Healthcare Translation Application

Code Documentation

Frontend Structure

HTML (index.html)

- Contains UI layout (language selector, recording controls, text display)
- Manages audio playback interface
- Responsive container-based design

JavaScript (app.js)

- Core functionality: WebSocket management, audio recording, transcription, translation, text-to-speech, session management

CSS (inline styles)

- Responsive layout with mobile compatibility
- Interactive UI elements with hover states

Main Function Blocks

1] startRecordingAndTranscribing(): Initializes recording session, establishes WebSocket, handles audio capture.

2] convertTextToSpeech() : Processes translated text, sends TTS request to backend, manages audio playback.

3] SaveTranscriptToFile(): Creates text output from translation, generates downloadable file, sends data to server for storage.

4] ClearSession(): Resets application state, clears text displays, and stops audio playback.

AI Tools Integration

Speech Recognition

Real-time transcription via WebSocket, processes audio for continuous recognition, supports language detection.

Translation Engine

Supports 10+ languages, real-time translation, maintains formatting.

Text-to-Speech

Converts text to speech, handles different accents/dialects, streams audio.

Data Flow

1. Audio capture → WebSocket → Backend processing
2. Transcription results → Frontend display
3. Translation → Text display
4. TTS request → Audio URL generation → Audio playback

Security Considerations

- Local development server (127.0.0.1:8000)
- Basic WebSocket
- Plaintext data transmission
- Local file saving
- Browser microphone access