**Criterion C: Development**

**Overview:**

This product is a linkage between MySQL database, HTML files and PHP files. The basic structure is linkage between PHP files and HTML files, almost every page has a PHP file. However, there are pages only utilize PHP file or HTML file. The main structure of the product is the linkage between different pages, which creates the website.

Although numerous techniques are used in my product, limited by the space, I will only show the main techniques.

**List of main techniques used:**

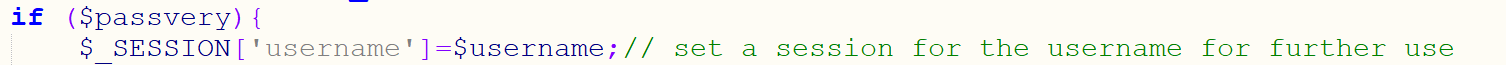
1. User System
   1. Session for checking the status of logging in
   2. Database for storing the users’ information
   3. Regular expression to check the form of email address
   4. Password hashing
   5. Update the data in the database
   6. The use of verification code
2. Functions
   1. API for cat face recognition and translation
   2. Retrieve from Database and show in the main viewing page
   3. Insert posts information into the database
   4. Searching and sorting in the database and show the results
3. Interfaces
   1. CSS and HTML arranging
   2. Use of JavaScript for pop-up windows

**Part 1: User System**

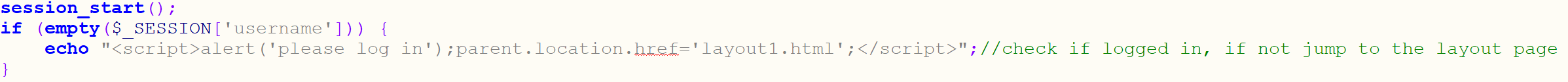
1. **Session for checking the status of logging in**

In the user system, there are lots of position that need to check whether the user is logged in or not, including posting and viewing the profile page to modify. Therefore, session is needed to achieve this function.

When a user is successfully logged in, the session is started and the variable inside the session, “$\_SESSION[‘username’]” is stored as the username of the user.

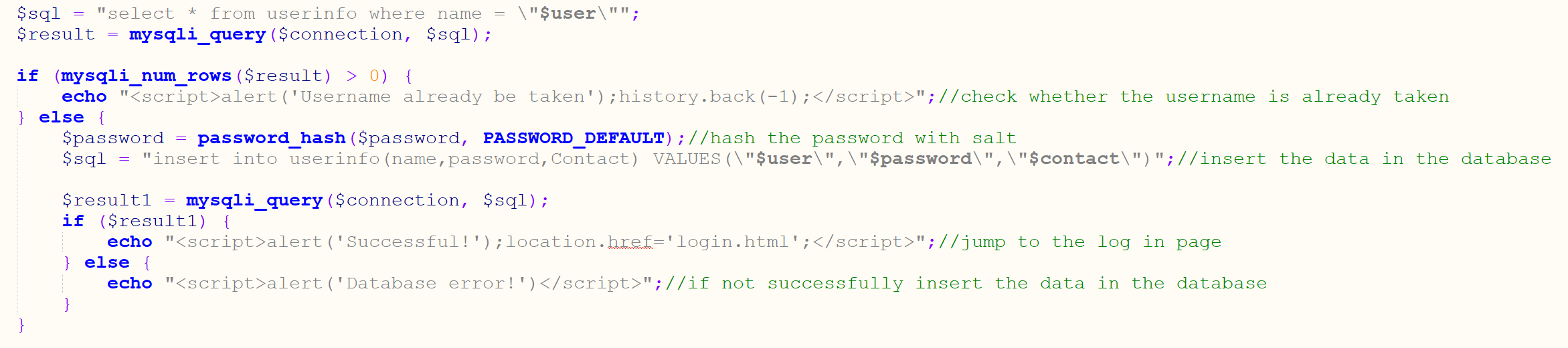


After that, when there is need to check whether the user is logged in or not, I can just utilize a selective structure to check whether the variable is empty or not, as shown in the code below.



1. **Database for storing the users’ information**

The users’ information is required to be stored when they registered as it will be used when logging in to check their username and password and when posting to add their contact information in the posts. The command, insert, is needed. The following code is used to add information into the table Userinfo using the values “$user”, “$password” and “$contact” gained from POST from the html file. The database is modified by PHP MySQLi extension.

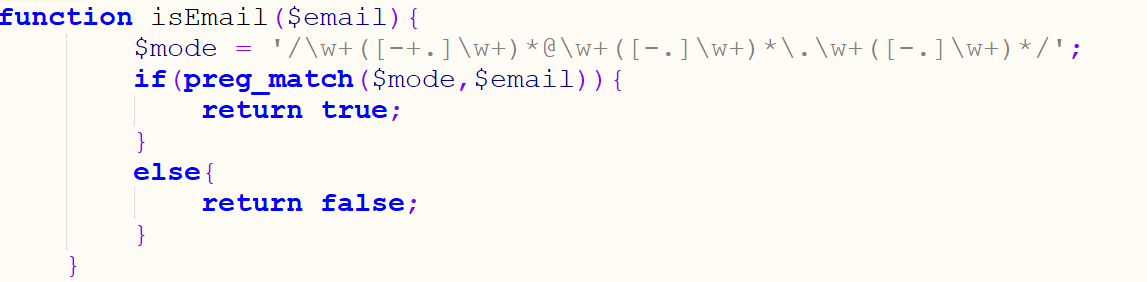


This is the part of sending data from the HTML file.



1. **Regular expression to check the form of email address**

I utilized regular expression to check the form of email address to make sure the email address inputted by the user in its form is valid. This is the PHP code for the function of checking the email address.



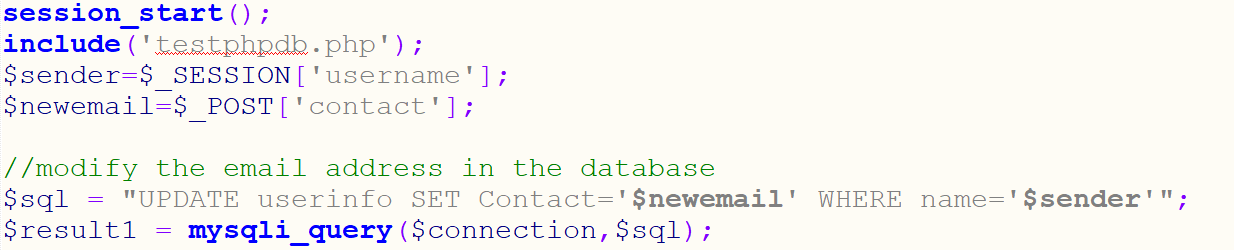
1. **Password hashing**

To make sure that even though the database is lost, the password of the users is still safe, I utilized hash algorithm, bcrypt, to encrypt the password in the database.



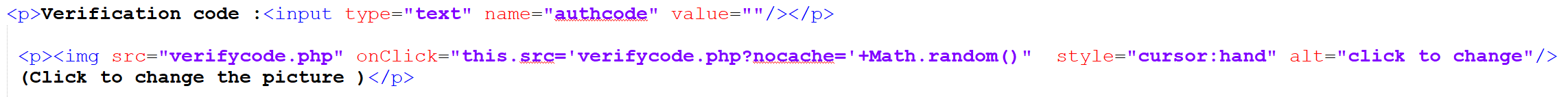
1. **Update the data in the database**

When the users modify their contact information, there is need to update it in the database. Thus, I utilized the command, update. I gain the username from the session and the new contact information from POST.

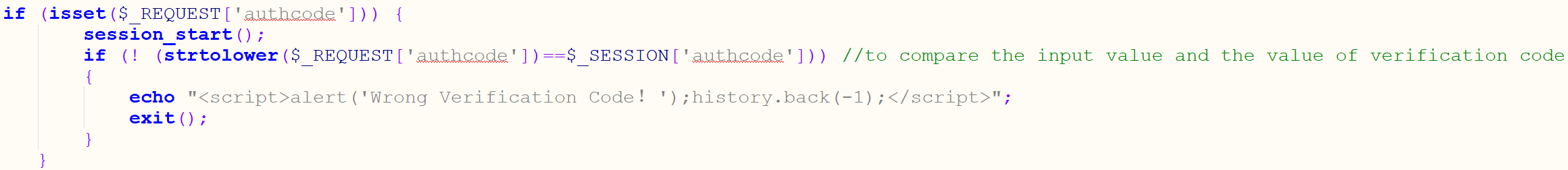
****

1. **The use of verification code**

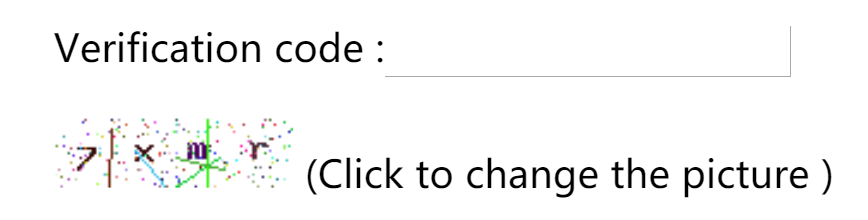
I utilized the code I gained from ‘bubuko’ to generate verification code. This is used to prevent automatic request to register by brute force. The verification code is composed of random generated colored lines, dots and letters. The file, verify code.php, can help to generate verification codes by employing the following code in the HTML file.



The codes below in the PHP file is to check whether the verification code inputted by the user is correct.



The resulted verification code is as shown.

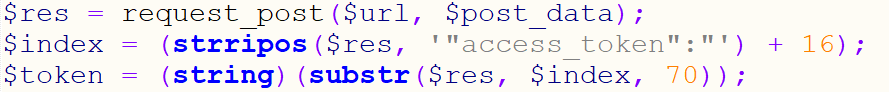


**Part 2: Functions**

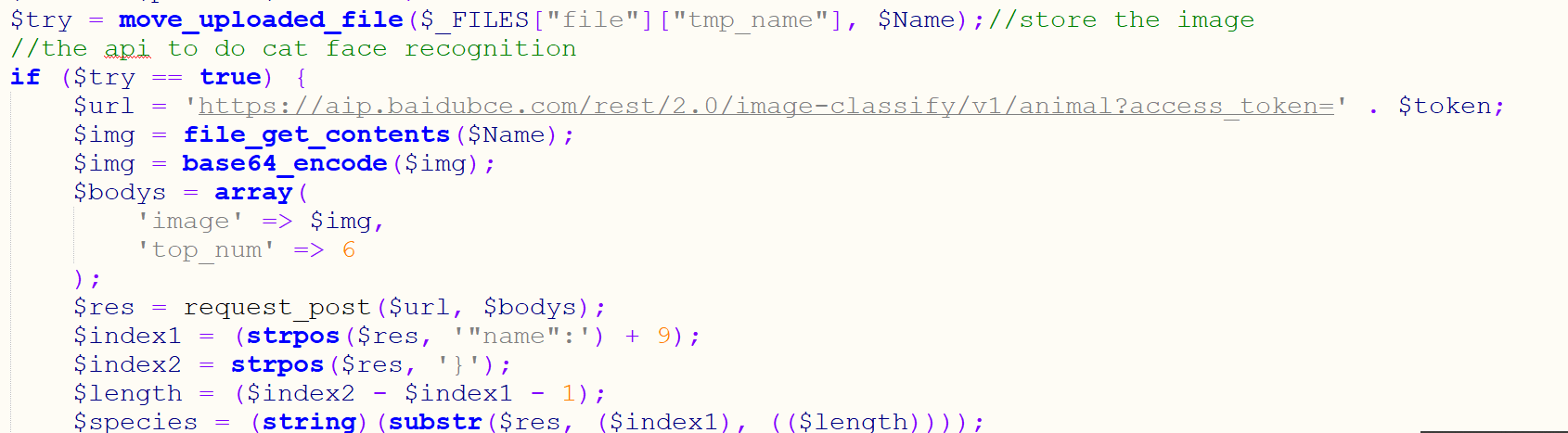
1. **API for cat face recognition and translation**

As it is hard to train a model to tell the breed of the cat in the picture, I utilized the API from Baidu to reduce the work of training by myself. As a result, I could just send the picture as a request to it and gain the response back, which is data in json type. I processed the data and select the partition I want.

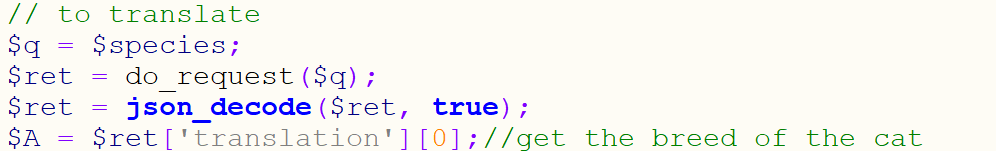
This is the process of automatically generating the token. As the token will be expired by 30 days, I made it automatically update by coding.



This the process of sending and retrieving data.

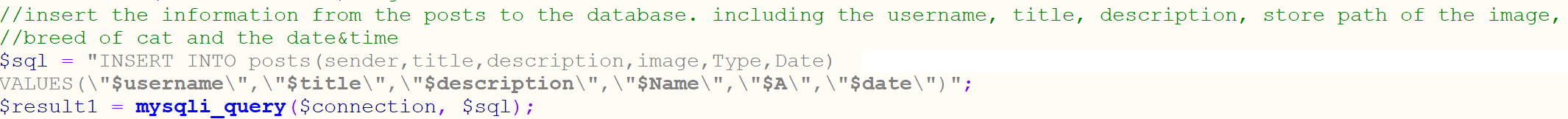


However, as the breed retrieved is in Chinese, but my website is in English, I used another API from Youdao to translate it into English.



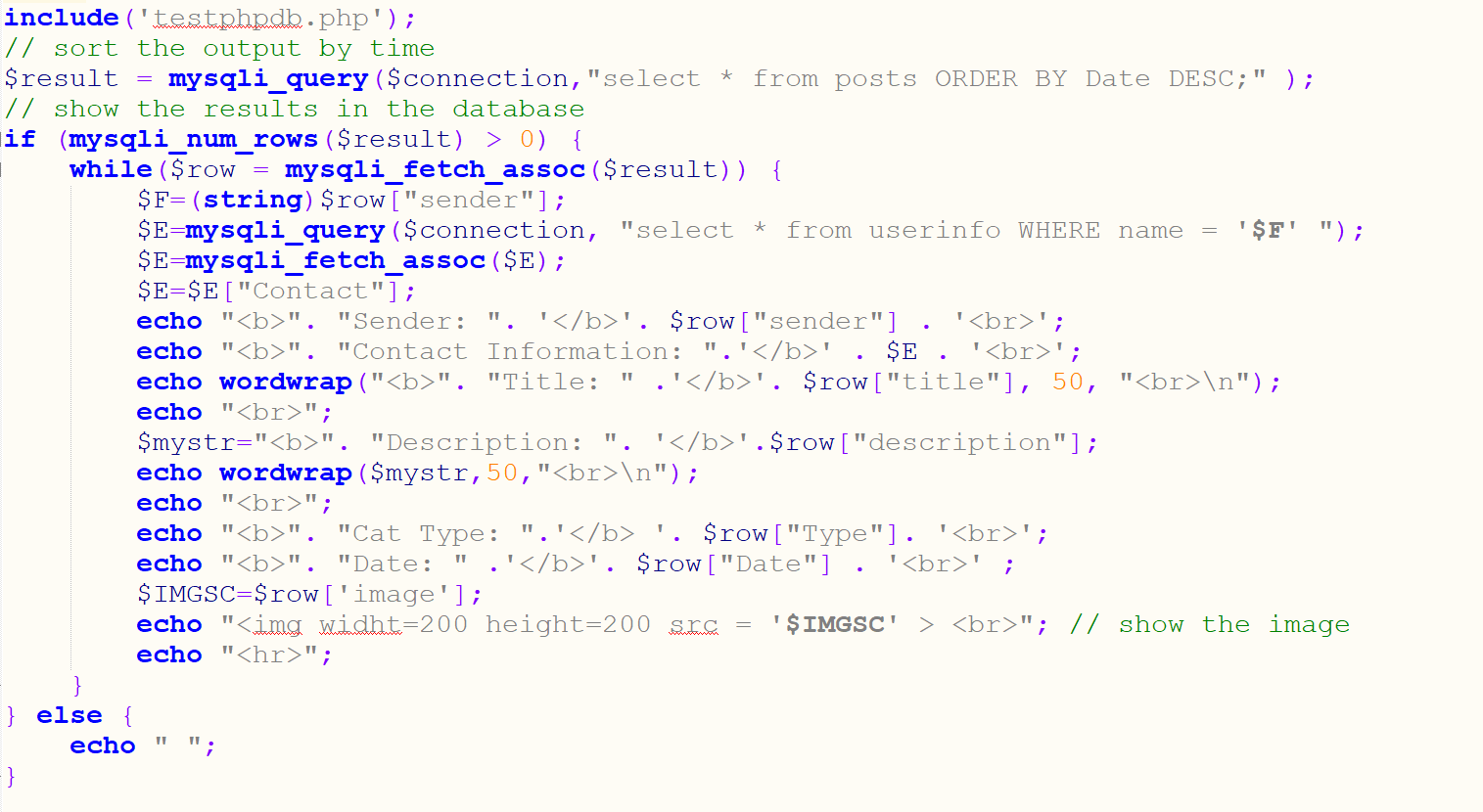
1. **Insert posts information into the database**

As the users inputted the posts information using the HTML page, it is needed to insert them into the database for further usage using the command INSERT.



1. **Retrieve from Database and show in the main viewing page**

As the information for the posts is stored in the database, it is required to retrieve from the database to show on the page for users to view. I used a while loop to automatically show all posts on the page according to the time they are inserted. As the image is stored in another folder and in the database, it is only stored as a path, I use HTML sentence to retrieve the image from its path. As I added echo ‘<hr>’ at the end of loop, there is a line to divide each post.



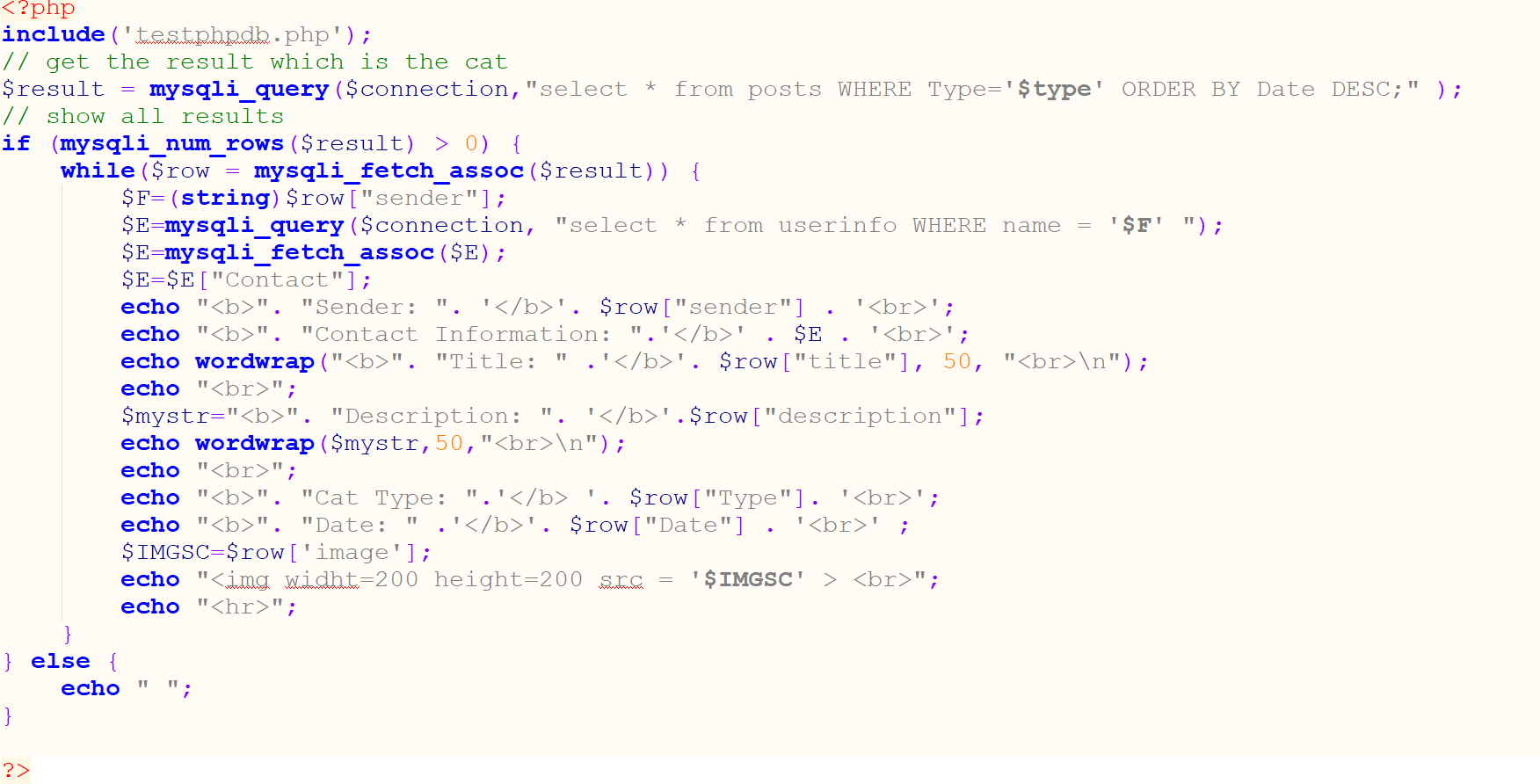
1. **Searching and sorting in the database and show the results**

As it is needed to show the result only containing certain type of breed, it is needed to have the function of searching.

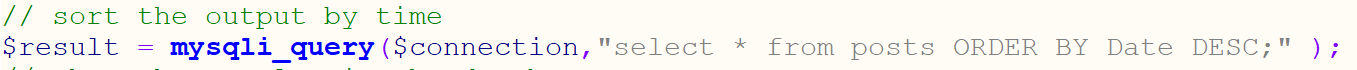
As the user inputted the type of cat that they want using the following code.



The website will automatically jump to the search result page, which only shows the type of cat the user wants.



Besides, the posts are sorted by recent. Therefore, it requires the use of “ORDER”.

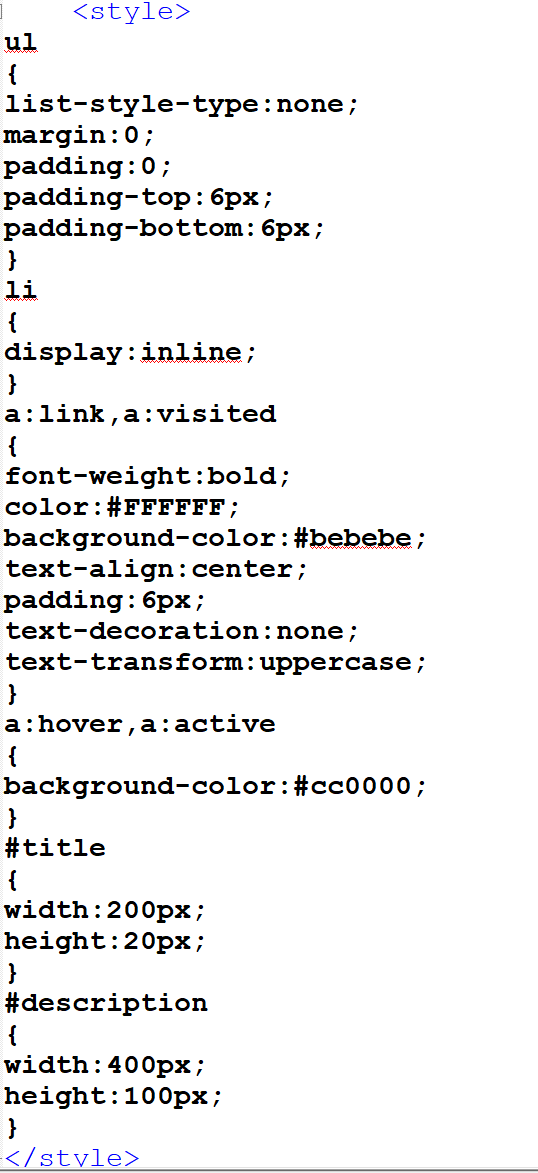


**Part 3:Interfaces**

1. **CSS and HTML arranging**

The arrange of the elements in the webpage and the design of them required the use of CSS, like the design of the menu bar and the design of the table showing all posts of a user.

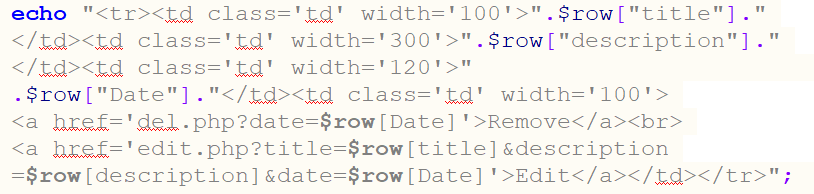
Here is the use of CSS in my design.



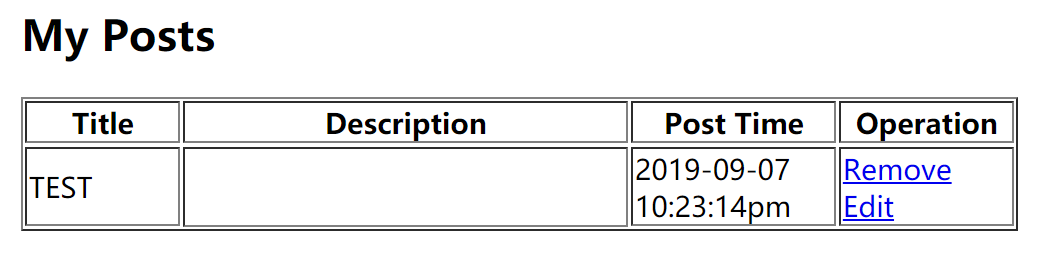
This is the result yielded. When having a mouse hovering on one of the bars, it will change its color to red.



Here is another method of writing that is specific to the object rather than controlling the whole page which I used and will produce the same menu bar.



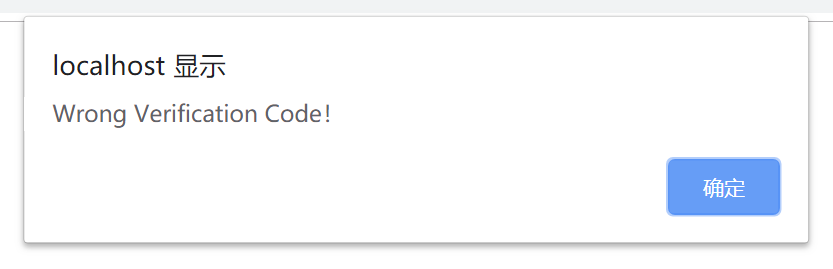
This is the code of creating the table shown below. In this I mainly used coding specific to design a table.

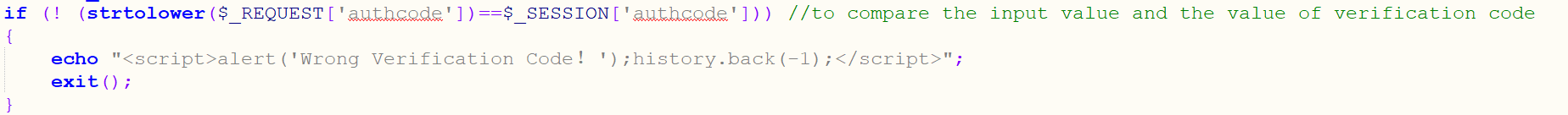


1. **Use of JavaScript for pop-up windows**

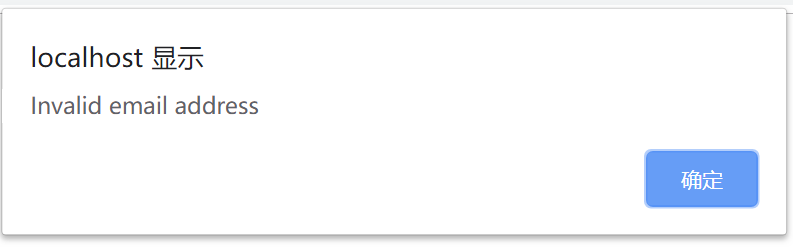
Sometimes it is required to notify the user the current state, like whether they successfully post or where is wrong in their inputting.

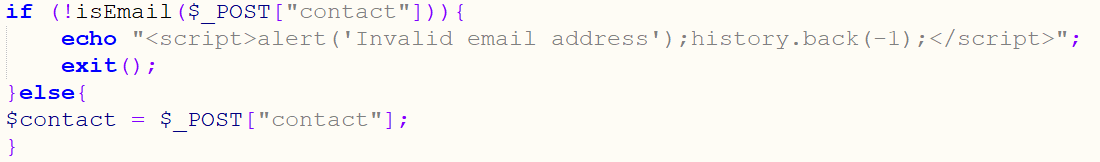
For example, when users in registering, they may make a range of mistakes, the re-typed password is different from the original one, the form of email address is invalid, the username is already been taken, the verification code is wrong and so on. As a result, a pop-up window is required. The following pictures shows all kinds of pop-up windows used in the registration page and the corresponding code.



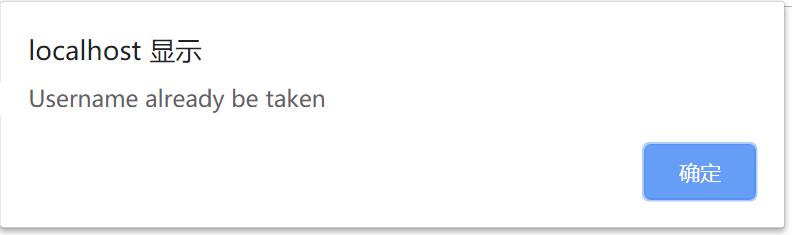


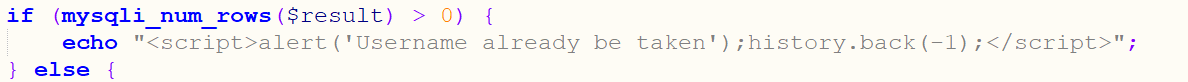
(Wrong verification code)



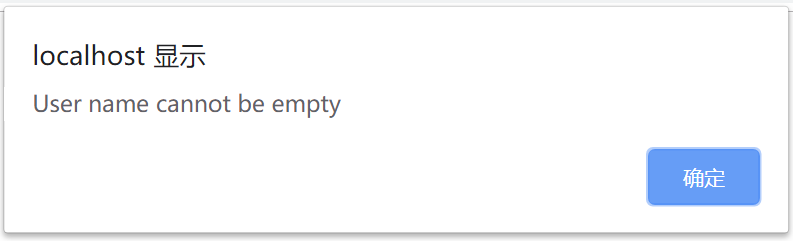


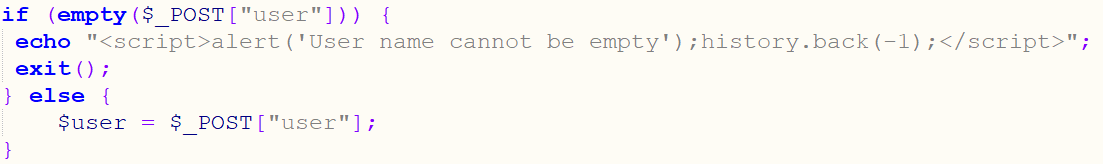
(Invalid email address)



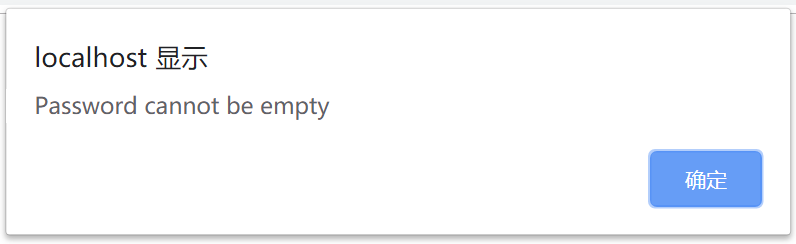


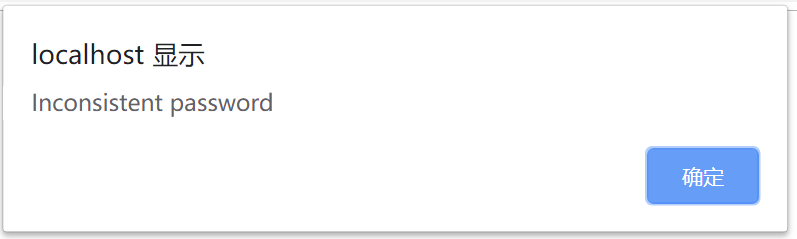
(Username already be taken)

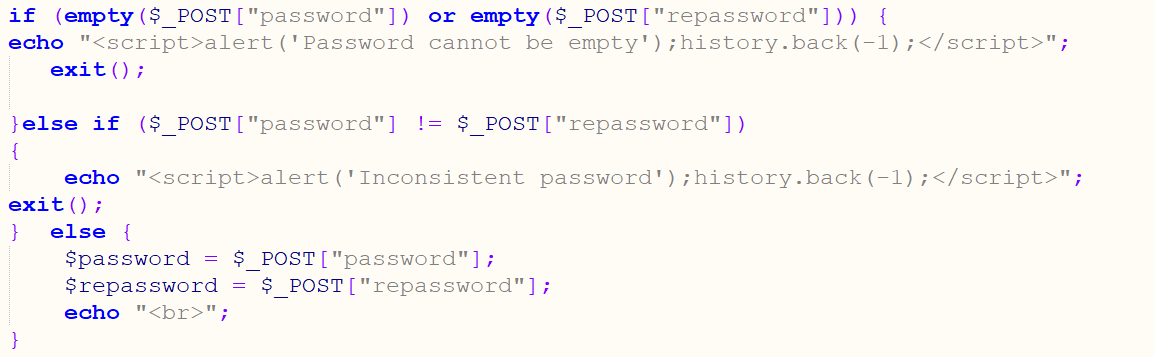




(Empty username)







(Empty or inconsistent password)

Word Count: 999

**Reference**

“php实现简单的验证码功能”. 03 09 2017.*bubuko*. Source Code. 06 2019. < <http://www.bubuko.com/infodetail-2290288.html>>.