**README THINGS:**

You’ll want to run init.php before you do anything with the site. It creates the user database and the site will be more useless if you can’t log in. That’s the other thing. If you don’t log in you can’t do certain things like view your times or upload splits. I’ve also included a few split files for you to try uploading.

**WHAT CHANGED:**

By and large the entire site changed, both in look and in content. The first implementation was really only a barebones, hey you can create an account on this website, lacking even proper login functionality. The current version includes the ability to create an account, log in, add games to a list of games (database), upload split files, and check the times that you’ve submitted with these splits. So, in short, in part two I vastly expanded the database functionality of the site.

**PART 3?**

There are many features that I intended to implement, but either due to time constraints or lack of expertise I was unable to do so. First and foremost, I wanted to be able to add a full set of splits to a user’s individual game table so that they could compare their times with others. Unfortunately, I was unable to do so because I could not implement dynamic table growth during split submission, and performing these column insertions by hand would have been excessively inconvenient. I also wanted to, when a given game from the list is clicked, display each user and their submitted times for that game, however I was unable to get the jQuery/Javascript to play nice with HTML elements generated with PHP. I would also want to style the site more. Perhaps one of my issues is working with such large windows, but everything feels quite bare, and with that in mind I might want to consolidate several of the pages into more efficient bundles somehow, which would increase aesthetic appeal and usability in my opinion.

**WHAT I LEARNED:**

My basic understanding of XML and database manipulation alone grew immensely. I also got much better at linking documents together with includes and such to create more complex pages whose code still appeared relatively clean (this wasn’t true for all pages/cases, but for instance the home page other than the ugly list is rather clean relative to my initial efforts). Generally I would say that I became more competent overall, but learned more about what skills I still lack with regard to database access, and event driven changes to the site.

**WHAT I LIKED**

I enjoyed the autonomy that we were provided. You could have very easily made the project an assignment like a much larger homework (perhaps giving a choice among a few options), but let us choose our own direction. This might have in some cases like my own led to projects that were somewhat more complex than the student could entirely handle, but that experience in and of itself is valuable, especially in the context of a practically applied skill such as web development. I think that the only ways it could have been better are all on my end. If I could have devoted more time and effort to it I’m very aware that the roadblocks I encountered during feature implementation could have been surmounted.