

In designing my to-do list, I didn't have a target audience in mind, but I aimed to make it simple and self-explanatory for what is presented. From the redesigning of the buttons, layout, and even color scheme of the project, as a means to make the page flow smoothly and have the user understand what the function of each button and presented option is there.

I wanted to make sure that when people looked at my page, that they wouldn't be searching how to use it. There's a large enough button on the screen that would draw people's attention into making their typed up tasks appear in the list. There are two lists, one to show active tasks and another to show history of completed tasks, that are distinguished through different sides of the screen. I designed the project so that once a task is complete, it will show in the history so that a user will be able to know what they have previously done or if they have to work on it again, they know what to type.

Of Schneiderman's 8 golden rules to interface design, I considered consistency of button functionality and a consistent look. I also considered informative feedback by having immediate response of adding an item to the list, or removing an item, and even the possible error of adding an empty task to the list which also pertains to dialogue yielding closure and simple error handling. The addition of the task history helps reducing short term memory by presenting all the info to the user, even the data that isn't useful anymore, which also supports reversal of action by allowing the user to retype a task word for word.

The interface is simple and easy to use and the colors make it visually appealing. If i were to change anything, I'd change allow the task submission to scroll down with the screen.

One affordance is the "add to list" button. It has a true affordance because it says it'll add to the list, leading the user to assume it adds to the list, which it does, proving true . It is visible due to the button having text and being directly beside the input.

One constraint is that when marking a task as complete, the user is unable to remove the task from the history. This is made clear by the complete button being removed from the task while in the history list.