Voice-Based AI Assistant – Project Documentation

1. Project Overview

This voice-based assistant listens for a wake word, records and transcribes user speech, detects the user's intent through semantic similarity, and performs actions like opening a website or replying through text-to-speech. It supports hands-free operation and provides visual and audio feedback.

2. Libraries and Their Functionalities

- vosk – Offline wake word detection using KaldiRecognizer.
- transformers (Whisper) – Transcribes recorded audio to text.
- sentence-transformers – Encodes and compares text for intent classification.
- gTTS – Converts text into speech (MP3).
- pygame – Plays the assistant's voice response.
- sounddevice – Captures microphone input.
- soundfile – Saves recorded audio to file.
- librosa – Loads and resamples audio for Whisper model.
- webbrowser – Opens websites (YouTube, Google, etc.) based on user intent.
- win11toast – Shows toast-style desktop notifications on Windows.
- os, sys, tempfile, time, queue, json, pathlib – Utility functions for I/O, flow control, and path management.

3. Working Flow

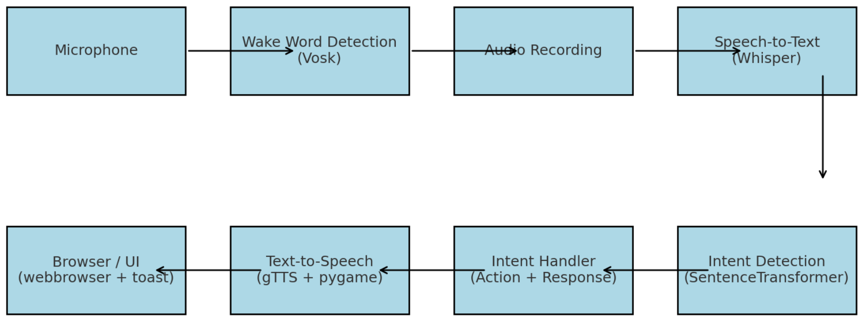
- 1. Loads models for wake-word detection, transcription, and intent classification.
- 2. Continuously listens for the wake word ('hello') using Vosk.
- 3. On wake word detection, records 4 seconds of audio input.
- 4. Transcribes the audio using Whisper.
- 5. Converts the transcription into an embedding and finds the closest intent.
- 6. Responds by speaking, opening a browser, or exiting based on detected intent.
- 7. Displays a toast notification with the response.
- 8. Returns to listening for the next wake word.

4. Function-Level Breakdown

Here's a breakdown of key functions:

1. record_audio() – Records 4 seconds of microphone input and saves it as a .wav file.
2. transcribe_audio() – Uses Whisper to convert recorded audio into text.
3. get_best_intent() – Matches the transcription to a known intent using cosine similarity.
4. speak() – Converts assistant response into MP3 using gTTS and plays it via pygame.
5. show_toast() – Displays Windows toast notification using win11toast.
6. handle_command() – Central logic that handles transcription, intent detection, and response dispatch.
7. listen_for_wake_word() – Continuously listens using Vosk for the wake word ('hello').
8. callback() – Handles audio streaming and pushes it into a queue for Vosk.

5. Data Flow Diagram


Microphone
Wake Word Detection (Vosk)
Audio Recording
Speech-to-Text (Whisper)
Intent Detection (SentenceTransformer)
Intent Handler (Action + Response)
Text-to-Speech (gTTS + pygame)
Browser / UI (webbrowser + toast)

6. Voice Assistant – End-to-End Summary

This assistant provides an offline-capable, natural interaction interface using audio commands. It supports various queries like weather, location, YouTube search, translation, etc., while handling casual small talk as well. It loops forever until a stop intent is issued, making it suitable for continuous desktop assistant usage.