

INFO1111: Computing 1A Professionalism

2023 Semester 1

Self-Learning Report

Submission number: 1

Github link: <https://github.com/Iambehindyou2/Info1111-Self-Learning>

Student name	Kailin Zhang
Student ID	5305576006
Topic	PHP
Levels already achieved	A
Levels in this report	B

1. Level A: Initial Understanding

1.1. Level A Demonstration

- The Installation of PHP and the IDE/Code editor in order to write in PHP. This includes getting familiar with PHP basic syntax and able to write simple PHP script
- Using PHP to connect to simple databases, this may include filtering, displaying and counting data etc
- Learning Object-oriented programming of PHP

1.2. Learning Approach

In the very beginning, before I selected my self-learning topic, I conducted some simple research on all the options available. I discovered that I had an interest in cybersecurity and PHP was a popular language for web development and back-end, so I chose it as my topic. Although I had zero knowledge of PHP, I first watched some brief introduction videos on YouTube to gain some basic knowledge. After understanding what PHP is, I searched for a tutorial on installation. Personally, I believe that learning without practical application is worthless because people will soon forget what they've learned. I like to learn by practising on my own to gain a better understanding. After I installed PHP and the IDE and was ready to go, I found a website called "w3school" that provided detailed knowledge and lectures on PHP. I went through the lectures while referring to a five-hour video on YouTube that talked about PHP. I found that incorporating videos and lectures helped me learn better.

1.3. Challenges and Difficulties

I think in general all computer languages, no matter which ones, are a challenge for me because I don't have experience with using them. After selecting the PHP, I found out that the PHP coding language is like a big ZIP file, it also contains HTML, MYSQL and other languages or forms of coding. For instance, using the level A self-learning as an example, in order for me to create an actual web I have to use HTML. However, at the same time I have to embed the PHP into the file in order for users to input value and also for "me" to receive the value. Connecting the database is like on a different level to be honest, personally speaking I think that is for level B and above. Because it requires knowledge toward using MYSQL and that is like another different set of coding. I already downloaded "DataGrip" and am exploring how to use DataGrip to manipulate SQL files and connect through with PHP.

1.4. Learning Sources

- https://www.youtube.com/watch?v=OK_JCtrrv-c&t=2640sThis is a 5 hour video explaining and tutoring on PHP, it helped me a lot with all the syntax and functions of PHP[1]
- https://www.youtube.com/watch?v=a7_WFUF94This is a short video briefly introducing PHP overall[2]
- https://www.bilibili.com/video/BV18x411H7qD/?spm_id_from=333.337.search-card.all.clickThis is also a very detailed tutoring video on PHP, it helped me a lot on making the local host as the server[3]
- https://www.w3schools.com/php/php_math.aspThis is a lecture in text, going through all types of math, functions, data type and all other syntax[4]

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Login Page</title>
5 </head> <!-- I used a HTML skeleton for startup-->
6 <body>
7 <h2>Login Form</h2>
8 <form action="New.php" method="POST">
9
10 <!-- Using method "post" so that it can hide the url parameter from users, making
11 it more secure especially for login page like this-->
12 </action>
13 <label for="username">Username:</label>
14 <input type="text" id="username" name="username"><br><br>
15 <label for="password">Password:</label>
16 <input type="password" id="password" name="password"><br><br> <!-- Using id Password can hide that actual password from
17 typing-->
18 <input type="submit" value="Login"><br><br><br><br><br><br> <!-- By giving the type "submit" a value, the submit button
19 then can be changed into "Login"-->
20 First Number: <input type="number" name="Num1"> <br><br><br>
21 Operation: <input type="text" name="Op"> <br><br><br>
22 Second Number: <input type="number" name="Num2"> <br><br><br>
23 <input type="submit" value="Calculate!"> <br><br><br>
24
25 </form>
26 </body>
27 <?php
28 #Check if the form has been submitted
29 if($_SERVER['REQUEST_METHOD'] == 'POST'){
30     $username = $_POST['username'];
31     $password = $_POST['password'];
32
33     #Check if the username and password are valid
34     if($username == 'Kailin' && $password == 'Zhang'){

```

```

32
33     #Check if the username and password are valid
34     if($username == 'Kailin' && $password == 'Zhang'){
35         //using if function can mimic the login page using correct username and password
36
37         exit(); //I don't have a another page or file to redirect so I left blank here
38     } else {
39         // Display an error message
40         echo "Invalid username or password";
41     }
42 }
43
44 # Check if the form for calculation has been submitted
45 if(isset($_POST["Num1"], $_POST["Op"], $_POST["Num2"])){
46     $Num1= $_POST["Num1"];
47     $Op= $_POST["Op"];
48     $Num2= $_POST["Num2"];
49     // Basically giving the Op 4 different values to check if the operations is one of "+","-","*","/"
50     //On the top I set the variables $Num1 and $Num2 into number, which the input can only be numbers
51     if($Op == "+") {
52         echo $Num1 + $Num2 ;
53     } elseif($Op == "-"){
54         echo $Num1 - $Num2;
55     } elseif($Op == "/"){
56         echo $Num1 / $Num2;
57     } elseif($Op == "*"){
58         echo $Num1 * $Num2;
59     } else{
60         echo "Input Invalid";
61     }
62 }
63 // There is a typo in the coding that the final web cannot operate single section, so for example if I use the
64 //calculator, the login output will also occur, Currently I don't know how to fix it.
65 <?>

```

2. Level B: Basic Application

2.1. Level B Demonstration

So my topic that I chose was PHP, so for level b I developed some what knowledge in XAMPP which is a web sever that supports PHP language and also MySQL database. I used XAMPP in level b to generate a local sever for me to use php html css with the MySQL database to create a sign up web page that can store the users' input (usernames and password) Meanwhile, I have also used Phpadmin to store the login data using MySQL.

2.2. Application artifacts

I have created a simple website with a sign-up page that can store user input in a database. The process I followed is as follows:

1. Downloaded XAMPP, a cross-platform server, to create a local host on my laptop.
2. Launched Apache and MySQL inside XAMPP to create a local server that supports PHP and MySQL databases.
3. Downloaded phpMyAdmin and inserted it into the XAMPP "htdocs" folder to help manage the MySQL database.
4. Created a MySQL database and user with all privileges using phpMyAdmin.
5. Created a table under the database structure to store users' input usernames and passwords.
6. Used Visual Studio Code to create PHP files for connecting to the database and processing the sign-up form.
7. Incorporated an HTML sign-up form template from the getbootstrap website, and used the "post" method for secure data transmission.
8. Utilized prepared statements with `mysqli_prepare()` to prevent SQL injection attacks.
9. Implemented error handling for cases when user information is already stored in the database.

The resulting website has a simple structure with a sign-up page that stores user input in the database. If the user's information is already stored in the database, it will detect and output an error; otherwise, it will store the information successfully.

Here is a code snippet from the PHP file for connecting to the MySQL database:

```
<?php
// Connection details
$hostname = "SQL";
$username = "k";
$password = "123";
$database = "registration";

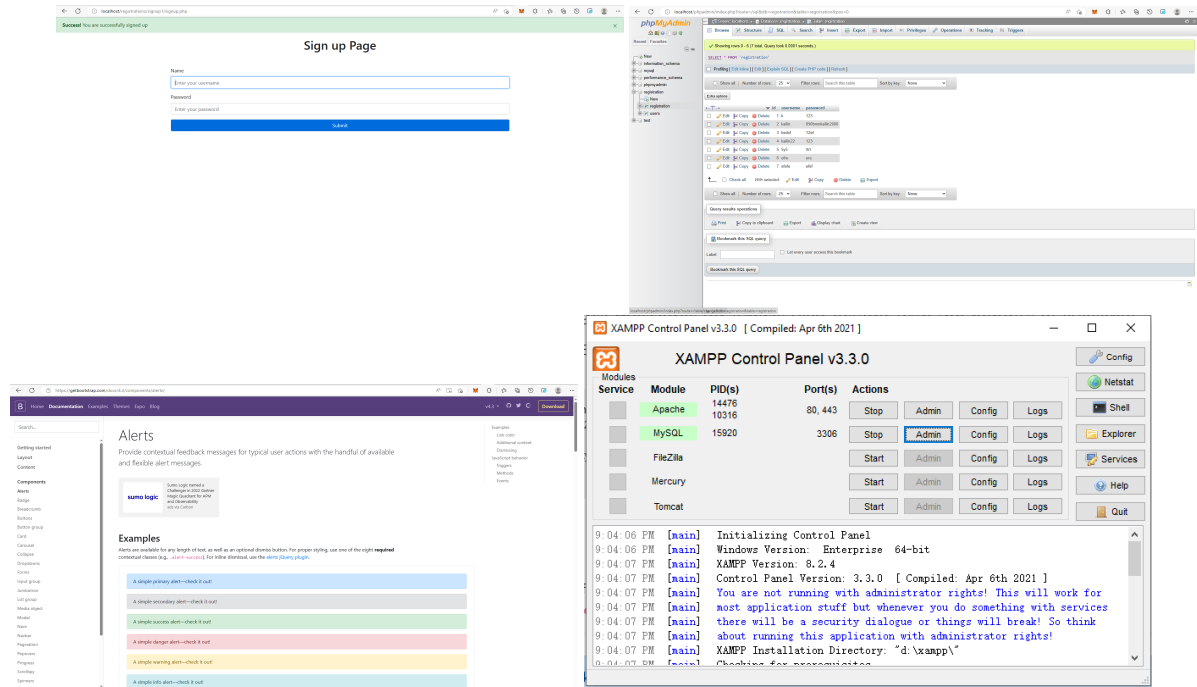
// Create connection
$conn = mysqli_connect($hostname, $username, $password, $database);

// Check connection
```

```

if (!$connection) {
    die("Connection failed: " . mysqli_connect_error());
}
?>

```



Bibliography

- [1] freeCodeCamp.org, “Php programming language tutorial - full course,” 2018. [Online]. Available: https://www.youtube.com/watch?v=OK_JCtrrv-c&t=2640s
- [2] Fireship, “Php in 100 seconds,” 2021. [Online]. Available: https://www.youtube.com/watch?v=a7_WFUIFS94
- [3] B. H. Programmer, “Black horse programmer php zero to proficiency tutorial (p1 basic 6 days),” 2017. [Online]. Available: <https://www.bilibili.com/video/BV18x411H7qD>
- [4] W3School, “Php tutorial,” N/A. [Online]. Available: https://www.w3schools.com/php/php_math.asp