A problem I have had in my life was that I wasn’t typing as fast as I would have liked to and it was slowing down my productivity when writing papers. To start this process, I switched from typing in the standard QWERTY keyboard layout in favor of the Dvorak layout because it is much more efficient and better on the fingers, but since I had to learn this new muscle memory, I need a means to train. My solution to this was to create my own typing trainer that allowed for me to increase the accuracy and speed of my keystrokes.

To create the main functionality of this project, I used an object-oriented approach because I thought it would be best to treat each letter in the text block as an object to keep track of the correctness and visual properties all in one place that would be easy to keep track of. I used JavaScript to implement this because it is a language that I am comfortable with because I used JavaScript and the NodeJS framework very frequently at my Co-op job, however I treated it more as a functional language than an object-oriented language at work. While working on this project, I learned a decent amount about how to use JavaScript to implement objects because it is a bit unorthodox in its implementation compared to some languages that are a bit more structured, like C++ or Java, for example. Learning to use the “this” property of a class in JavaScript is somewhat difficult, but it seems to work roughly as a public method or variable, and whenever an object that belongs to that object is used within the class, you must use “this.variable,” whereas variables defined with “let” or “const” can be used throughout the class without “this.” But are not accessible from the outside directly. I really like working in the object-oriented paradigm because everything is easier to keep track of and test because it is very organized in implementation. You can define helper functions within the class without making them publicly accessible, so it’s very easy to know which attributes of the class you can see when working elsewhere in the codebase. This gets to be especially helpful when working in very large code bases because you can place one class per file and organize the files with namespaces, or other similar organization methods.