# Prompt Engineering

We would like to acknowledge the support provided by Gemini, GPT -4o and Claude prompt engineering in resolving several critical development challenges. Although we used them all during the period of implementation, some of them were better in specific stages.

For the prototype, **Gemini** helped us the most, as it is especially good at medium-complex tasks. In the next stage, we used more **GPT -4o**, especially for the frontend, because we observed that it worked very well when it came to critical logic, code generation and debugging. GPT -4o helped us with styling, keeping the same aesthetic on all the pages, as it knew very well how to provide information about the CSS library that we had. When we were almost finished we switched more to **Claude**, because it excels in complex tasks and deep debugging. One thing we did using Claude was the automated tests.

# Bug 1: You couldn't buy a card from the SHOP

The "*buy a card*" button didn't trigger the desired JavaScript function because the event listener was not attached correctly and the buyerId used in buyCard() was incorrect. Claude helped us find the causes of this issue and repair it.

# Bug 2: You couldn't find friends

When you were trying to find or add friends via a **search or invite** system, nothing would happen or no results appeared even if the friend existed. We used Gemini to help us identify the problem and it turned out we had case sensitivity issues.

# Bug 3: You could add too many cards in the DECK

The app allowed users to add **the same card multiple times** even if they didn't have it and that violated deck-building rules. In this case, we used GPT-4o to repair this. It added a count (like an inventory check) so that you can use as many cards as you have and not more.