

Objective: Develop a real-time, conversational voice interface using the Gemini Live API, replicating the functionality of the existing Revolt Motors chatbot.

1. Analyze the Benchmark

Please visit revoltmotors.com. On the bottom right corner, click on "Rev" and then "Voice Chat" (or use the direct link provided below).

Interact with the assistant to understand its capabilities. Pay close attention to:

- The natural flow of conversation.
- The ability to speak in various languages.
- How the system handles user interruptions.
- The response latency.

2. The Task: Replication

Your task is to use the Gemini Live API to replicate this web application. The core functionality must mirror the live version (live.revoltmotors.com).

Functional Requirements:

- **Interruptions:** The user must be able to interrupt the AI while it is speaking; the AI should stop, listen to the new input, and respond appropriately. While the Gemini Live API handles interruptions natively, you must ensure your implementation allows this feature to function smoothly.
- **Latency:** The time between the end of the user's question and the start of the AI's response should be low (aim for the 1-2 second benchmark of the live application).
- **UI/Frontend:** The design does not need to be replicated exactly. A clean, functional, and simple interface is sufficient.

Technical Constraints:

- **Architecture:** The solution must use a **server-to-server** architecture. The difference between server-to-server and client-to-server is explained in the Gemini Live API docs.
- **Stack:** The backend must be built using **Node.js/Express**.

API Keys and Model Selection:

Create a free API key at aistudio.google.com. Your final submission must be configured to use the native audio dialog model: `gemini-2.5-flash-preview-native-audio-dialog`. However, be aware that this model has strict rate limits on the free tier. For development and extensive testing, we recommend temporarily switching to `gemini-2.0-flash-live-001` or `gemini-live-2.5-flash-preview` to

avoid request-per-day limitations. Additionally, the interactive playground at <https://aistudio.google.com/live> is a useful resource for understanding the API's behavior.

3. System Instructions

Once your web app is working as expected, write a simple system instructions prompt so the AI only talks about Revolt motors.

4. Use of AI Tools

A primary goal of this assessment is to evaluate your proficiency in using AI (e.g., Claude Code, Cursor, ChatGPT, Gemini, etc.) to accelerate development. You are encouraged to use any AI tool you prefer.

You are free to use any online resources or documentation.

5. Submission Requirements

1. **Demo Video:** Submit a 30-60 second screen recording demonstrating your working application. The video should show:
 - Natural conversation with the AI
 - Clear interruption of the AI mid-response (demonstrating the interruption feature works correctly)
 - Overall responsiveness and latency

Upload the video to Google Drive and share the link with public viewing permissions enabled.

2. **Source Code:** GitHub repository link containing your complete implementation and a README with setup instructions.

Links:

- Rev Live web app (The Benchmark): <https://live.revoltmotors.com/>
- Gemini Live API documentation: <https://ai.google.dev/gemini-api/docs/live>
- Interactive Playground: <https://aistudio.google.com/live>
- Example Applications: <https://ai.google.dev/gemini-api/docs/live#example-applications>