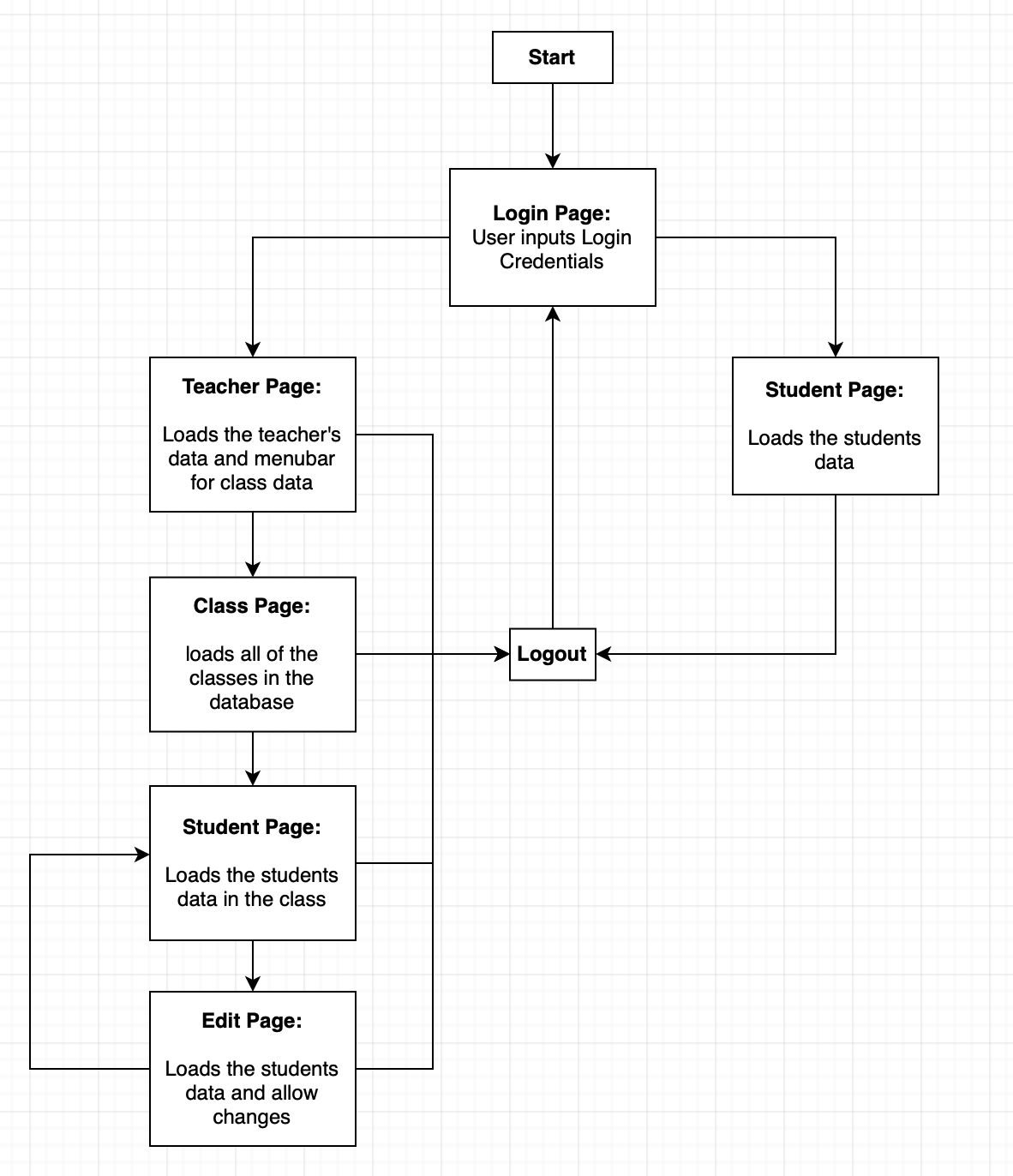
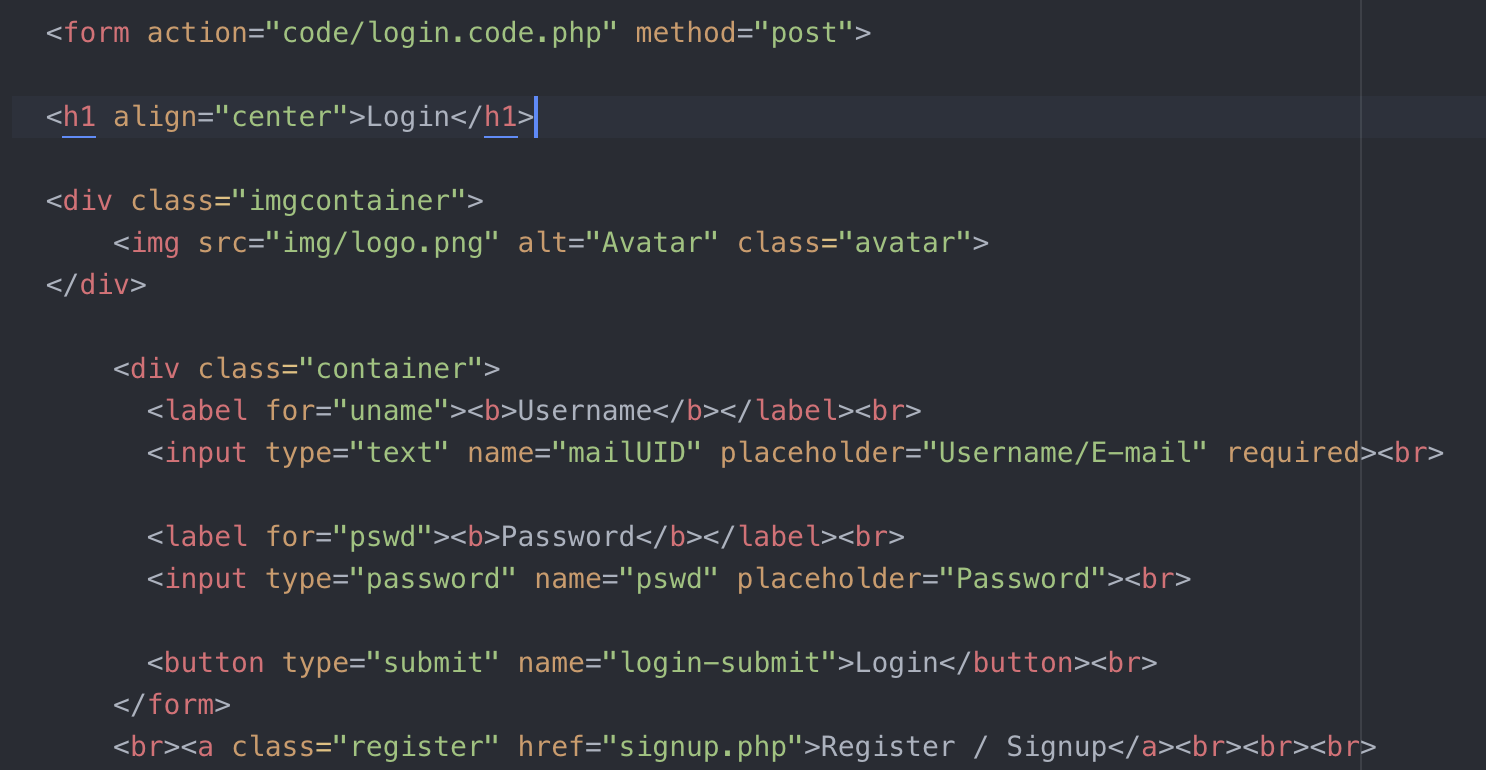
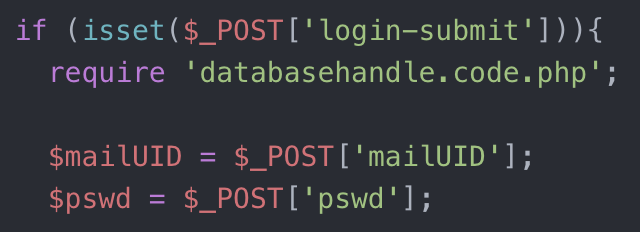
## Criterion C: Development

The program is made to act as a database of the students present in the school and as a disciplinary system. The database will be doing double duty to store the student’s and teacher’s basic data and the disciplinary points. The program is made using 3 languages, which are PHP, HTML, and CSS. The software used are Atom, XAMPP and PHPMyAdmin. These 3 software help me handle the codes, UI and the database, which is MySQL.

1. **Navigation of the system.**
2. **UI Code / PHP Code**
3. **Front page**

^ Figure 2.1 – HTML code

The Code above is to provide the user input and connect to the php file. It also give navigation to the register page where the user is able to register a new account.



^ Figure 2.2 – PHP code, checker

Responsible to make sure that the user is not able to access the file through searching it in the search bar and is present throughout the code. This also stores the user input of the login credentials.

if (empty($mailUID) ||empty($pswd)) {header("Location:../index.php?error=emptyfields&mailUID=".$mailUID); exit();}

The Code above checks whether the input field is empty.



^ Figure 2.3.1 – PHP code / Login

This part of the code checks with the database, the statement is prepared and checked whether the code is able to connect to MySQL. after it is able to connect, it would get the result of the search from the statement. The result would then be used as the row and find the password and is verified with the password inputted in the login page. If it’s wrong then it would reload the page.



^Figure 2.3.2 – PHP Login

This part of the code would be after the password check when its correct. When it is correct, the program would make a session in which the user is Logged In. the program would then checks the account’s type, teacher or student, and would direct it to the correct page. This would signal as a successful login.

else{ header("Location: ../index.php"); exit(); }

If the account is not registered in the database, the program would reload the page.

1. **Register page**



^Figure 2.4.1 – HTML Code / Register page.

This code is used to connect the page to the corresponding php code file. In this case, it’s the signup code to be able to register a new account. When a button is clicked the code is activated / run.



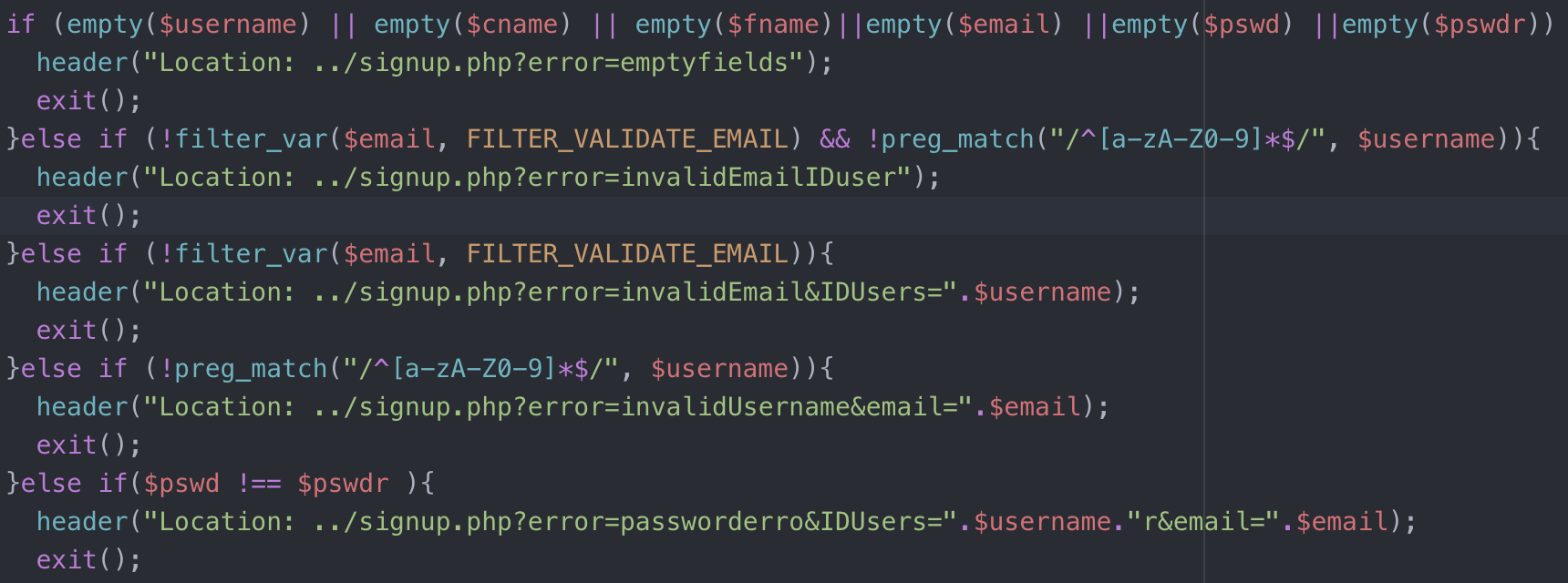
^Figure 2.4.2 – HTML Code

This code is used for inputting the new account’s information. This is repeated for other information such as the date of birth, class ID.



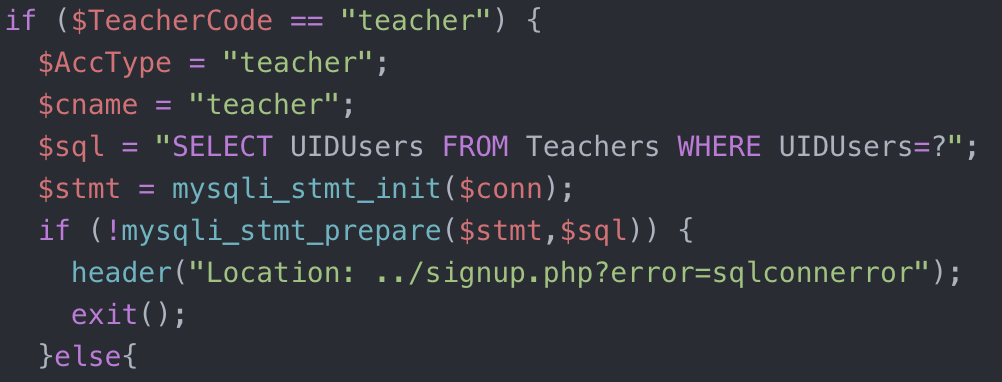
^Figure 2.4.3 – HTML Code

This is similar to figure 2.4.2 but the input type is changed to password for security as the information inputted is the password and the special code to change the account type to a teacher.



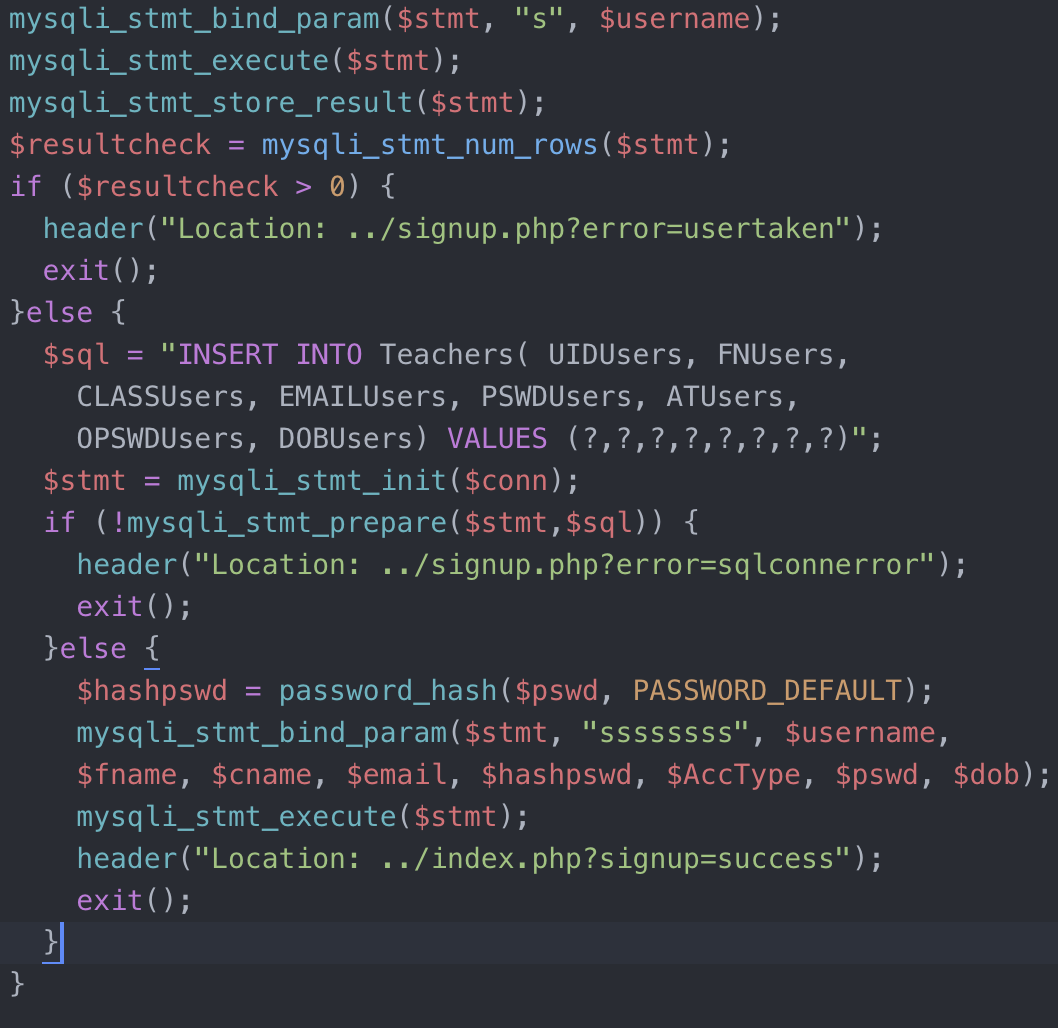
^Figure 2.5 – Error Handling

Error handlings that is present during the signup process. Check whether the important fields are empty



^Figure 2.6 – Establishing connection with database

Establish a connection with the corresponding database whether the code is given to make the account a teacher or a student.



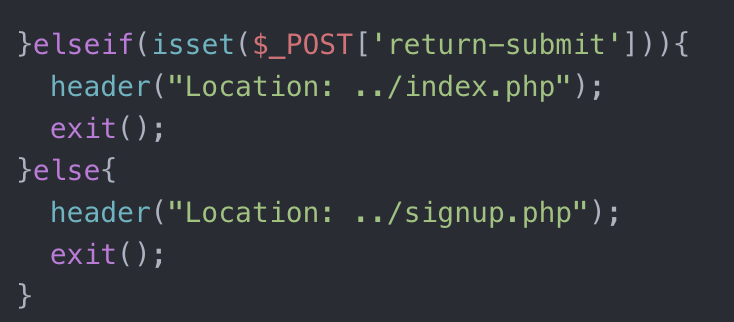
^Figure 2.7 – Insert into the database

The program firstly checks whether the username is taken, this would minimize overlapping usernames and would reloads the page if the username is taken. It then creates a new row and fill the rows with the corresponding data to the database. This is also done for the students as well, not just teachers. It also hashes the password. The OPSWDUsers data is used during testing as this stores the unhashed password in case original password is forgotten.

Hash password

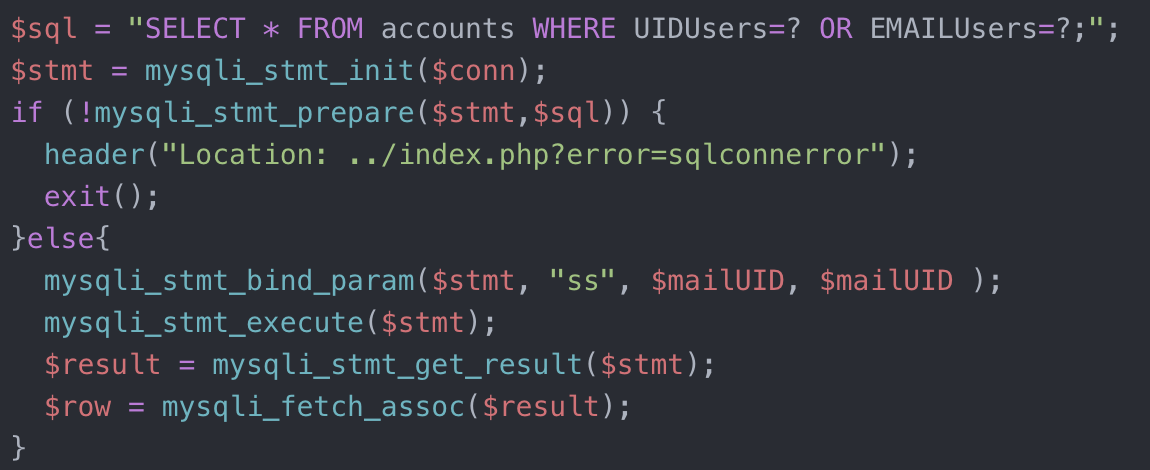
$hashpswd = password\_hash($pswd, PASSWORD\_DEFAULT);

This is used to hash the password so that the user who is able to see the database is not able to understand the password. This would increase the security of the program.



^Figure 2.8 – return button

This is used for the user to be able to return from the register page without registering a new account.

1. **User’s Main page.**

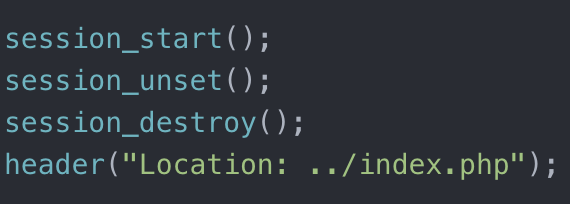
^Figure 2.9 – Fetch user details

this part is used to fetch the data from the database concerning the user. This was used in the login part but this is done for a different reason, that is to display the information.

^ Figure 2.10 – sliding navigation bar ( HTML code )

Used for an animated navigation bar in the teacher page to get to other pages. This is only available on the teacher’s page and throughout the pages for the teacher. the animation slides the bar from the left ( unseen ) to the right ( seen / visible ).

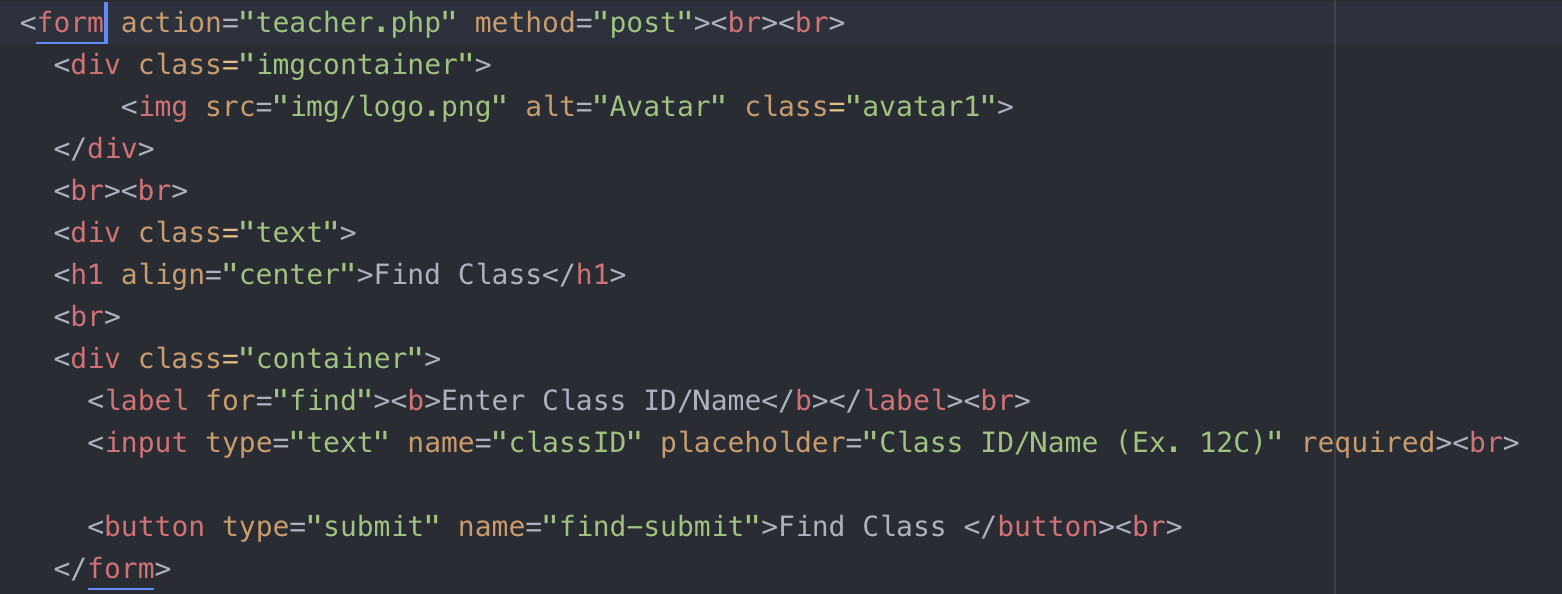
<a href="code/logout.code.php">Logout</a>

This is the code used for the user to log out of the account and go back to the main page.

^Figure 2.11 – logout code

Used to destroy the session thus logging out the user from the program. The user is then directed to the main page / login page.

1. **Search & Student Page**

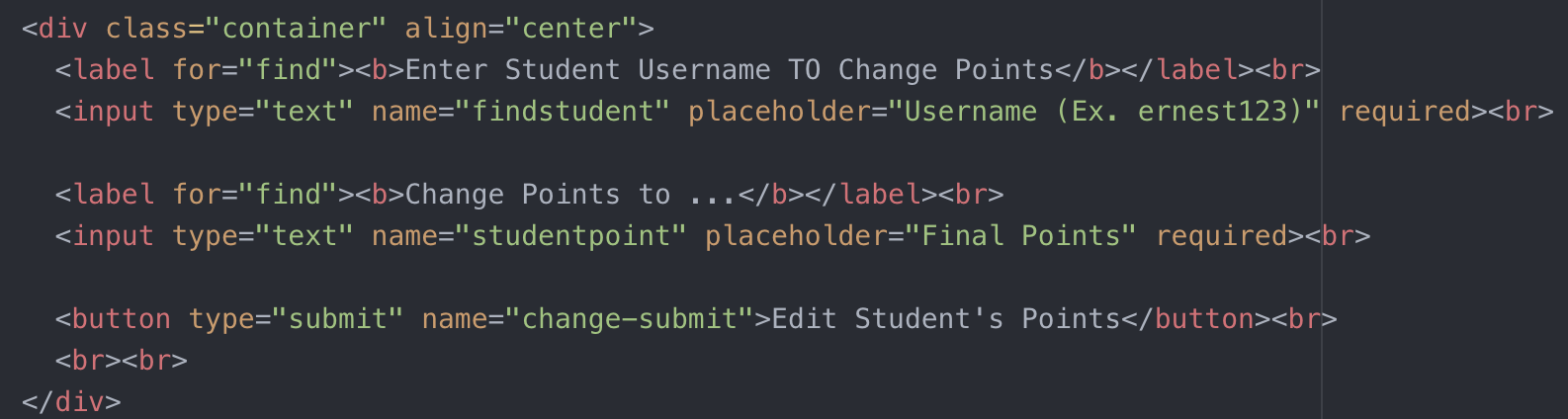
^ Figure 2.12 – find class input

Used for the user to input the class id to find the class and output the students registered. This is connected to the teacher.php, when the button is clicked the user is directed to the other page and the searched class is fetched from the database.

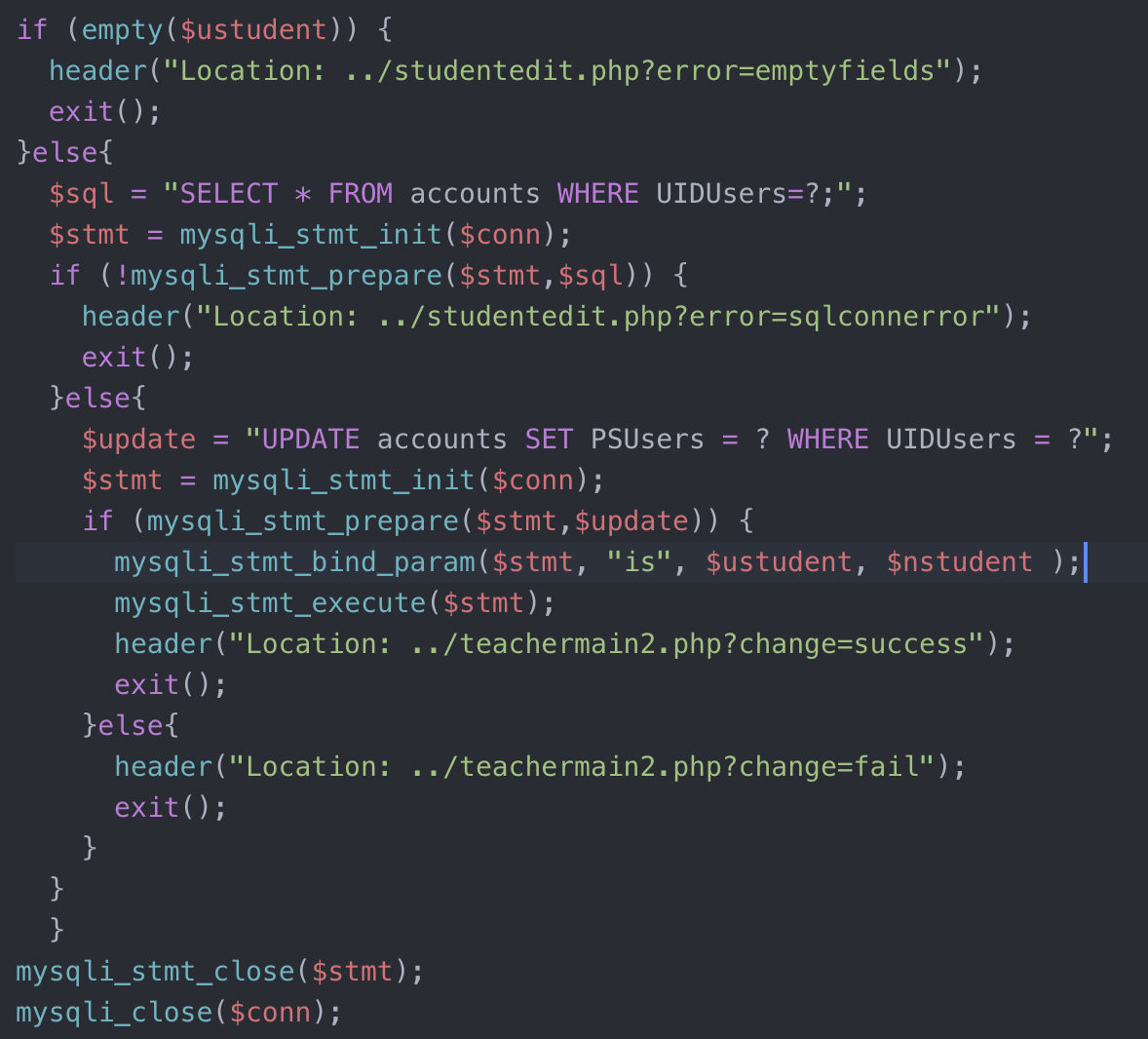


^Figure 2.13 – table for Students

This is a code used to display the student’s data in table form. It shows the student’s ID, Full name, Username, Date of Birth and Disciplinary points.



^ Figure 2.14 – User input to edit the students points.

^ Figure 3.15 – edit students points

This is used to edit the students points. It finds the student’s account and overwrites the points using an Update Query. After successfully changing the points it would later jump the search page.

Word count: 804