

Code Review

Throughout the process of writing the code, we endeavoured to keep our code as clean as possible. We made sure that all our code adhered to the clean code tips presented in class. Also, we used proper naming conventions in all our code. We had variable names like “li” and “ul” which are not typically good variable names but we could not get rid of them because they are built in tag names. All the commented code was also taken out in order to improve readability. We also set up our variables/code in a way that allows for logical read throughout the code. This is beneficial because it will be easier to make changes as we continue on with our milestones.

To ensure we had great architecture, we made our files loosely coupled. Our `validate_signup.js` and `validate_login.js` files handle the verification of our signup and login pages respectively. We also made sure our CSS and HTML files run independently. This is to make sure when changes are made to one file it does not affect another.

We changed the `signup.html` and `login.html` to php files to be able to access the database. We also merged all the css files for our video pages into one css file.

The hardcoding that we had in our last MVP is now completely taken care of as we have a functioning database now. So a user that signs up on the website will be stored into the Hygge database and will be retrieved when a user logs in.

Since our code is set up in a logical manner and it is commented appropriately, this makes it easy to read, test and debug. Our code is pretty extensive now, because all the new functionalities are added to our website. Our code can only handle limited security risks as we only have our signup and login functionality set up in the database. Our current website is not equipped to handle a Denial of Service (DOS) attack or any major security risks. This would be a task for future milestones, and an important one at that.