

AI Usage in the Project

- <https://eduardz1.github.io/La-Nuit-de-L-Info/>

In our submission, AI is primarily utilized through the “Know More” button, available after each question related to pollution in the environment, particularly in oceans. This feature submits a query to Gemini using a custom prompt to retrieve detailed information on the topic.

For future improvements, we propose integrating a chatbot to address user questions that may arise after completing the “game.”

We also utilized AI (suno.ai) to generate the jingle for the podcast.

Relevant Code Implementation

The implementation relies on the NPM package `@google/generative-ai` to access the Gemini APIs. This approach simplifies the content creation process by shifting the responsibility from developers to generative AI, making the solution both efficient and scalable.

```
1 import { GoogleGenerativeAI } from "@google/generative-ai";
2
3 const genAI = new GoogleGenerativeAI("API_KEY");
4 const model = genAI.getGenerativeModel({ model: "gemini-1.5-flash" });
```

JS JavaScript

```
1 const [loading, setLoading] = useState(false);
2 const [aiResponse, setResponse] = useState("");
3
4 async function aiRun(query) {
5   setLoading(true);
6
7   try {
8     const result = await model.generateContent(query);
9     const response = result.response;
10    const text = response.text();
11    setResponse(text);
12  } catch (error) {
13    setError(error);
14  } finally {
15    setLoading(false);
16  }
17 }
18
19 const handleGeminiQuery = (query) => {
20   setPromptButtonFirstQuestion(true);
21   aiRun(query);
22 };
```

JS JavaScript

Accessibility Enhancements with AI

To make our podcasts more accessible, subtitles were generated using OpenAI Whisper, a state-of-the-art automatic speech recognition (ASR) tool. Due to performance constraints, we opted for the small model instead of more resource-intensive alternatives.

Also, this document was collaboratively written with the assistance of a large language model (LLM).