DAY 8

Text-to-Speech and Speech-to-Speech Applications Using Gemini, ElevenLabs, and AssemblyAI

Bringing AI to Life with Voice: Text-to-Speech and Speech-to-Speech Systems

Today's session was all about transforming text and spoken words into **real, natural-sounding speech** using modern AI tools. I worked on two exciting projects that brought voice to AI: one converting **text into audio**, and the other converting **spoken questions into spoken responses** — building the core of a conversational assistant.

Text-to-Speech (TTS) using Gemini + ElevenLabs

In this project, I created a simple but powerful tool that takes a user's typed question, passes it to **Google Gemini** for a smart AI-generated response, and then uses **ElevenLabs API** to convert that response into **lifelike spoken audio**.

Key Features

- Takes user input as plain text.
- Gemini generates a natural language response.
- ElevenLabs converts the response to speech.
- The voice output is saved as a .wav file (e.g., t_to_v_001.wav).

How It Works

- **Input**: User types a question in the terminal (e.g., “What is AI?”).
- **Processing**:
  - Gemini processes the prompt and generates a high-quality reply.
  - A new .wav filename is automatically generated for each session.
- **Output**: The response is passed to ElevenLabs, and the returned audio is saved locally.

Technologies Used

Tool/API	**Purpose**
google.generativeai	Text generation via Gemini
ElevenLabs API	High-quality voice synthesis
dotenv	Secure API key loading
re,os	Filename generation and file saving

Why It Matters

- Turns any AI reply into realistic human speech.
- Helps build audio-based interfaces for education, accessibility, and support.

- Enables voice-based feedback for chatbots or AI teaching assistants.

This project completed the first half of a voice experience: **from text → to speech**.

Speech-to-Speech Using AssemblyAI + Gemini

This project brought the **second half of the voice loop** — allowing a user to speak into a microphone, letting AI understand it, and then responding out loud.

It combines:

- **Speech-to-Text** via **AssemblyAI**
- **AI response** via **Gemini**
- **Text-to-Speech** using a voice engine like pyttsx3 or ElevenLabs

Workflow

1. **User speaks a question** for 5 seconds (e.g., “Explain the water cycle”).
2. The audio is cleaned and formatted as a .wav file.
3. It’s uploaded to **AssemblyAI**, which transcribes it into text.
4. The transcription is sent to **Gemini**, which returns a smart reply.
5. The reply is **converted to speech** and saved as an audio file.
6. The assistant speaks the reply aloud and logs the conversation.

Tech Stack

Library/Service	**Purpose**
sounddevice,pydub	Recording and formatting audio
AssemblyAI API	Real-time transcription
Gemini	Natural language understanding
pyttsx3 or ElevenLabs	Text-to-Speech conversion
dotenv	API key management
os,time,requests	File handling HTTP polling

Experience

- ❖ Smooth integration of audio, transcription, and AI response.
- ❖ Voice felt natural and responsive — like talking to a personal assistant.
- ❖ Each interaction felt intelligent and human-like.

Why It Stands Out

This project closes the loop: **You speak → AI thinks → AI speaks back.**

It's not just smart — it's conversational.

Final Takeaways

Project	Input	Output	Core Tech
Text-to-Speech	Text	Spoken .wav file	Gemini + ElevenLabs
Speech-to-Speech	Voice	Spoken .wav file	AssemblyAI + Gemini + TTS

Both tools successfully demonstrated **voice-based AI applications** in real-time. With just a microphone and two API keys, I built the foundation of a **voice-first intelligent assistant**.

Real-World Applications

- Virtual assistants that respond like a human
- AI tutors or e-learning bots with real voices
- Accessibility tools for visually impaired users
- Voice AI agents in customer support or therapy
- Multimodal AI systems that process sound + language together

What I Learned

- How to handle full audio workflows: input, process, output
- How to chain together APIs and services to build a real conversation system
- How to manage file formats and handle real-time interactions
- How voice adds **emotional depth** to AI responses