Software Requirement Specification (SRS)

\*\*Project Title:\*\* My Voice Clone Agent

\*\*Prepared By:\*\* Himanshi Shrivastav

\*\*Technology Stack:\*\* Python, Deep Learning, Voice Cloning, NLP, Audio Processing

# 1. Introduction

## 1.1 Purpose

This document defines the requirements to develop an AI-powered virtual voice agent that can talk exactly like the user, Himanshi, replicating her voice, tone, pitch, language style, and humor. The voice model will be trained on real-life call recordings and voice messages, even if the recordings contain multiple speakers.

## 1.2 Scope

The system will:  
- Accept multi-speaker voice data (calls, messages)  
- Identify and extract Himanshi’s voice segments  
- Train a neural TTS model to generate speech in her voice  
- Allow real-time interaction in cloned voice  
- Run offline for privacy and control

## 1.3 Intended Audience

• Freelancers and developers interested in voice tech  
• AI/ML students or professionals  
• Personal assistant creators  
• Human-computer interaction researchers

# 2. Overall Description

## 2.1 Product Perspective

This product is a standalone application or backend engine that allows training and generation of cloned voice, based on the user's original speech. It can be optionally integrated with a GUI or chatbot frontend.

## 2.2 User Characteristics

• The user will provide recorded calls and messages  
• The user knows which speaker is them (manual confirmation needed)  
• The user is female (used for gender verification)

## 2.3 Assumptions

• User has clear recordings with sufficient samples of their own voice  
• User will confirm identity of speaker during diarization labeling  
• Internet may be used during training, but the final product runs offline

# 3. Functional Requirements (Detailed)

* FR1: Voice Data Upload  
  - Support .wav, .mp3, .ogg formats, batch upload
* FR2: Speaker Diarization  
  - Separate different speakers, label as Speaker 1, 2 etc.
* FR3: Manual Speaker Labeling  
  - User marks 'Himanshi' speaker manually
* FR4: Gender Identification (Optional)  
  - Cross-check selected speaker as female
* FR5: Data Cleaning & Alignment  
  - Trim silence, normalize audio, transcribe
* FR6: Voice Cloning Model Training  
  - Use TTS model to train on filtered voice
* FR7: Voice Generation (Text-to-Speech)  
  - Input text, output audio in cloned voice
* FR8: Real-Time Interaction (Speech-to-Speech)  
  - Mic input -> STT -> response with cloned voice
* FR9: Memory & Personality Layer  
  - Add humor, slang, personality tone to responses

# 4. Non-Functional Requirements

|  |  |
| --- | --- |
| Requirement | Description |
| Performance | Should generate response in < 2 seconds |
| Accuracy | 90%+ similarity in tone, pitch, delivery |
| Offline Support | Entire system should run offline post-training |
| Security | All training data is stored locally, no cloud processing |
| Scalability | Modular, support retraining with new data |

# 5. System Design (Stage-wise)

1. Stage 1: Data Ingestion & Preprocessing
2. Stage 2: Speaker Diarization
3. Stage 3: Speaker Identification
4. Stage 4: Transcription & Alignment
5. Stage 5: Voice Cloning Model Training
6. Stage 6: Response Generator
7. Stage 7: Interactive Agent Layer (Optional)

# 6. Tools & Libraries

|  |  |
| --- | --- |
| Category | Tools / Libraries |
| Diarization | pyannote.audio, Resemblyzer, Kaldi |
| Voice Cleaning | pydub, librosa, sox |
| Transcription | Whisper, SpeechRecognition, Vosk |
| TTS Model | Coqui TTS, Tacotron2, YourTTS |
| UI | Flask, Tkinter, React.js (optional) |
| Speech Input | PyAudio, sounddevice |

# 7. Dataset Requirements

• At least 10–20 minutes of your voice (high-quality)  
• Must include different emotions and speaking speeds  
• Should include daily phrases, personal expressions, jokes

# 8. Future Scope

• Add lip-sync facial avatar with MediaPipe or DeepFace  
• Create API endpoint for cloned voice usage  
• Add context-aware NLP model (using RAG or GPT-2)  
• Multi-language support (Hinglish, Hindi)