

INFO1111: Computing 1A Professionalism

2023 Semester 1

Self-Learning Report

Submission number: 1

Github link: <https://github.com/Iambehindyou2/info1111>

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

Instructions

Important: This section should be removed prior to submission.

You should use this L^AT_EX template to generate your self-learning report. Keep in mind the following key points:

- **Submissions:** There will be three opportunities during the semester to submit this report. For each submission you can attempt 1 or 2 levels. Each submission should use the same report, but amended to include new information.
- **Assessment:** In order to achieve level B, you must first have achieved level A, and so on for each level up to level D. This means that we will not assess a higher level until a lower level has been achieved (though we will review one level higher and give you feedback to help you in refining your work).
- **Minimum requirement:** Remember that in order to pass the unit, you must achieve at least level A in the self-learning (unless you achieve level B in both the skills and knowledge categories).
- **Using this template:** When completing each section you should remove the explanation text and replace it with your material.
- **Referencing:** You should also ensure that any resources you use are suitably referenced, and references are included into the reference list at the end of this document. You should use the IEEE reference style [?] (the reference included here shows you how this can be easily achieved).

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

- This is a 5 hour video explaining and tutoring on PHP, it helped me a lot with all the syntax and functions of PHP
- This is a short video briefly introducing PHP overall
- This is also a very detailed tutoring video on PHP, it helped me a lot on making the local host as the server
- This is a lecture in text, going through all types of math, functions, data type and all other syntax

1.5. Application artifacts

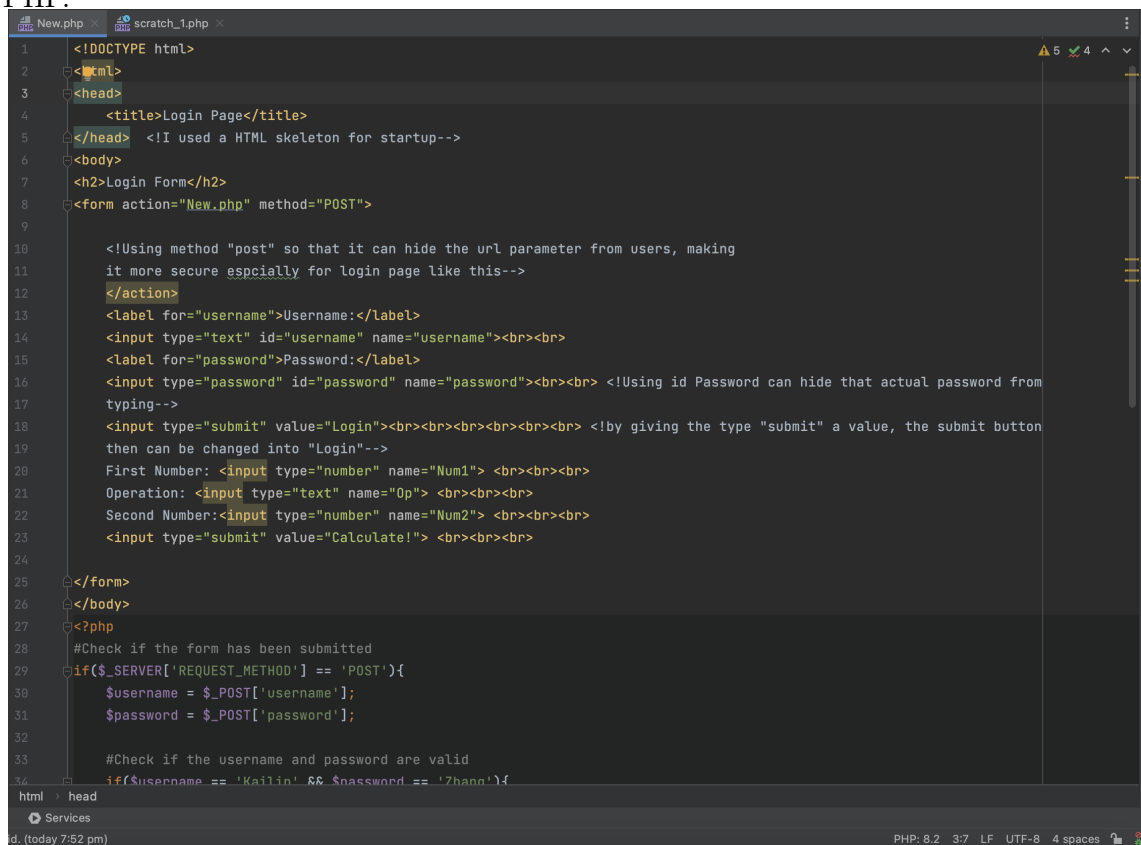
Installation of PHP and IDE: I first installed Homebrew on my MAC OS by running the command “/bin/bash -c and then installed PHP by running the command “brew install php”. After adding the Homebrew PHP path to my path, I was able to use PHP.

Creating a local server: To create a local server, I navigated to the directory where I had saved my PHP file and ran the command “Php -S localhost:4000”. This created a local server that I could use to make a simple webpage by copying the URL to my Chrome.

Creating a login page and calculator: After successfully creating a local server and linking to the PHP file, I created a simple login page that allowed users to input their username and password. I used signing values to variables, if function and method post to hide the URL parameter. If the password and username were not valid, it would display invalid information. Additionally, I also coded a simple calculator that could perform operations such as addition, subtraction, multiplication, and division. The user could input different numbers and then click on the operation they wanted to perform.

Commenting in the coding document: To further explain my work, I used // for PHP and <!-> for HTML to add comments to the coding document.

Challenges faced: One of the problems I faced was that the calculator and login page could not operate separately. This meant that both would output at the same time, and I currently do not know how to fix it. However, I am still learning and making progress on PHP.



```
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> <!-- 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 == '7hann'){
```

```

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 }// There is a typo in the coding that the final web cannot operate single section, so for example if I use the
63 //calculator, the login output will also occur, Currently I don't know how to fix it.
64 ?>
65
html > head

```