

AI Podcast App – System Overview





This web-based interactive podcast app allows users to either:

- Tap a **canned question**, or
- Ask their **own question via microphone**

Leonardo da Vinci then answers — in voice — using GPT-4 and ElevenLabs.

System Architecture

1. Frontend: `/pages/index.js`

- Built with **Next.js** (deployed via **Vercel**)
- UI includes:
 -  `podcast.mp3` autoplay intro
 -  Canned questions
 -  Custom voice input (mic recording with 2s silence detection)
 -  Pause/Resume button logic tied to audio playback state
- Audio output handled via:

```
js
CopyEdit
<audio src="/api/ask-stream?question=..." autoplay />
```

2. Backend Proxies

`/api/transcribe.js`

- Accepts `audio/ogg` (recorded via mic)
- Uses OpenAI Whisper to transcribe speech to text

`/api/ask-stream.js` (see canvas)





- Takes user question (from transcript or canned)
 - Sends it to **GPT-4** with persona prompt:
"You are Leonardo da Vinci. Answer as he would, thoughtfully and clearly."
 - Sends GPT's answer to **ElevenLabs** using `text-to-speech/stream`
 - Uses `Readable.fromWeb()` to stream audio back to the client
-

Required Environment Variables

Set in `.env.local`:

```
env
CopyEdit
OPENAI_API_KEY=sk-...
ELEVENLABS_API_KEY=eleven-...
ELEVENLABS_VOICE_ID=EXAVITQu4vr4xnSDxMaL
```

What Works Now

- Desktop Chrome: Fully functional, fast, reliable
 - iOS Safari & Chrome:
 -  Transcription works
 -  GPT/ElevenLabs pipeline works
 -  Audio playback has noticeable delay (up to 10s)
 -  Eventually plays response correctly
-

Known Issues

1. **Mobile audio delay**
 - iOS Safari buffers streamed audio longer than desktop
 - No errors, but playback is slow to begin
 2. **No response caching**
 - Repeated questions hit GPT + ElevenLabs again
 - Can be optimized via memoization or edge caching
 3. **No SSR/SEO**
 - Entirely client-driven
 - That's fine for your podcast use case
-

Public Assets

- `public/podcast.mp3`: Leonardo's opening monologue
 - `public/silent.mp3`: Used to unlock iOS autoplay
 - `public/leonardo.jpg`: Optional image
-

Testing Completed

Feature	Desktop	iOS
Canned questions	✓	✓
Mic transcription	✓	✓
GPT answer	✓	✓
ElevenLabs audio	✓	✓
Autoplay delay	✗	⚠
UI Pause/Resume	✓	✓

Files Overview

- `pages/index.js` — Frontend interface:
 - Canned questions
 - Voice input (mic recording with silence detection)
 - Audio playback with pause/resume logic
- `pages/api/ask-stream.js` — Serverless API:
 - Sends the question to GPT-4
 - Sends the GPT answer to ElevenLabs
 - Streams back voice audio to the browser
- `pages/api/transcribe.js` — Upload endpoint:
 - Receives mic recording
 - Uses OpenAI Whisper to transcribe
- `public/silent.mp3` — 1-sec silent clip to unlock iOS autoplay
- `public/podcast.mp3` — Optional Leonardo intro message

Environment Variables

Set the following in `.env.local`:

```
env
CopyEdit
OPENAI_API_KEY=sk-...
ELEVENLABS_API_KEY=eleven-...
ELEVENLABS_VOICE_ID=EXAVITQu4vr4xnSDxMaL
```

Testing Results

Feature	Desktop	iOS
Canned questions	✓	✓
Mic transcription	✓	✓
GPT answer	✓	✓
ElevenLabs audio	✓	✓
Autoplay delay	✗	⚠
Pause/Resume UI	✓	✓

Known Limitations

- **iOS Audio Delay:** iOS buffers streamed ElevenLabs audio longer than desktop.
 - **No Caching:** Each question re-triggers GPT and ElevenLabs.
 - **UX Gaps:** No loading animation or audio cue during long waits.
-

Future Ideas

- Use ElevenLabs async TTS API to preload full MP3s
- Add CDN caching layer
- Improve mobile UX (e.g., “Da Vinci is thinking...” loading animation)