For the regular hearing test we encountered issues with getting all of the audio files to import properly and then having the test load the next audio file after a user clicks submit. Since we were using React for the application, the test is written as a functional component. To solve the load issue audio file links are imported from an object from a separate Javascript file. This object can be indexed by the current selected language (stored as a variable) and the number of the current audio file. The audio file name is then set as the source for an audio component on the page and linked to the play button. Loading the audio files this way makes sure that every word for each language would load at the right time. We also encountered a couple bugs which prevented the page from updating the audio file after a user clicks submit. These were solved by adding keys to the icon image tags so that React would not confuse the components and update at the wrong time, and by using React's built in effect hook with useEffect() to tell the page to update when specific variables change. The final major problem we encountered was that the page would not reset the test properly at the end. This is because the audio links are stored in an array and removed as they are used, and when the test restarted it would not return all of the links to the array. This was fixed by resetting all of the other variables to default values and making a full copy of the array of links which could be copied back into the main array at the end of the test. We had similar issues with loading the audio for the frequency hearing test. The same basic structure of importing the audio links from a separate Javascript object was use. The issue here turned out to be that there was a new variable introduced in the frequency test which had not been referenced by useEffect() , so when it changed the page would not update. All of these bugs have been resolved and the tests are working seamlessly now.