OratorAI

# Description

## Introduction

OratorAI is an advanced voice assistant that leverages state-of-the-art AI models to provide real-time voice interaction capabilities. The solution integrates OpenAI's Whisper for speech-to-text conversion, Piper for text-to-speech synthesis, and Google's Gemini-1.5-Pro model for intelligent conversational responses. This document outlines the design choices, models, libraries, and parameters used to build OratorAI.

## Models

1. **Whisper**: A robust model by OpenAI for speech-to-text conversion, known for its high accuracy in transcribing spoken language into text. It is particularly effective in noisy environments and supports multiple languages.  
2. **Piper**: A fast and efficient text-to-speech model that converts text into natural-sounding speech. We utilized both male and female voice models to provide users with customization options.  
3. **Gemini-1.5-Pro**: Google's conversational AI model, used to generate intelligent and context-aware responses based on the transcribed text.

## Libraries Used

The following Python libraries were utilized in the development of OratorAI:  
1. **torch**: For loading and utilizing the Whisper model.  
2. **transformers**: Provides the pipeline functionality for both speech-to-text and model-based tasks.  
3. **piper-tts**: A library for text-to-speech synthesis, handling the Piper models.  
4. **langchain**: Used for chaining together the AI model's responses and managing conversational context.  
5. **dotenv**: For securely managing API keys and other environment variables.  
6. **requests**: To handle any HTTP requests, particularly for fetching external data if needed.

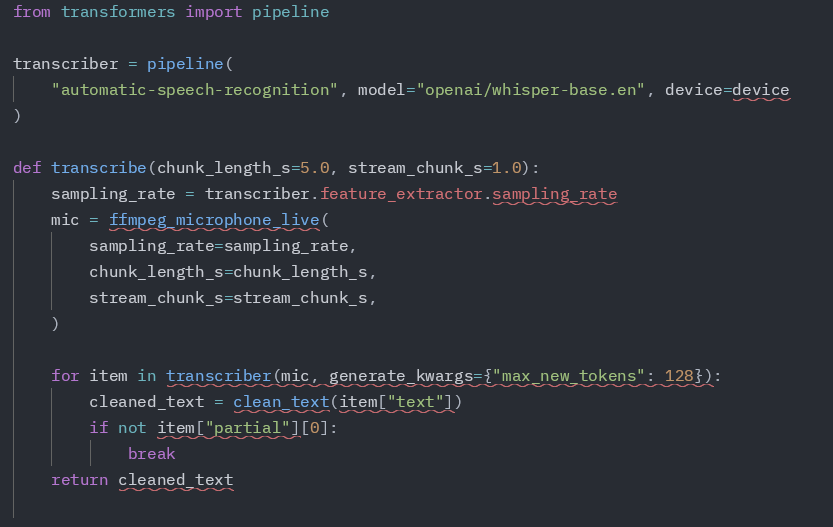
## Parameters and Configurations

Several parameters were configured to optimize the performance and flexibility of OratorAI:  
1. **Sample Rate**: Set to 22050 Hz for both Whisper and Piper models to balance quality and performance.  
2. **Voice Selection**: Users can dynamically switch between male and female voices, which are loaded from pre-trained models.  
3. **API Key Management**: The API key for Google’s Gemini-1.5-Pro model is securely stored in a `.env` file.

# Code Snippets

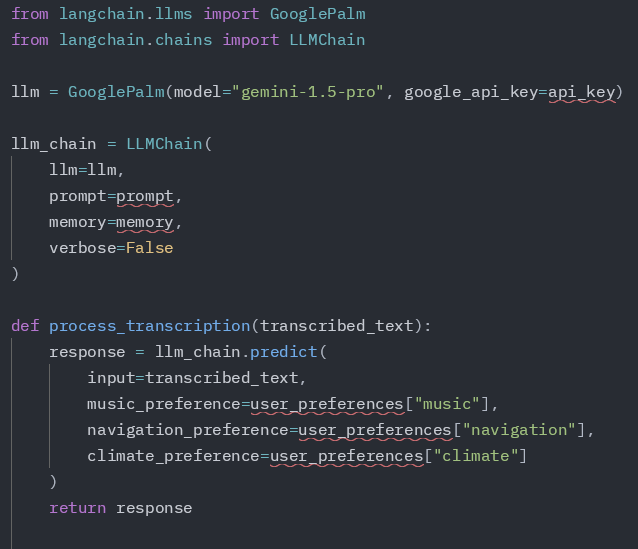
## Speech-to-Text Implementation

The following code demonstrates how the Whisper model is used for real-time speech-to-text conversion:



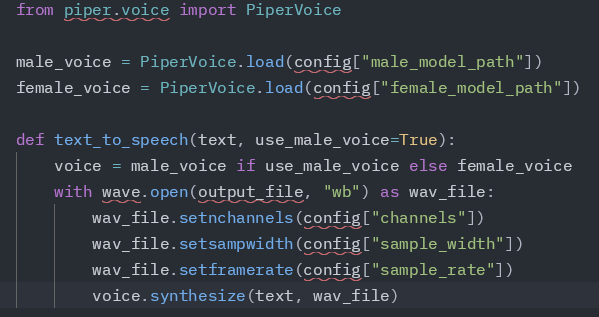
## Conversational AI Integration

Here’s how the Gemini-1.5-Pro model is integrated to generate intelligent responses:



## Text-to-Speech Implementation

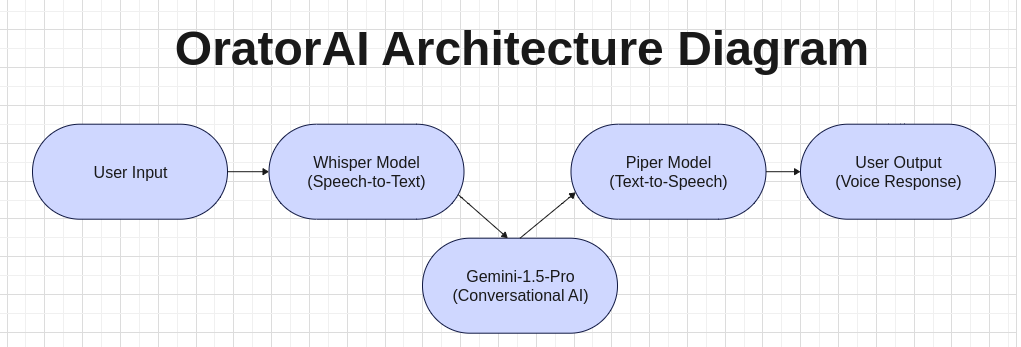
The Piper model is used to convert text back into speech, offering both male and female voices:



# Relevant Documentation and Diagrams

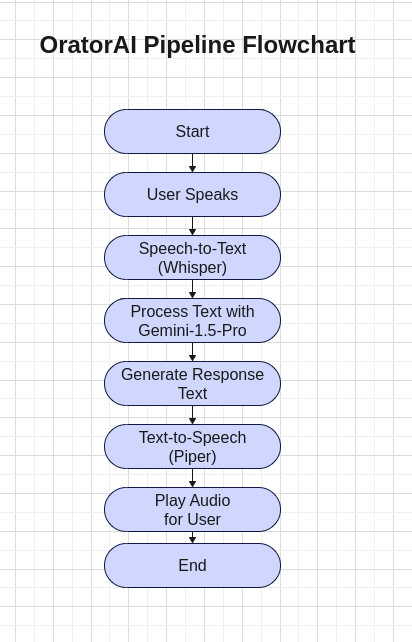
## Architecture Diagram

Below is a diagram that illustrates the architecture of OratorAI, showing the interaction between the models, libraries, and components:



## Flowchart of the Pipeline

The flowchart below outlines the pipeline process from speech input to AI response and text-to-speech conversion:



# Conclusion

In conclusion, this project demonstrates the successful integration of advanced AI models to create a seamless user experience. The OratorAI pipeline effectively transforms user speech into meaningful interactions, paving the way for future innovations in AI-driven communication.