# Krishna-Inspired Emotional Support Chatbot – Full Step-by-Step Guide

This document contains a detailed, step-by-step roadmap for building a Krishna-inspired emotional support chatbot that recognizes sadness and guides users with calm, compassionate messages inspired by Krishna's teachings. The chatbot can speak in a customized real human voice provided by the creator and eventually expand to intelligent LLM-based guidance.  
  
----------------------------------------  
TABLE OF CONTENTS  
----------------------------------------  
1. High-Level Milestones  
2. Step-by-Step Implementation Guide  
3. Example Code Snippets  
4. Testing Checklist  
5. Suggested Timeline  
6. Practical Tips  
  
----------------------------------------  
1. HIGH-LEVEL MILESTONES  
----------------------------------------  
1. Project scoping & safety plan  
2. Prepare voice assets & legal consent  
3. Prototype: text emotion detection + scripted Krishna replies  
4. Add voice playback using recorded voice  
5. Add RAG + LLM for Krishna-style intelligent guidance  
6. Integrate voice cloning for dynamic TTS  
7. Add conversation management and personalization  
8. Testing & ethical review  
9. Deployment & monitoring  
10. Maintenance & improvement  
  
----------------------------------------  
2. STEP-BY-STEP IMPLEMENTATION  
----------------------------------------  
STEP 0 — Decisions  
- Choose platform: Web app (React + FastAPI)  
- Mode: Text chat + optional voice playback  
- Prepare disclaimers and privacy terms  
  
STEP 1 — Safety & Ethical Setup  
- Crisis detection and escalation flow  
- Display disclaimers for spiritual-only purpose  
- Secure storage & encrypted transmission  
  
STEP 2 — Prepare Voice Recordings  
- Record 5–10 minutes of real human voice in calm tone  
- Prepare short phrase WAV clips for greetings, empathy, comfort  
- Obtain consent for voice cloning (if used later)  
  
STEP 3 — Minimal Prototype (Text Only)  
- Backend: FastAPI (Python)  
- Emotion detection: Hugging Face transformer model  
- Create JSON file with Krishna-inspired replies  
- Return message based on detected emotion  
  
STEP 4 — Voice Playback (Static Files)  
- Map replies to pre-recorded WAV files  
- Frontend: Play audio when message received  
- Use S3/static server for hosting audio  
  
STEP 5 — Add RAG + LLM for Dynamic Guidance  
- Build Bhagavad Gita verse database  
- Generate embeddings and index using FAISS/Pinecone  
- Retrieve relevant verses and prompt LLM to respond as Krishna  
  
STEP 6 — Voice Cloning (Optional)  
- Train or register custom voice with ElevenLabs / Resemble / Azure TTS  
- Generate speech dynamically from LLM output  
  
STEP 7 — Dialogue Management  
- Track emotion state and recent context  
- Offer grounding audio if sadness persists  
- Personalize messages with user name  
  
STEP 8 — Testing  
- Evaluate emotion accuracy, tone, safety response  
- Verify voice sync and crisis escalation  
- Refine and gather user feedback  
  
STEP 9 — Deployment & Monitoring  
- Backend: AWS ECS / Google Cloud Run / Render  
- Frontend: Vercel / Netlify  
- Database: Pinecone / FAISS / PostgreSQL  
- Set up logging, cost tracking, moderation review  
  
STEP 10 — Maintenance & Improvement  
- Add multi-language support  
- Periodically retrain emotion model  
- Expand voice library  
  
----------------------------------------  
3. EXAMPLE CODE SNIPPETS  
----------------------------------------  
Python emotion detection example:  
from transformers import pipeline  
emotion = pipeline("text-classification", model="j-hartmann/emotion-english-distilroberta-base")  
def detect\_emotion(text):  
 res = emotion(text)[0]  
 return res['label'], res['score']  
  
LLM Krishna prompt example:  
"You are Krishna: wise, compassionate, concise. Use the provided verses as grounding.  
If user shows suicidal intent, provide crisis resources and avoid spiritual-only messages."  
  
----------------------------------------  
4. TESTING CHECKLIST  
----------------------------------------  
- Crisis triggers and escalation tested  
- Audio and text mapping verified  
- Voice cloning consent documented  
- Privacy and disclaimer displayed  
- LLM never provides medical advice  
  
----------------------------------------  
5. SUGGESTED TIMELINE (7 WEEKS)  
----------------------------------------  
Week 0: Planning, voice prep, safety docs  
Week 1: Text prototype  
Week 2: Add voice playback  
Week 3–4: RAG + LLM  
Week 5: Voice cloning integration  
Week 6: Testing  
Week 7: Launch MVP  
  
----------------------------------------  
6. PRACTICAL TIPS  
----------------------------------------  
- Begin with static audio before cloning  
- Keep fallback pre-approved verses for accuracy  
- Use caching to save LLM/TTS costs  
- Ensure compliance with privacy and ethical guidelines  
  
----------------------------------------  
END OF DOCUMENT  
----------------------------------------