

Patricia Mae I. Vargas  
CMSC 198: AR11  
2020-01023

### **Weekly Journal: July 8-12, 2024**

In the third week of my internship at RGRR Webmaker, our OJT supervisor consistently assigned us tasks and deliverables to complete within the week. We continued working on the Water District system project. Based on prior client interviews, we needed to implement several changes in the code. Specifically, we adjusted the combined approach for computing bills under a single OR, modified and removed redundant form inputs such as the type of connection when adding a client, and managed these details behind the scenes.

We were also tasked with creating a JavaScript popup box to display the water consumption of clients over the past six months. Additionally, we received MySQL code that required modification to automatically show the count of ORs in an HTML form field. The function we created to find the latest OR number initially searched the entire database, which was inefficient for a system with potentially thousands of clients over several years. We optimized the function to search for ORs within the current month. However, this approach presented challenges, such as handling scenarios where no ORs were generated within the month or at the start of a new month.

In addition to these tasks, we were instructed to improve the overall code quality for the project. We also began exploring GCash integration. While I initially studied Adyen for this purpose, I found its configuration complex and switched to the PayMongo API. I am still in the process of familiarizing myself with and studying this API for payment integration.

Concurrently, we continued working on two store projects, although these were given lower priority. We were also tasked with experimenting with OpenCart. I personally tried using OpenCart to build an eCommerce website. The installation process was straightforward, thanks to the clear documentation provided. However, setting up a local development environment with XAMPP was initially unfamiliar and confusing. Understanding OpenCart's structure, navigating the admin panel, and familiarizing myself with the file hierarchy took some time, but once I acclimated, I appreciated the organized and modular system. The distinction between the admin and frontend sections helped compartmentalize tasks effectively.

Customizing themes was both challenging and rewarding. Editing '*twig*' files and CSS required a solid understanding of OpenCart's templating system. I experimented with various themes from the OpenCart marketplace, learning a great deal about responsive design and user interface adjustments to match the clothing store's branding. Adding products, categories, and attributes was relatively straightforward but required meticulous attention to detail, especially for clothing items with various sizes and colors. I developed an efficient workflow for batch uploading products, which saved time. However, since some details about the clothing and seashells store of our clients weren't discussed to us yet, I have worked with dummy data. I am

still working on it and have not yet made the final output, though the experience was invaluable. It highlighted that building such sites from scratch was unnecessary, given the tools available.

As mentioned in last week's journal, our OJT supervisor has consistently emphasized the importance of maintaining privacy and confidentiality in our work. Consequently, I am unable to share specific examples or attach our work products. Despite this restriction, our supervisor has been diligent in monitoring our progress and providing guidance.

He regularly checks in with us through our dedicated Messenger group chat and conducts Zoom meetings to discuss our tasks and address any challenges we encounter. This approach ensures that we remain on track and receive timely feedback and support.

Our work schedule continues to follow the structured hours of 8:00 AM to 5:00 PM, resulting in a total of 40 hours dedicated to our projects during the week of July 8-12. This time commitment has been officially documented and validated by our manager and OJT overseer.

Here is a single picture allowed by OJT supervisor to be shown:

```
1 public function getOr(){
2     //just select what you need and don't bring out everything for security
3     $sql = "SELECT MAX(or_no) AS max_or_number FROM (
4     SELECT or_no FROM payments WHERE MONTH(or_date) IN (MONTH(CURRENT_DATE), MONTH(DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH))) AND YEAR(or_date)
5     UNION ALL SELECT or_no FROM otherpayments WHERE MONTH(or_date) IN (MONTH(CURRENT_DATE), MONTH(DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH))) AND
6     ) AS combined_data";
7     $data = array();
8     $statement = $this->executeSQL($sql, $data);
9     $result = $statement->fetch(PDO::FETCH_ASSOC);
10    $max_or_number = $result["max_or_number"];
11    $max_or_number = $max_or_number + 1;
12
13    return $max_or_number;
14
15 }
16 }
```