel anxious, help me
```

🆘 CRISIS RESOURCES (Built-in)

If you mention self-harm, suicide, or crisis keywords, the chatbot automatically provides:

- 🇮🇳 **India:** AASRA (1800-599-0019), iCall (9152987821)
- 🇺🇸 **US:** Crisis Text Line (text HELLO to 741741)

- 🌍 **Other:** Localized helplines

💡 OPTIONAL ENHANCEMENTS

Add Streamlit Web UI

```bash
pip install streamlit
# Create app.py with Streamlit wrapper
streamlit run app.py
```

Enable Voice

```bash
pip install pyttsx3 SpeechRecognition
# Add voice input/output
```

Conversation History

```python
# Save/load conversations as JSON
# Add to run_chatbot() function
```

Questions? Issues? Check troubleshooting section above or run with `--debug` flag 🎯