8-Ball Application Backend Documentation

How detailed did we explain your prompt to ChatGPT?

We did give it pretty detailed prompts. We made sure to include the vocabulary that we knew and to be explicit of what we wanted. The times we didn't was when we wanted to get something that we didn't know, like an animation feature moving left to right. There was a point where we wanted a bobbing feature and it gave us something decent enough and under our specifications. It was however too slow, but that had been because our input had given it a time for bobbing we thought would look ok but upon viewing the page was too slow for me. We also made sure to prompt it in steps.

How much of the code did we change for it to work?

We did not change it too much. The times we changed it the most was when we decided to edit the animation in the CSS ourselves because we weren't sure what we wanted it to look like and we directly exhausted the parameters to give us an animation we were satisfied with.

The second time we changed it significantly was when we could not figure out how to get a button to work without wanting to Google it ourselves. We just guessed what syntax was correct and even added some ids to the elements that it guessed into the HTML file. Finally, we asked it a question of how to add the feature we wanted from its normal mode, not its code. It gave us a view of how JavaScript and HTML interacted and how they viewed each other. Finally we put this together with some of the code that had already been generated into the JavaScript file to give me a much cleaner and concise code and function.

Did we find it difficult to use ChatGPT's code or was it was easy for you to navigate? (Elaborate)

We found it very easy. But to be fair we also explicitly told it to write in specific languages so we would know what its output would kind of look like. The only one we were somewhat confused about was when it wrote a long JavaScript function. But most of it was just trying to access the correct element, so we instead changed the description of the function to be directly accessed by the HTML itself instead of the JavaScript trying to correctly guess what button we wanted.

We do want to mention that we used 'shell_gpt'. It was easy enough because this way we could also write our prompts. Most of our workflow this time was something like this:

Write into 'prompt[1-9]'
`sgpt --code < <prompt[1-9]> >> <index.html|script.js|style.css>`

Then we just opened the file (if it wasn't already opened) and moved what we wanted or even just left it in as in CSS we were mostly appending to the file anyways. Most of the CSS was done without problems like this actually. If you read the prompts we gave it, you could see it was

mostly us telling it to add something to a specific element/to a group we wanted and seeing what it looked like. Then doing it again. This was our experience for the CSS at least. Similar for the HTML, but with a little more hiccups as it guessed a little more and added information that would be useful for the other two files to address the elements but we wasn't explicit about adding their ids or anything. Still an overall quick and efficient way of doing this basic material.