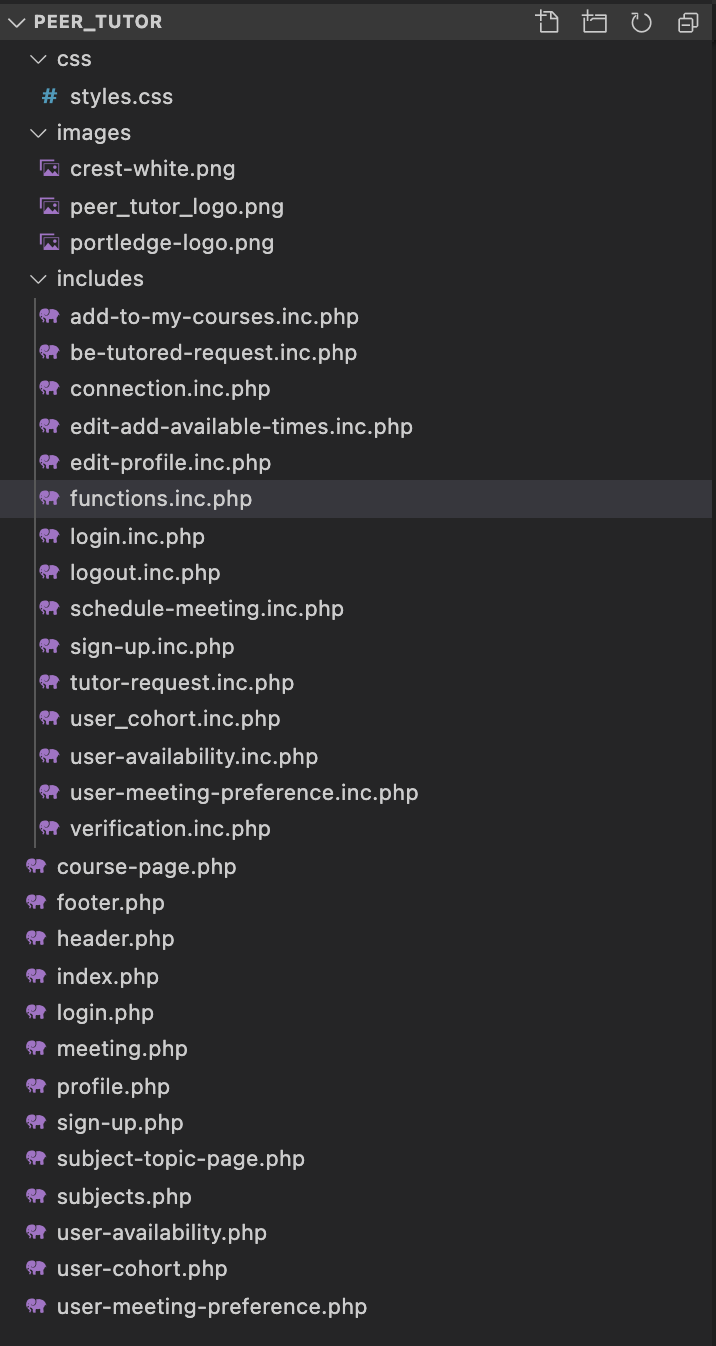
## Criterion E: Product development

## Complex techniques used to address the client’s requirements

* Arrays, including appending to $\_POST, $\_GET, and $\_SESSION
* Subroutines, parameter passing
* Structures Query Language (SQL) to develop back-end database
* Cascading style sheets (CSS) or schema

## Organisation of website (Document object model)

This page outlines the internal file structure of the website



I have separated the css and the images for the site in their own files. This allows for easy access when adding more files to the codebase.

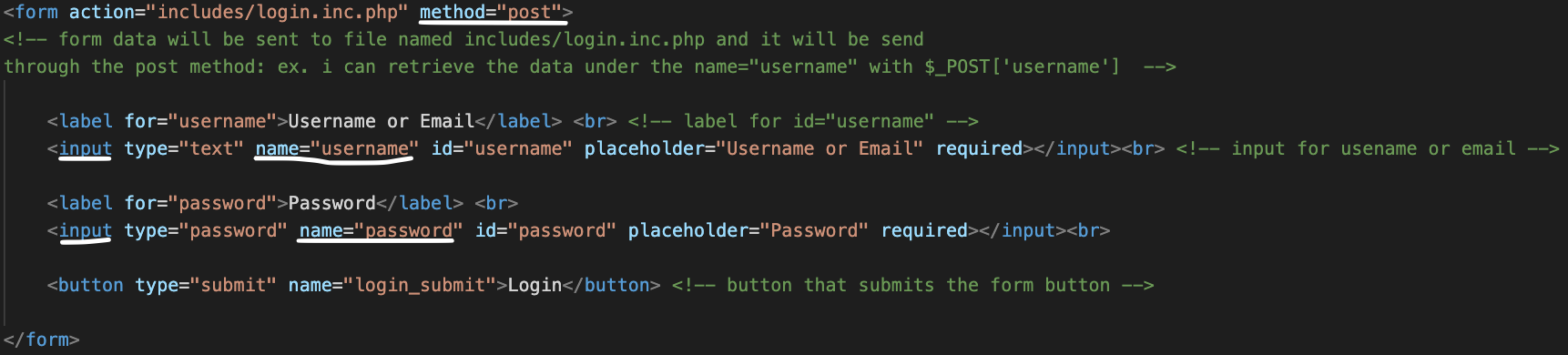
When using php, I am constantly dealing with forms. To input data when using the website, I used forms, and when you submit the form, the data has to be sent to somewhere. I decided that my form data would be sent to a file with the same name as where the form originates, plus a “.inc” extension. This extension does not affect any code in the file, but just helps me stay organized with where my forms are, and where that form data is.

For example, when logging in, there is a form in the file “login.php”, and the data from the login form gets sent to the file “includes/login.inc.php”.

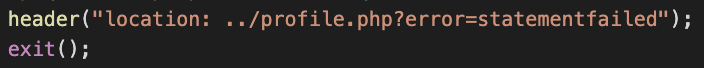
## Technique: Arrays, including appending to $\_POST, $\_GET, and $\_SESSION

In php, you can access data in any file through superglobals. Superglobals are built in php lists that you can append data to. The three different superglobals I used were $\_POST, $\_GET, and $\_SESSION.

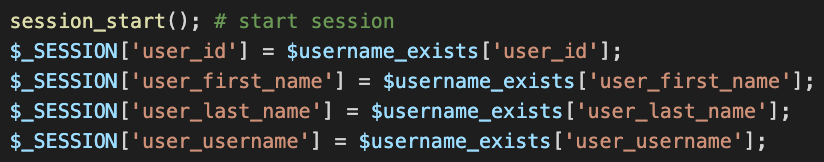
$\_POST:

The $\_POST superglobal was used to collect data from forms. In my website, most of the input is handled through forms, and when creating a form and setting the “method” to “post”, you tell php that you want to append the input data to the superglobal $\_POST. Then, when collecting the data, you can access it by calling $\_POST[‘(name of input\*)’]

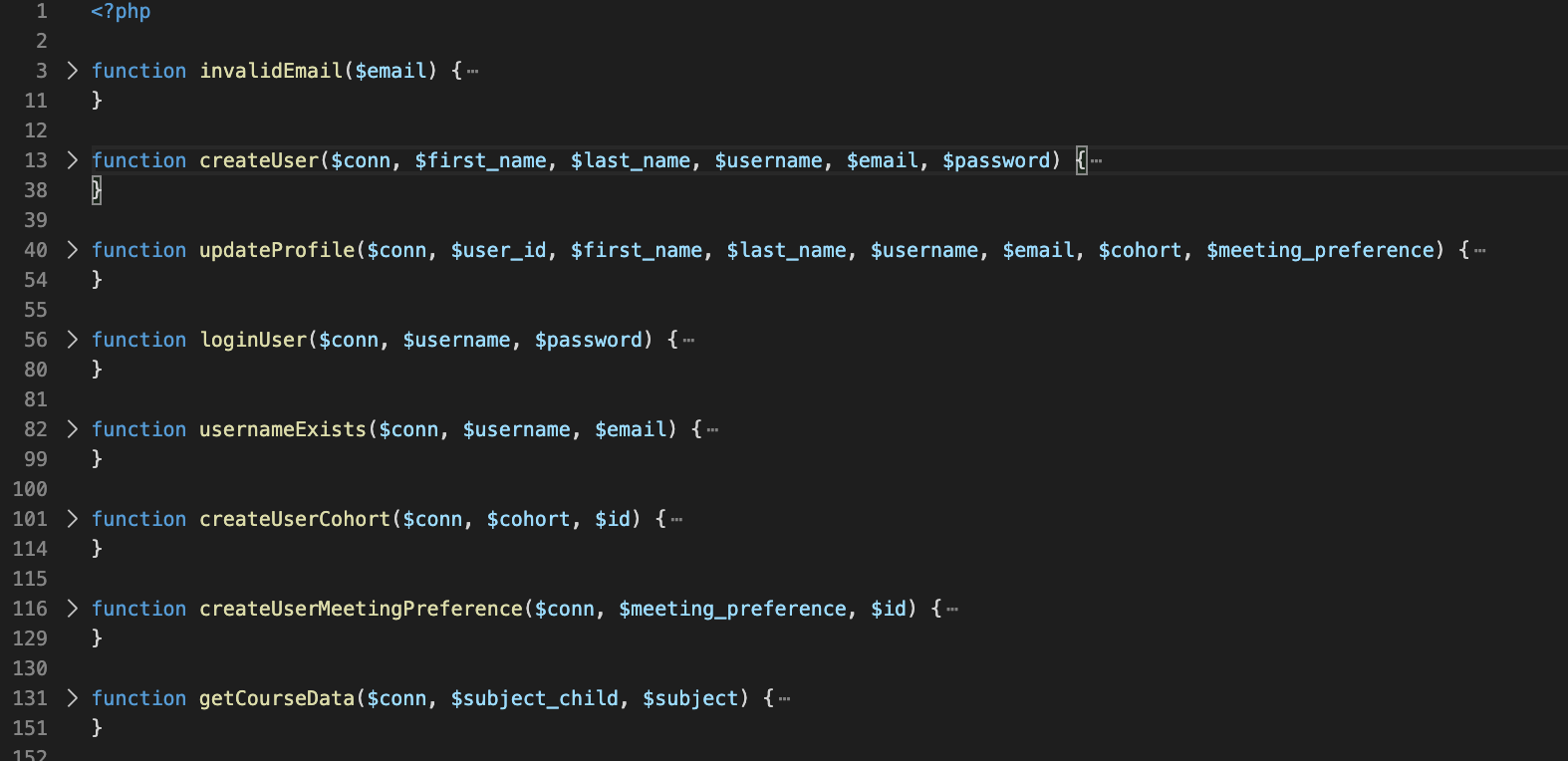
$\_GET:

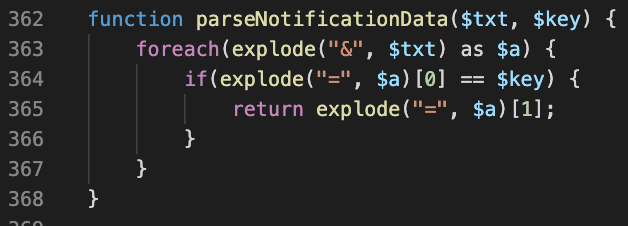
The $\_GET variable is used when you do not need the data to be private. This is because $\_GET variables appear in the url of the site. When appending data to $\_GET, all you need to do is in the “action” part of the form, add variables in the following format: “?variableA=valueA&variableB=valueB&variableC=valueC” etc. You can also do this with the header() function; I used this to send the user back to a certain page after completing a function.

$\_SESSION:

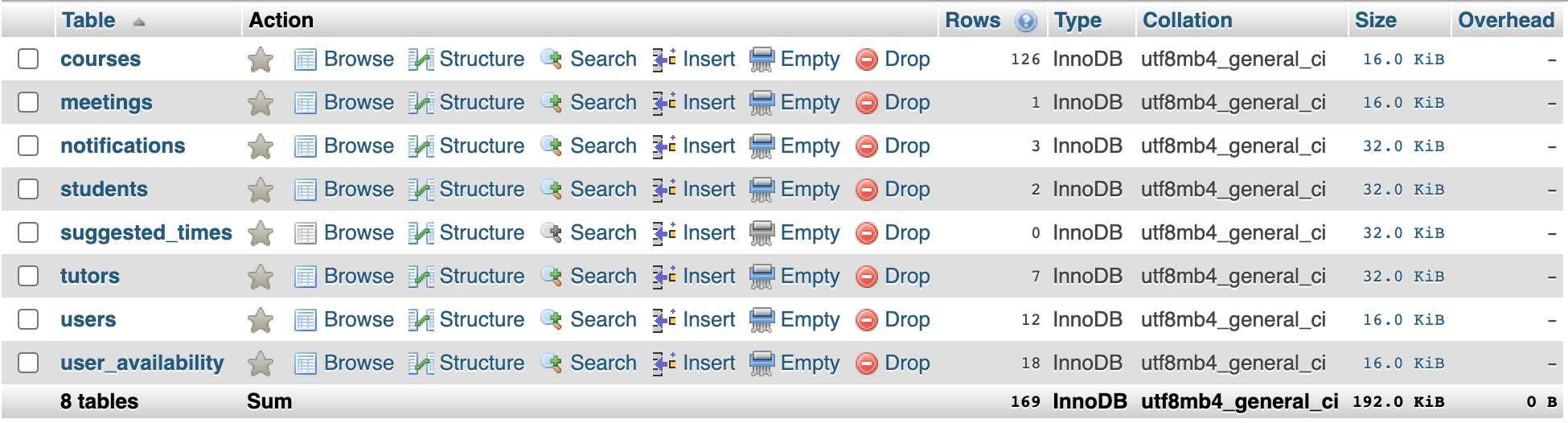
The $\_SESSION variable is used for information to remember who is using the website. Data stores here would contain the users id, name, and username. I created the $\_SESSION variables once the user logs in or signs up. To remember these variables across the different pages, you have to call the session\_start() function.

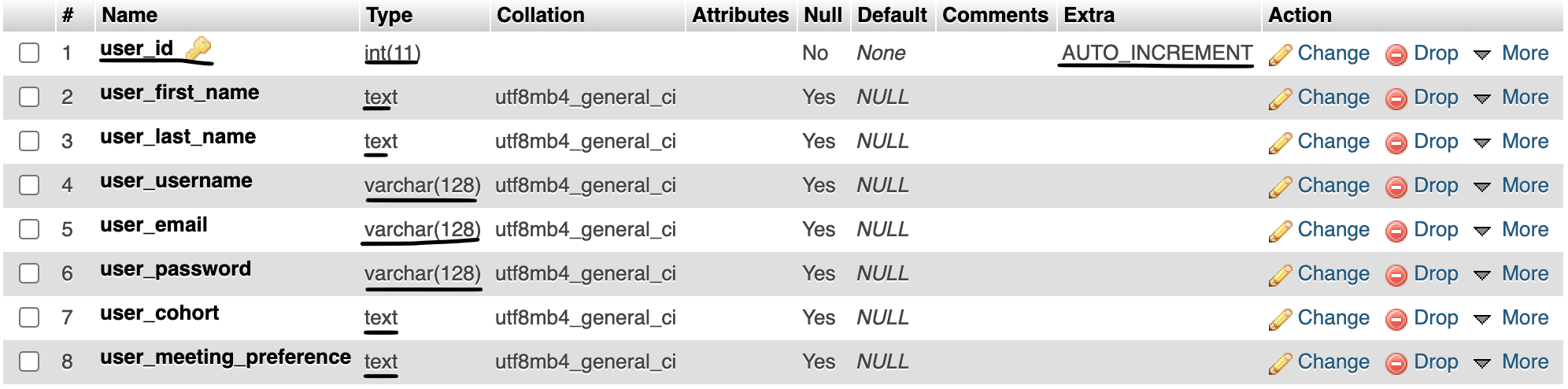
## Technique: Subroutines, parameter passing

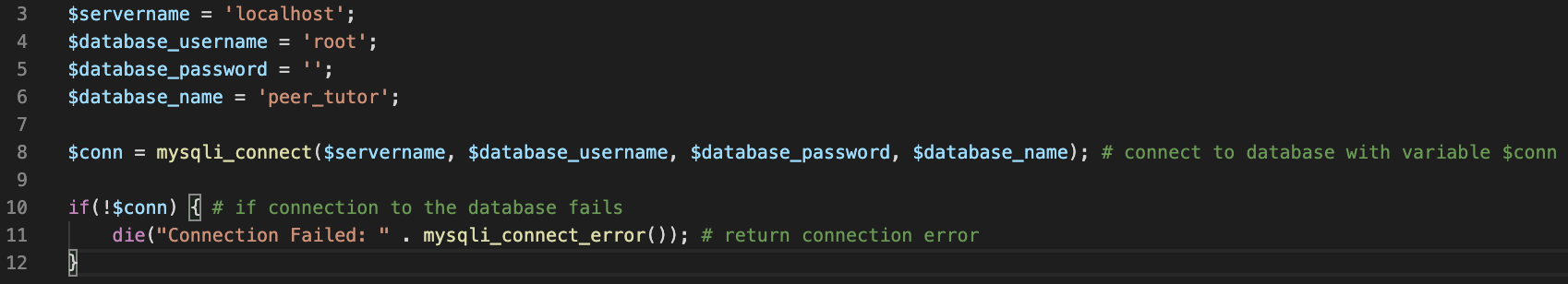
To make the process of running the same code multiple times more efficient, I used subroutines. My functions were kept in my “functions.inc.php” file on the website.

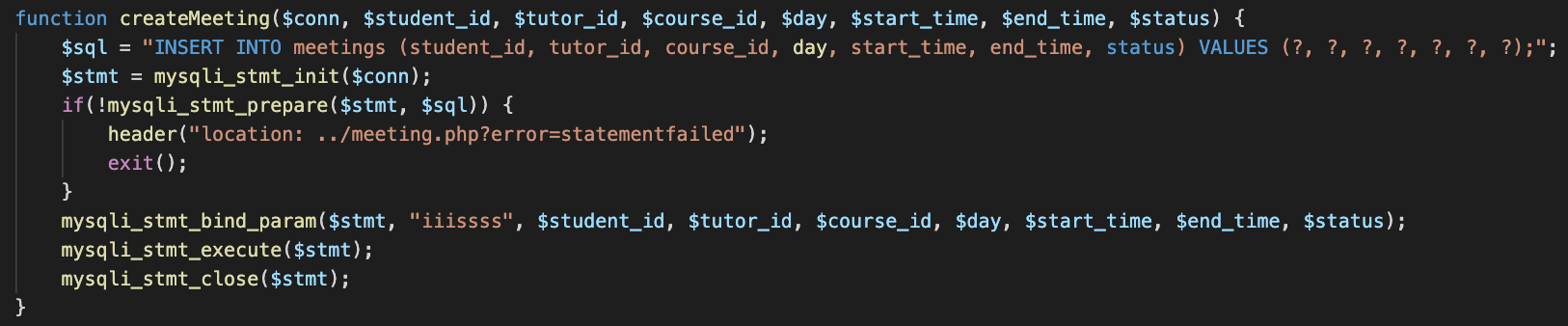
These subroutines had parameters that were passed into them so they could manipulate it. These arguments were specific to the function of the subroutine.

## Technique: Structures Query Language (SQL) to develop back-end database

To retain the data the user inputs into the website, I needed to connect the php to a database, using SQL. I made different tables in my database, each with a different purpose for the site. The was a table for the users, tutors, students, meetings, and user\_availablity are some examples.

Within each table there is an automated id to assign each cell in the table a unique value. Also, each cell has a specific data type: int, text, varchar, etc.

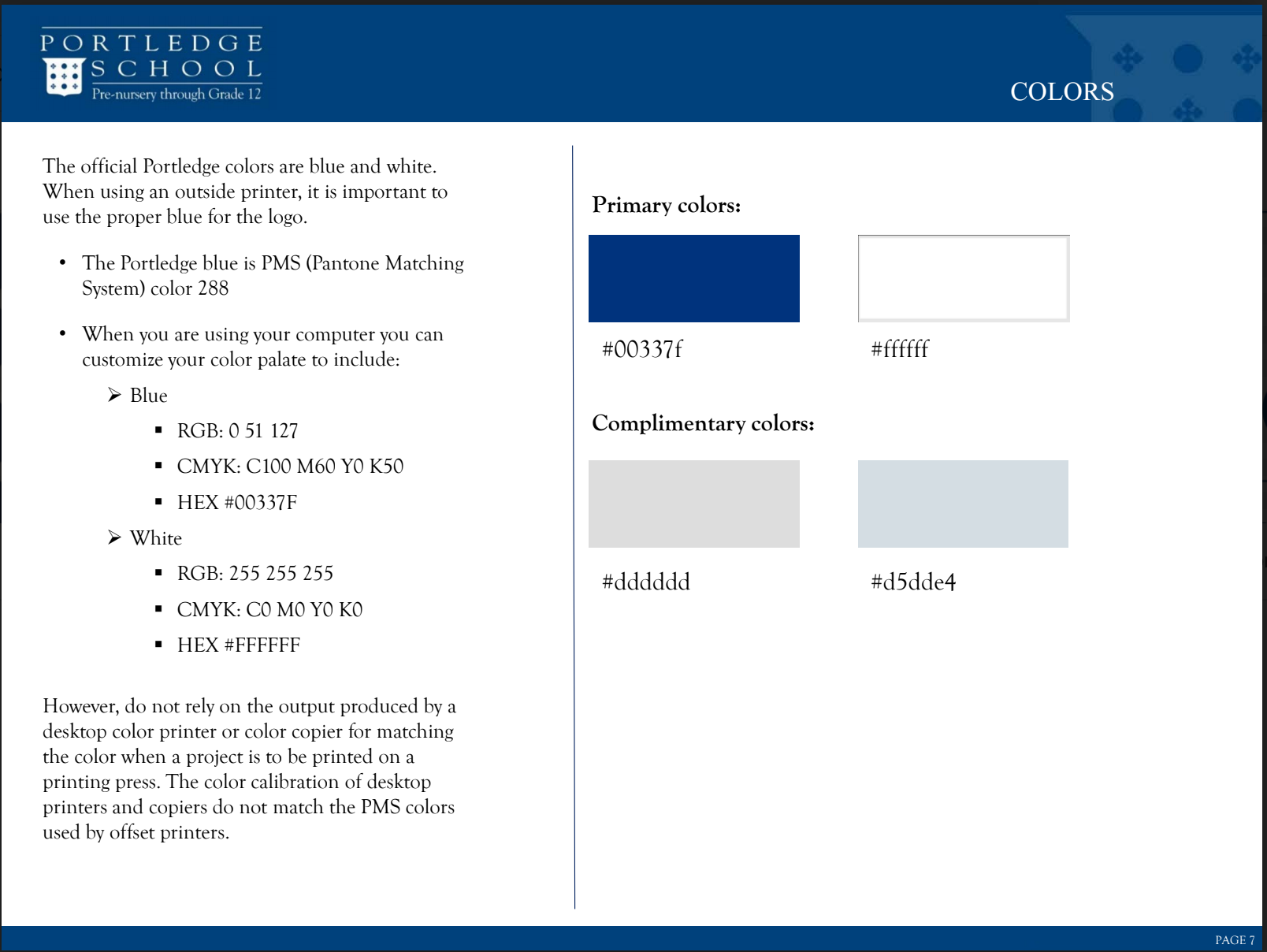
To connect the php to the database, I used php functions that are specific to SQL, such as mysqli\_connect().

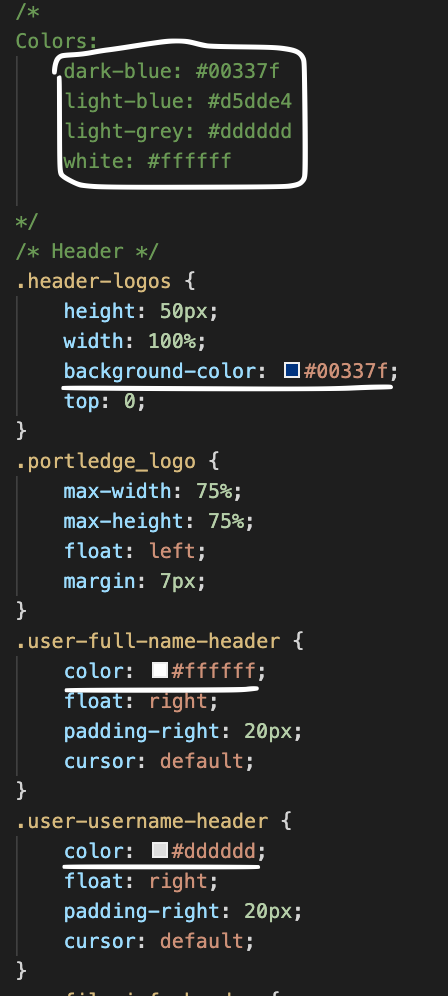
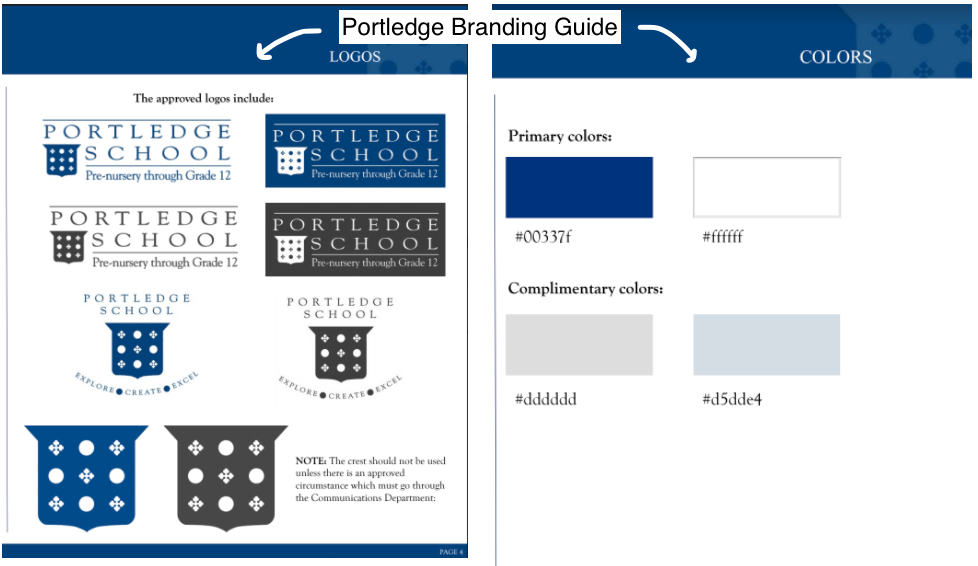
Then, when having to input data into the certain tables, I used SQL specific functions such as INSERT, DELETE, UPDATE, and SELECT.

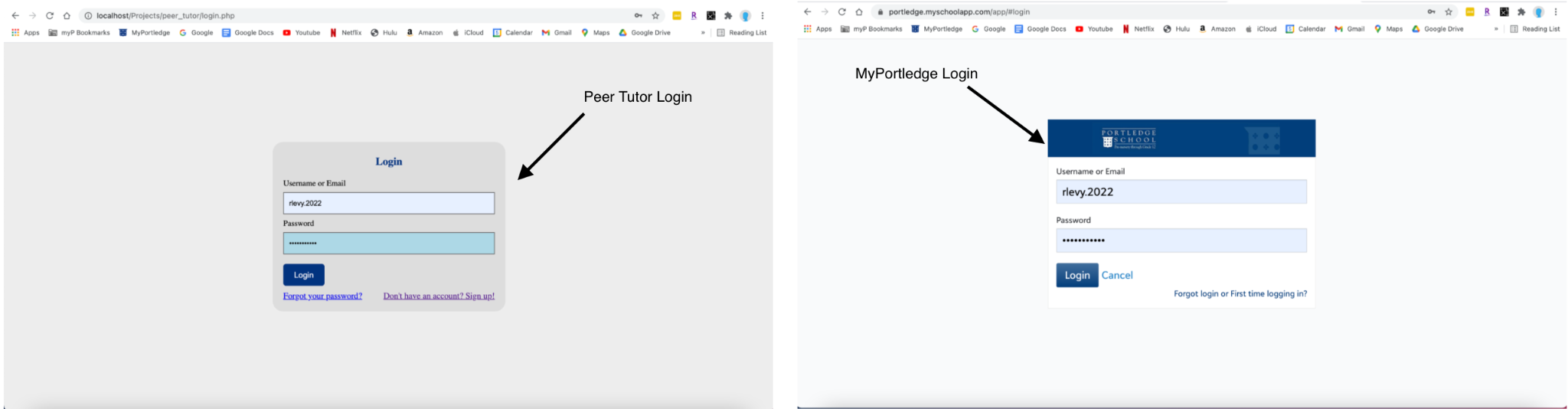
This included php functions such as mysqli\_stmt\_init() to initialize the SQL statement, mysqli\_stmt\_prepare() to test to see if the SQL syntax correct, mysqli\_stmt\_bind\_param() to append the data I wanted to add to the table, mysqli\_stmt\_execute() to run the function, and mysqli\_stmt\_close() to close the function once it completed.

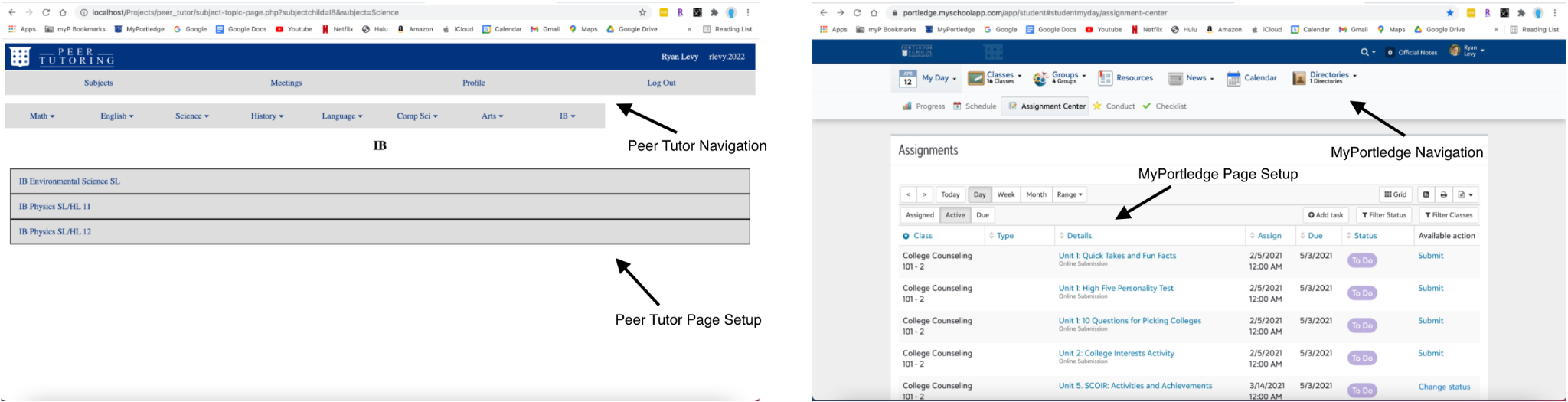
## Technique: Cascading style sheets (CSS) or schema

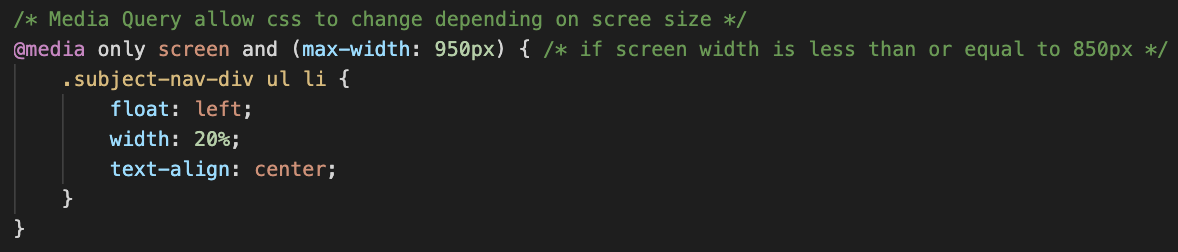
To add style to my site, I used cascading style sheets. I had one main file called “styles.css” that was included in most files, but in some cases the css was embedded on the file itself, using the <style></style> html tag. Sometimes the css was not updating on certain pages and I could not figure out why. When I re-created the css on the file instead of on a separate file, it worked. This was a way to solve some bugs I experienced with the css.

My css style was close in theme to other Portledge websites, since my client’s website is for Portledge School. I used the Portlege Branding files to get access to the official Portledge logo and color scheme. I also looked at websites such as portledge.myschoolapp.com for inspiration on general style (on navigation and overall look).







Some advanced css techniques I used were media queries to change the css of certain elements when the screen changes size. Media Queries are css properties that allow you to dynamically change elements when certain conditions are met. For this media query, I checked the users screen size and if iis width was 950px or less, the navigation width changed to 20%. This allowed for the navigation of the subjects to change from one row to two rows.

In my css, I needed to create naming conventions that would be constant throughout development. I created my html element classes to fit the name of what the element’s function was, to then easily refer to it in the css file. For example, “subject-nav-div” referred to the subject navigation divider, and the “path-student” and “path-tutor” class names referred to the tutor and student paths the user could take when on a class page.

Word Count: 977

Bibliography

* Portledge School, Brand & Visual Style Guide, 2016-2017
* *Portledge.myschoolapp.com*, portledge.myschoolapp.com/app/student#studentmyday/assignment-center.
* contributors, phpMyAdmin. *PhpMyAdmin*, www.phpmyadmin.net/.