Voice Translation App: Functional Documentation

# Overview

The Voice Translation App is a mobile-responsive web application designed to enable real-time voice translation. It provides speech-to-text transcription, language translation, and text-to-speech synthesis to facilitate communication across different languages. Built using ReactPy, the app features an intuitive, mobile-first interface with support for selecting predefined or custom languages, displaying transcriptions and translations, and playing translated audio. This documentation details the app's core functionalities and UI elements for developers, stakeholders, and technical users, focusing on its capabilities without including a user guide.

# Key Functionalities

## Language Selection

- \*\*Input Language\*\*: Users can select the language of the spoken input from a predefined list (e.g., English, Spanish, French, German, Chinese) via a dropdown menu. Alternatively, users can type a custom language to accommodate unsupported languages or dialects.

- \*\*Output Language\*\*: Users choose or type the target language for translation, using the same predefined list or custom input option.

- \*\*Implementation\*\*: Two dropdown selectors in a top bar allow language selection. The UI supports custom input through text fields within the dropdown, with responsive design for mobile (vertical stacking) and larger screens (horizontal alignment).

## Audio Recording and Transcription Display

- \*\*Recording Mechanism\*\*: A floating action button (FAB) toggles audio recording. It displays a microphone icon (🎙️) when idle and a stop icon (⏹️) during recording, changing color from purple to red for visual feedback.

- \*\*Transcription Output\*\*: After stopping the recording, the app displays the transcribed text of the input audio in the selected input language. The transcription appears in a card with a "Transcript" header and a scrollable text area for lengthy content.

Backend Processing\*\*: Transcription leverages server-side processing with AI-enhanced accuracy, optimized for medical terminology.

## Translation Display

- \*\*Translation Process\*\*: A "Translate" button triggers translation of the transcribed text into the selected output language.

- \*\*Display Format\*\*: The translated text is shown in a separate card labeled "Translation," with a scrollable area for long text. The stacked card layout allows simultaneous viewing of the original transcript and translation.

- \*\*Real-Time Feedback\*\*: Translation results are updated dynamically upon user action, maintaining a clean and accessible presentation.

## Audio Playback

- \*\*Text-to-Speech\*\*: A "Play Translated Audio" button converts the translated text into audio for playback.

- \*\*Audio Player\*\*: An HTML5 audio player appears below the translation card, offering standard controls (play, pause, volume, seek).

- \*\*Seamless Integration\*\*: Audio is generated on-demand and played directly in the browser, ensuring a smooth user experience.

# UI and Design Features

- \*\*Mobile-First Layout\*\*: Optimized for small screens with large, tap-friendly buttons, responsive typography (e.g., `text-sm` for mobile, `sm:text-base` for larger screens), and a constrained width (`max-w-2xl`) for desktops. Uses Tailwind CSS for styling.

- \*\*Color Palette\*\*: Employs soft blues (e.g., `bg-blue-800` for the top bar, `bg-blue-600` for buttons), purples (e.g., `bg-purple-600` for playback and FAB), and neutral grays (`bg-gray-100` background, `bg-gray-50` text areas) for a professional, inviting aesthetic.

- \*\*Stacked Cards\*\*: Transcript and translation are displayed in white cards (`bg-white`, `shadow-md`, `rounded-lg`) with scrollable text areas (`min-h-[100px]`, `max-h-[200px]`).

- \*\*Floating Action Button (FAB)\*\*: Positioned at the bottom-right (`bottom-6 right-6`), sized for easy tapping (`w-16 h-16` on mobile, `sm:w-20 sm:h-20`), with dynamic icons and colors.

- \*\*Top Bar\*\*: Language selectors are housed in a responsive top bar (`flex-col sm:flex-row`, `bg-blue-800`), with clear labels and focus states (`focus:ring-purple-500`).

- \*\*Responsive Typography\*\*: Headers (`text-2xl` to `md:text-4xl`), card titles (`text-lg` to `sm:text-xl`), and body text (`text-sm` to `sm:text-base`) scale for readability.

- \*\*Minimal Design\*\*: Rounded corners (`rounded-lg`), subtle shadows (`shadow-md`), and smooth transitions (`transition duration-200`) ensure a clean, uncluttered interface.

# Technical Notes

- \*\*Backend Integration\*\*: Relies on server-side endpoints for recording (`/start\_recording`, `/stop\_recording`), transcription (using OpenAI's API), translation, and text-to-speech. The frontend communicates via HTTP requests (e.g., `aiohttp`).

- \*\*Recording Control\*\*: The "Stop Recording" button sends a signal to the server to halt PyAudio recording, process the audio, and return the transcript, replacing the original Enter key dependency.

- \*\*Error Handling\*\*: Displays user-friendly error messages in the transcript or translation cards for issues like microphone access or server failures.

- \*\*Custom Language Support\*\*: Custom language inputs are passed to the backend, with accuracy dependent on API capabilities (e.g., OpenAI transcription models).

- \*\*Dependencies\*\*: Built with ReactPy, FastAPI, PyAudio, OpenAI, and aiohttp. Requires a valid `OPENAI\_API\_KEY` in a `.env` file.

- \*\*Limitations\*\*: Audio quality affects transcription accuracy; server-side processing requires a stable internet connection; custom languages may have variable support.

# Version Information

- \*\*Date\*\*: August 20, 2025

- \*\*Status\*\*: Based on the latest implementation with a mobile-first, responsive UI and server-side PyAudio recording.

This documentation focuses on the app's functionalities and design, omitting usage instructions as requested. For implementation details or troubleshooting, refer to the source code or contact the developer.