Multilingual Text-to-Speech (TTS) + Voice Cloning

This project implements a multilingual Text-to-Speech (TTS) and Voice Cloning system using [Coqui TTS](https://github.com/coqui-ai/TTS) — running entirely on CPU with automatic language translation for non-supported inputs.
It converts text into natural-sounding speech and can clone a user's voice from short audio samples.

🌍 Features

- **Text-to-Speech (TTS)** – Convert text to realistic human speech.
- **Voice Cloning** – Clone your own or any voice using short recordings.
- **Multilingual Support** – English US, French FR, Portuguese BR (+ auto-translate for others).
- **Streamlit Frontend** – Simple, modern web UI.
- **FastAPI Backend** – High-performance API service.
- **Fully Local** – Works offline (no GPU or cloud dependency).
- **Auto Translation** – Detects and translates non-supported languages to English.

□ System Architecture

```
                 User Input (Text or Audio)
                            │
                            ▼
                 Streamlit Frontend (UI)
                            │
                            ▼
FastAPI Backend ─► Language Detection & Translation
                            │
                            ▼
  Coqui YourTTS Model (Multilingual + Voice Cloning)
                            │
                            ▼
                     Audio Generation
                            │
                            ▼
                    Output (.wav file)
```

Folder Structure

```
tts_multilang_clone/
│
├── app.py # FastAPI backend (TTS + cloning API)
├── app_ui.py # Streamlit user interface
├── requirements.txt # Dependencies list
├── venv/ # Virtual environment (created later)
├── uploads/ # Uploaded voice samples
├── outputs/ # Generated speech/cloned audio
└── README.md # Documentation
```

⚙️ Installation Guide

1. Clone or Download the Project

```bash
git clone https://github.com/YOUR_USERNAME/tts_multilang_clone.git
cd tts_multilang_clone
```

2. Create a Virtual Environment

```
python -m venv venv
venv\Scripts\activate
```

📦 3. Install Dependencies

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

If you don’t have the file, install manually:

```
pip install fastapi uvicorn streamlit TTS==0.22.0 librosa==0.9.2 numpy==1.26.4 pydub deep-translator
```

🚀 Running the Project

🖥️ Start the Backend

```
python app.py
```

You should see:

```
INFO:     Uvicorn running on http://127.0.0.1:8000
```

🌐 Start the Frontend

Open a new terminal and run:

```
venv\Scripts\activate
streamlit run app_ui.py
```

The Streamlit app will open in your browser:
☞ http://localhost:8501

🎧 Using the Application

▶️ Text-to-Speech

1. Enter text in any language.
2. Choose language (en, fr-fr, or pt-br).
3. Click **Generate Speech**.
4. Listen to the generated output or download it from /outputs.

🗣️ Voice Cloning

1. Upload a short voice sample (5–10 sec, .wav/.mp3).
2. Enter text to be spoken in that cloned voice.
3. Choose the desired language.
4. Click **Clone Voice**.
5. The cloned voice audio is saved automatically under /outputs.

How It Works

1. **Frontend (Streamlit):** Collects user input text/audio.
2. **Backend (FastAPI):**
   - Detects and translates unsupported languages (via Deep Translator).
   - Sends the processed text/audio to the model.
3. **Model (YourTTS):**
   - Generates natural speech or cloned voice output.
4. **Output:** The resulting .wav file is streamed back to the user.

🌐 Supported Languages

Type	Code	Notes
English	en	Fully supported
French	fr-fr	Fully supported
Portuguese	pt-br	Fully supported
Hindi, Spanish, etc.	Auto-translated to English	

🔍 Troubleshooting

Issue	Cause	Fix
ConnectionError	Backend not running	Run python app.py first
CUDA not available	No GPU	Ignore; runs on CPU
KeyError: 'hi'	Language unsupported	Auto-translates to English
Port 8000 already in use	Another process active	Change port in app.py (e.g., 8010)

□ Model Information

- **Base Model:** tts_models/multilingual/multi-dataset/your_tts (Coqui-AI)
- **Architecture:** Based on VITS (Variational Inference + GAN)
- **Features:**
  - End-to-end text-to-wave generation
  - Multilingual & multi-speaker training
  - Zero-shot voice cloning (speaker embedding extraction)

□ Example API Endpoints

POST /tts/

Generate TTS output.

Form fields:

- text: Input text
- language: en / fr-fr / pt-br

POST /clone/

Clone voice and generate speech.

Form fields:

- text: Text to speak
- file: Uploaded .wav or .mp3 sample
- language: Language code

📂 Output Directory

Generated audio files are stored in:

tts_multilang_clone/outputs/

Example files:

tts_1c4f9a9b1c.wav
clone_a64dbd879e.wav

□ Future Enhancements

- Extend multilingual support (add Hindi, Spanish, German).
- Integrate emotion-based synthesis (happy, sad, neutral).
- Add waveform visualizer in Streamlit.
- Add voice comparison module for accuracy evaluation.

👨‍💻 Author

Rishabh Upadhyay
M.Tech Artificial Intelligence | Amity University, Noida
📧 LinkedIn • GitHub