Final Report

for

ThinkFood

By

ICS Group 06

Contents

Table of Figures

1. Project Overview

ThinkFood is a web application where customer can order foods online. Instead of googling each restaurant to order foods, with ThinkFood, it lets you easily find and order foods in one website without much hustle. The feature for the website is to have a menu for each category by foods and restaurants. Customers can make purchases, view order history and much more. For administrators, they can add or remove menu from the website, and as restaurant owners they can receive orders from the administrator.

The objective is to design a website where customers can add, remove, rate, comment, make purchases, and view history of their orders. As for Administrator, they can add and remove product, view the order of their customers, and deliver the food to the customer. For the Restaurant users they can view the orders that customers made. We will implement a database and have it all integrated into the website.

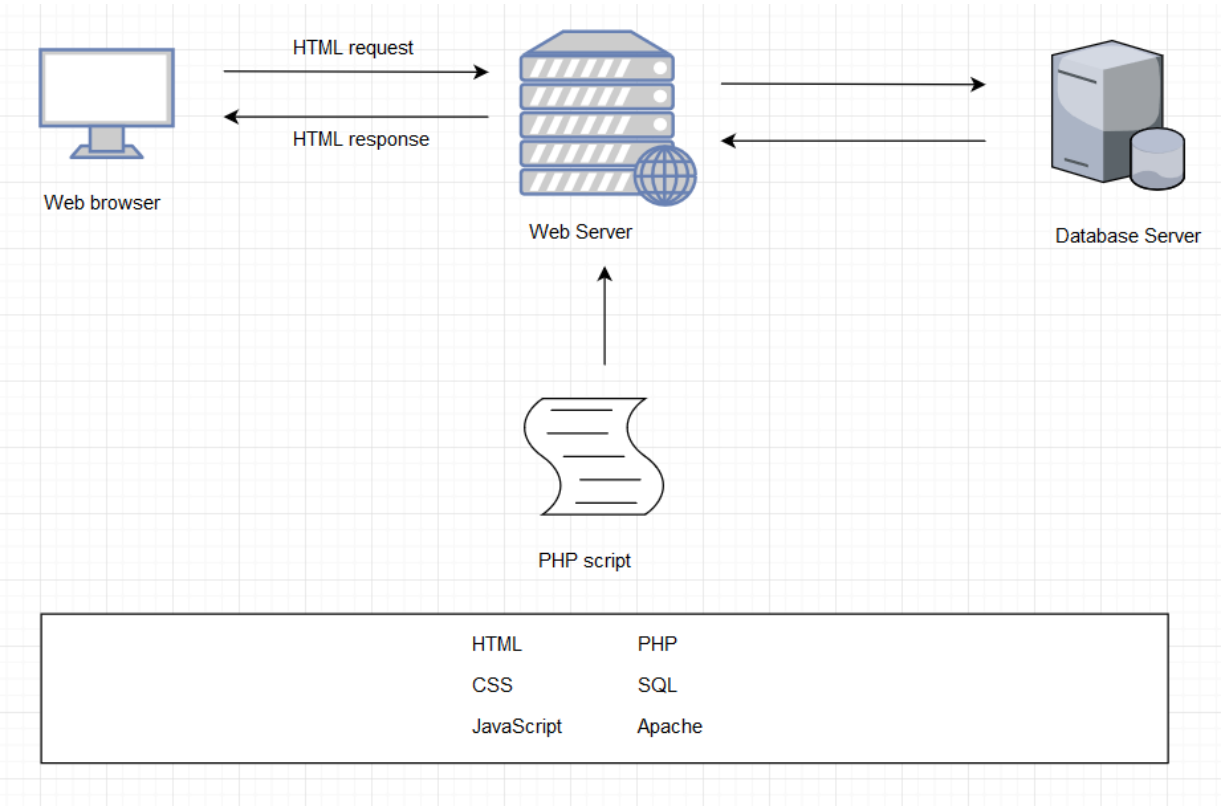

Figure 1. System Architecture

2. Project Specifications

2.1 Function Requirements

- R01 – The client must be able to log in to an admin account
- R02 – The client must be able to add products to the web site
- R03 – Must be able to display products by categories
- R04 - Customers must register to place orders
- Must provide a fully functioning shopping cart utility where a customer can:
    - R05 – display the current items in the cart
    - R06 - add selected products to the cart
    - R07 - delete products (individually or all) from their cart
    - R08 - adjust the quantity of a selected product currently in the cart
    - R09 - preserve the contents of the cart if the user's session is disconnected
    - R10 - have the cart accessible regardless from any machine the customer is logging in
    - R11 - Create a unique file for each order that contains the order confirmation information
    - R12 - include a secure, online payment process
- R13 - Allow customers to view their order history

Extra:

- R20 – Allow customers to provide reviews for any of the products they have purchased
- R21 – Must be able to display the evaluation reviews for any products
- R22 – Provide customers with appropriate product recommendations when they log in
- R23 – Provide customers with any product recommendations when they look at a particular product
- R24 – Ask customers to agree with the term condition on their first log in

2.2 Non-Function Requirements

- R12 - include a secure, online payment process
- R15 - Use free open source software
- R16 - Have the system fully tested and up and running in 8 weeks

2.3 Group Roles

Week	ChangSin Park	Qi Li	Rolando Pacho	Quang Pham	Jonghyun Choi
02	DBA/Back-End developer	Front-End developer	Middle-ware developer	Front-End developer	Team Lead Technical Writer
03	DBA/Back-End developer	Team Lead Front-End developer	Middle-ware developer	Front-End developer	Technical Writer DBA/Back-End developer
04	Front-End developer	DBA/Back-End developer	Team Lead Front-End developer	Middle-ware developer	Technical Writer DBA/Back-End developer
05	Team Lead Middle-ware developer	DBA/Back-End developer	Technical Writer Front-End developer	Front-End developer	Middle-ware developer
06	DBA/Back-End developer	DBA/Back-End developer Technical Writer	Front-End developer	Team Lead Front-End developer	Middle-ware developer
07	Front-End developer	Middle-ware developer	DBA/Back-End developer Technical Writer	DBA/Back-End developer	Team Lead Front-End developer
08	Middle-ware developer	Front-End developer	Front-End developer	Team Lead	Technical Writer

Table 1. Weekly Roles

3. Usability Guide

3.1 Customer

3.1.1 Register

Step 1. Go to register page and accept or reject the privacy policy agreement.

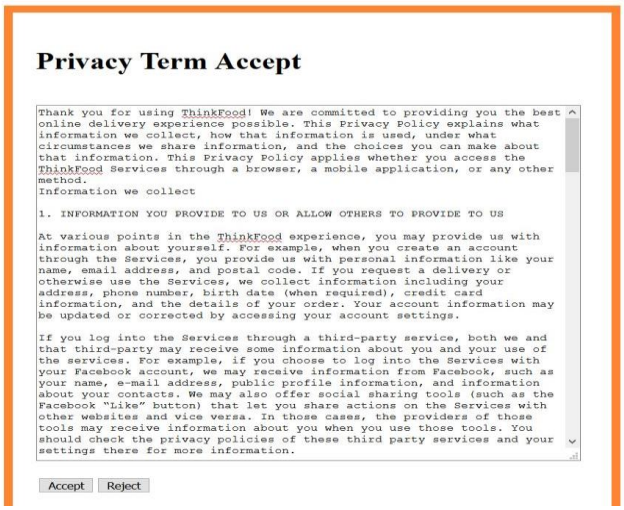

Figure 2. Privacy Term Accpet

Step 2. Fill the form and submit

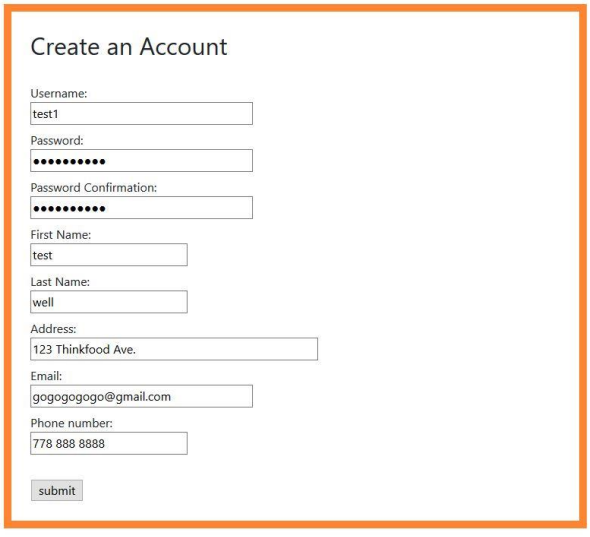

Figure 3. Login

3.1.2 Login

Step 1. Go to login page.

Step 2. Input username, password and then login.

Figure 4. Login

3.1.3 Product choose

Step 1. Go to product page

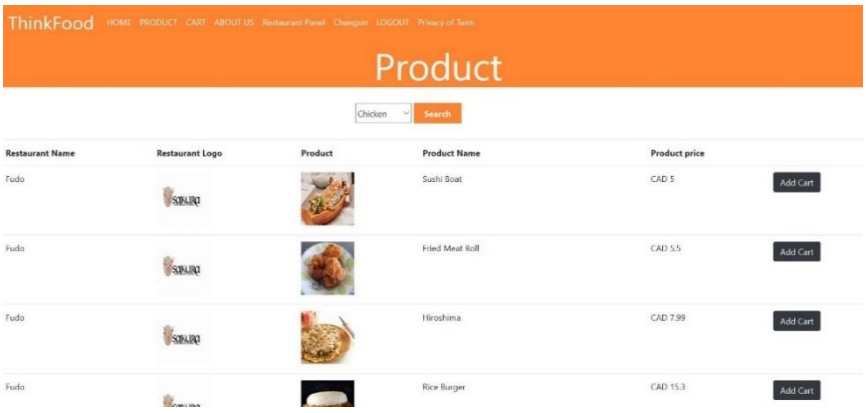

Figure 5. Product

Step 2. Add items to cart

Figure 6. Add to cart

3.1.4 Review

Step 1. Click product image to see that product's review and reputation.

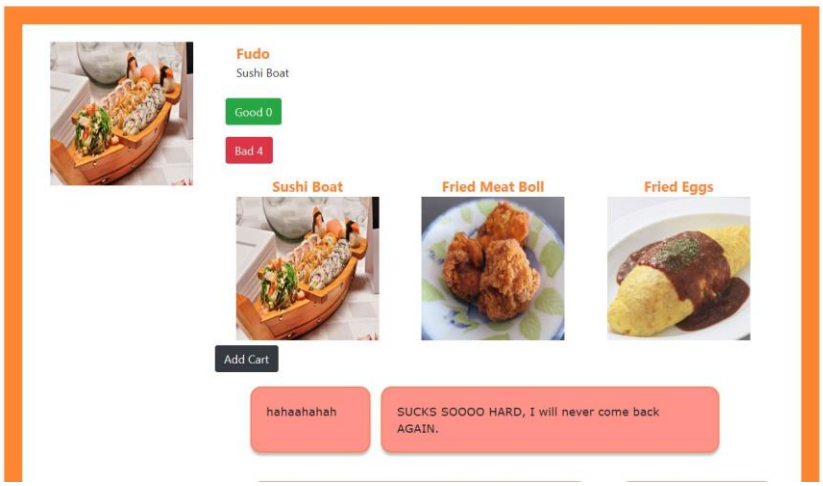

Figure 7. Product Review

Step 2. After checkout, user can to go to Order History to rate or review products they bought.

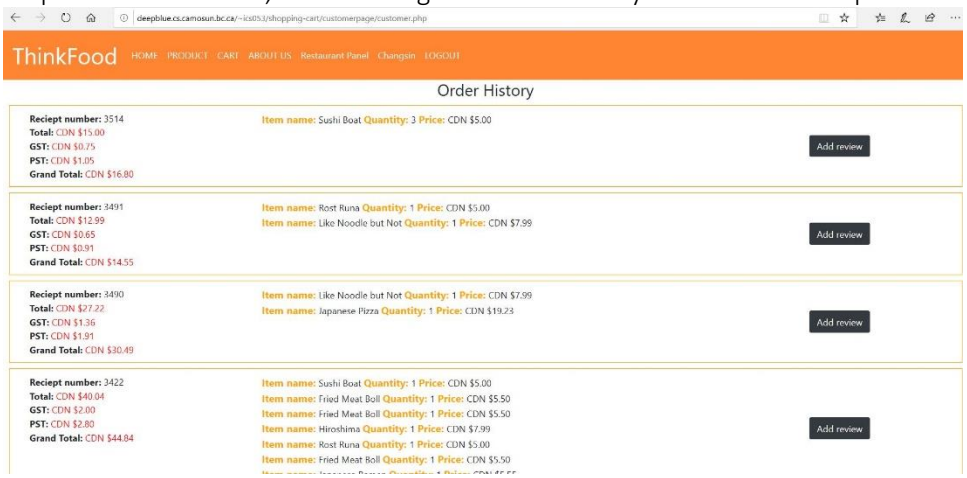

Figure 8. Order History

Figure 9. Review

3.1.5 Cart

Step 1. Go to cart to check the selected product

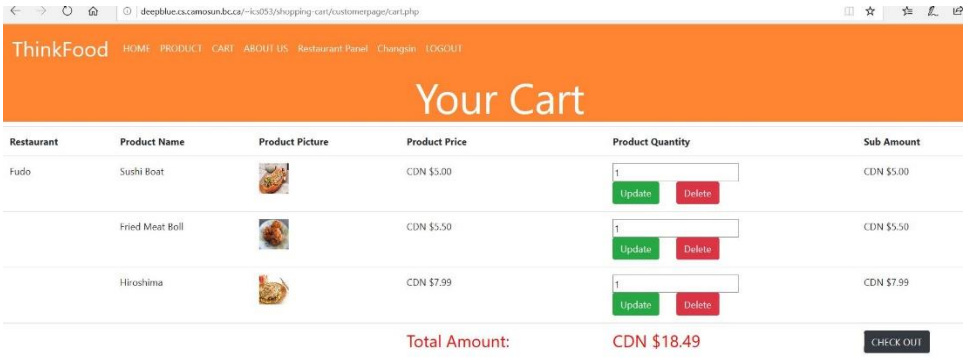

Figure 10. Cart

Step 2. Change each item's quantity

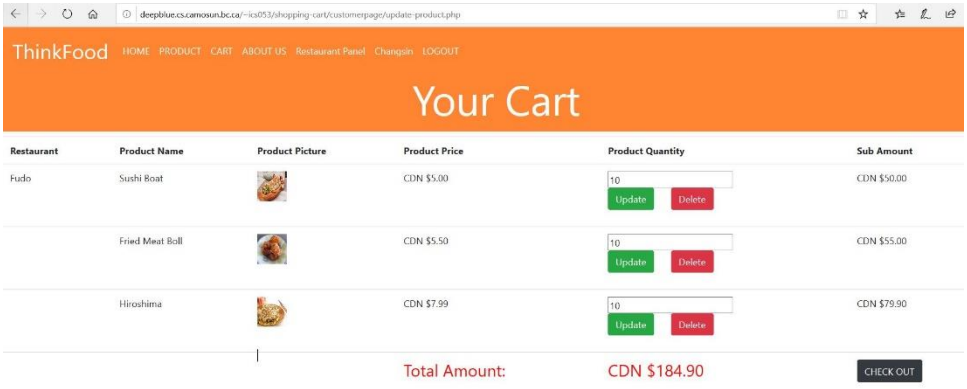

Figure 11. Cart

Step 3. Update quantity or delete a product

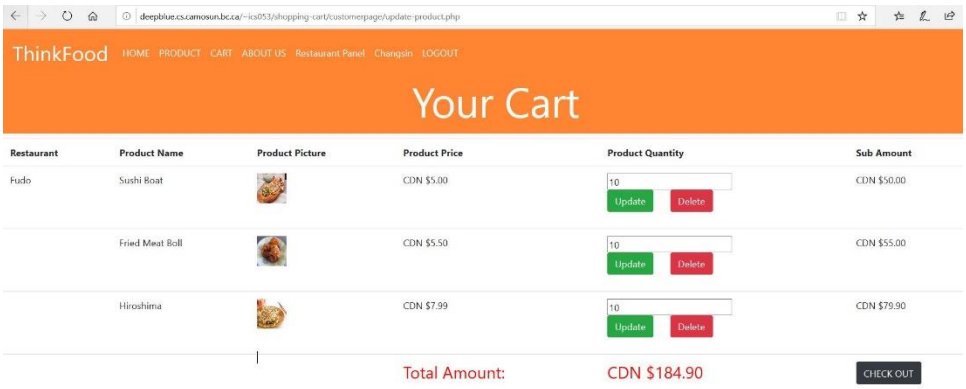

Figure 12. Update a cart

Step 4. Press check out button

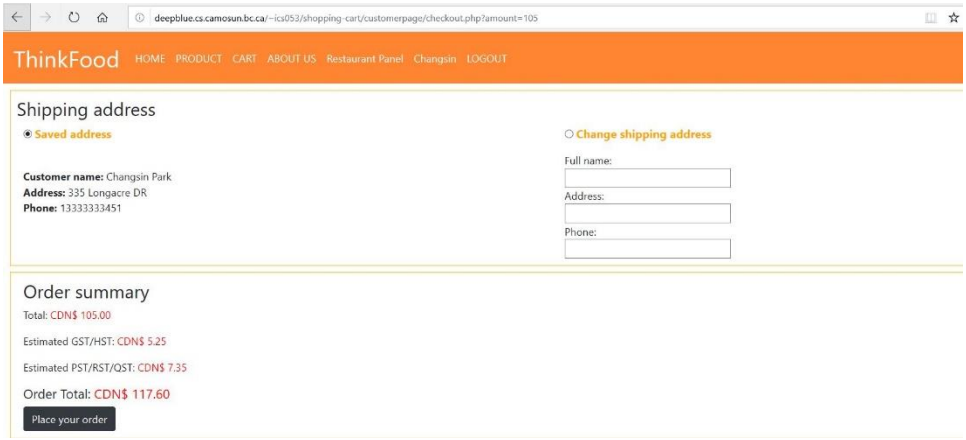

Figure 13. Check out

3.1.6 Check out

Step 1. In the checkout page, users should be able to modify their shipping address and check the info

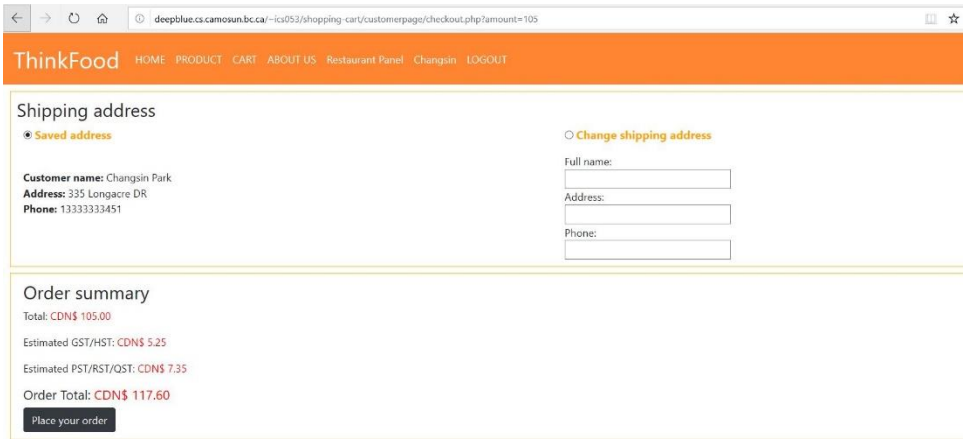

Figure 14. Order

Step 2. Click 'Place your order' button

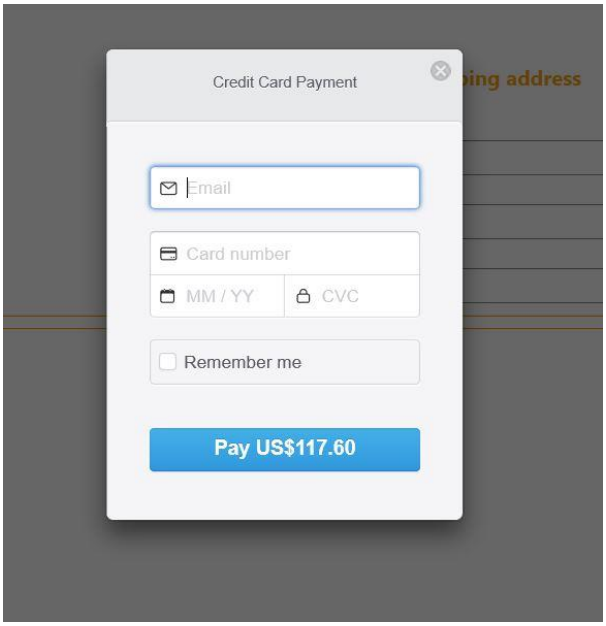

Figure 15. Payment

Step 3. After payment, users can go to their account view session, and see their order history

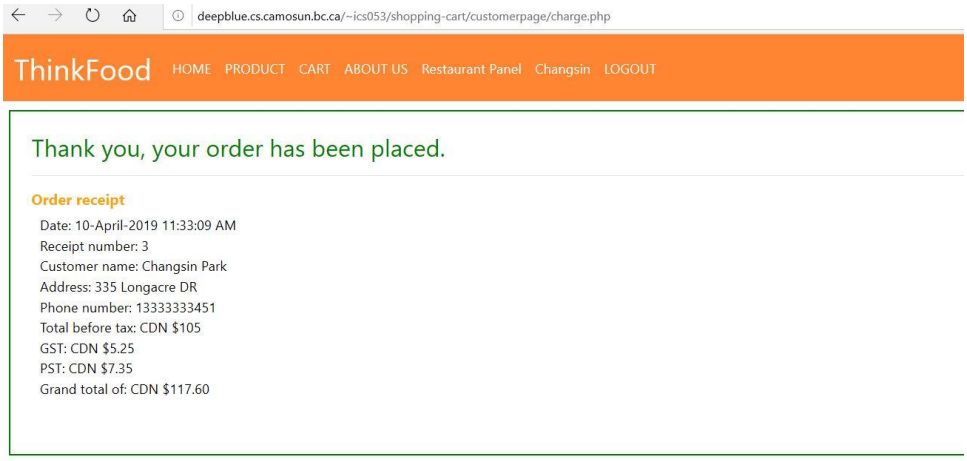

Figure 16. Order Detail

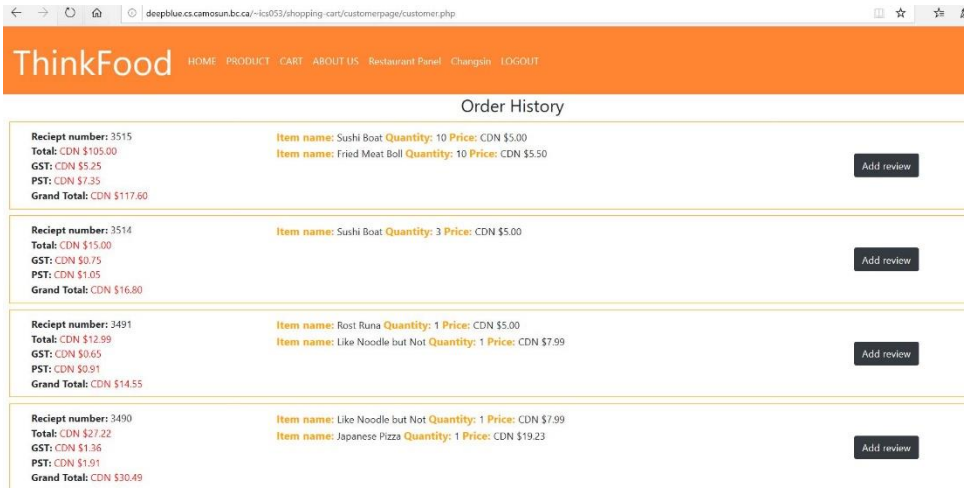

Figure 17. Order History

3.2 Restaurant

Step 1. Once a customer's orders confirmed, a client is be able to view all the orders

Step 2. By monitoring food status, when the restaurant is ready to deliver their order, it changes status from 'in progress' to 'on the go'

Step 3. After delivery is succeeded, the order status changes to 'completed'

Step 4. When the order delivery is completed, restaurant owner will be able to remove the order

Figure 18. Restaurant manage

3.3 Administrator

3.3.1 Administrator page

Step1. After login, you can go to administrator page automatically

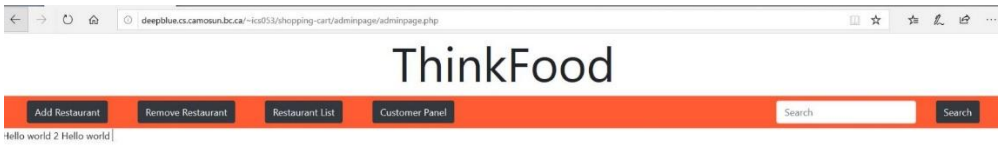

Figure 19. Administrator Page

3.3.2 Add a restaurant

Step1. Fill out mandatory items

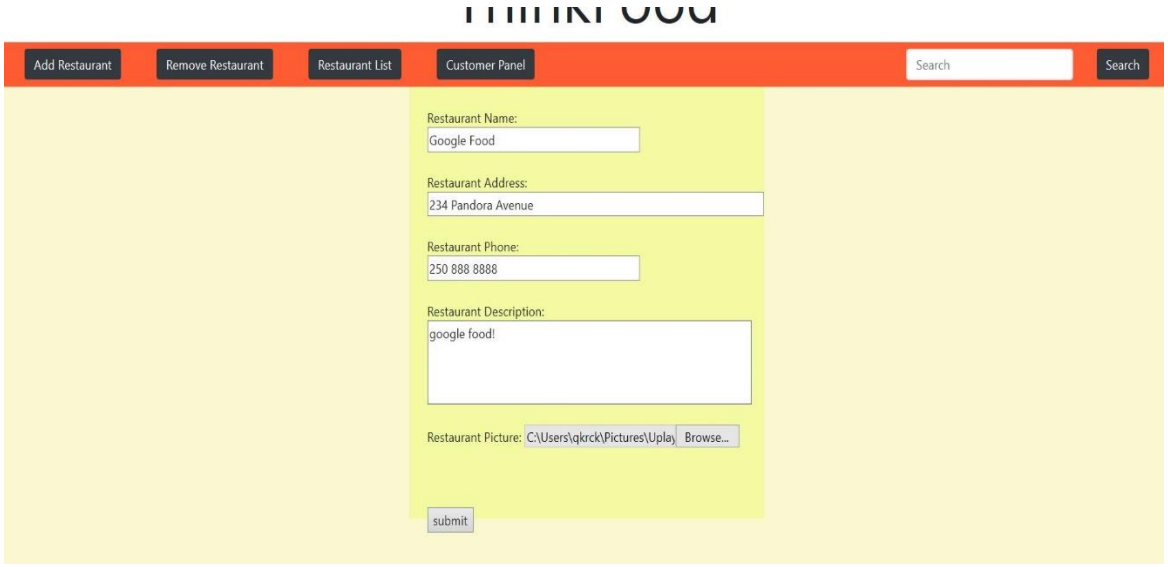

Figure 20. Enroll a restaurant

3.3.3 Add a product

Step1. Go to restaurant list page and add a product on the restaurant

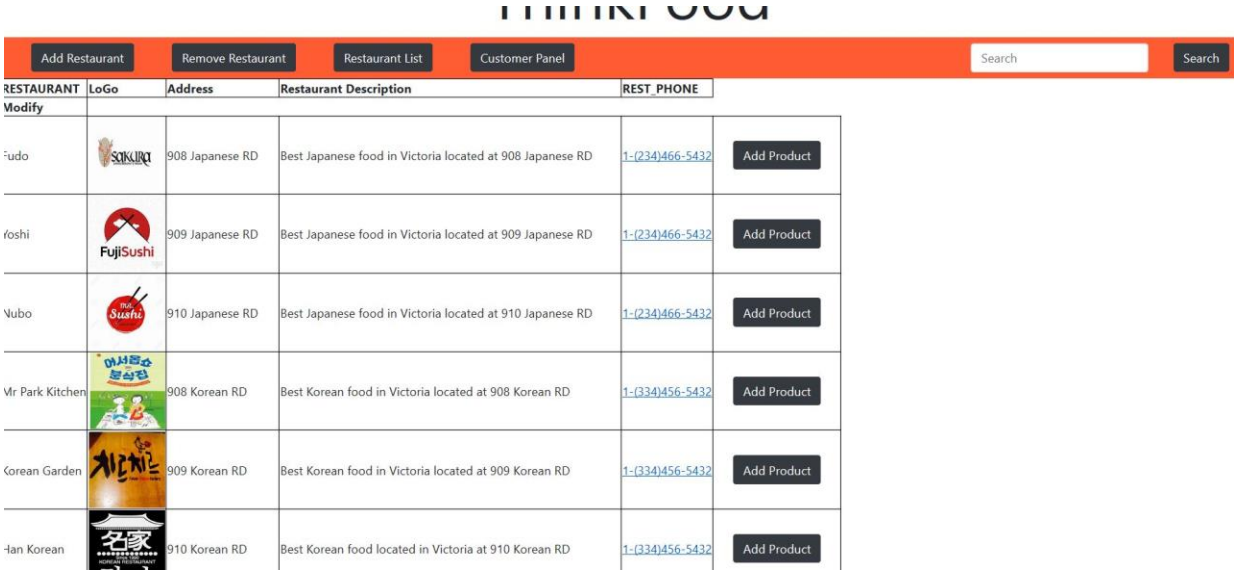

Figure 21. Restaurant Page

Step2. Fill out manadatory items

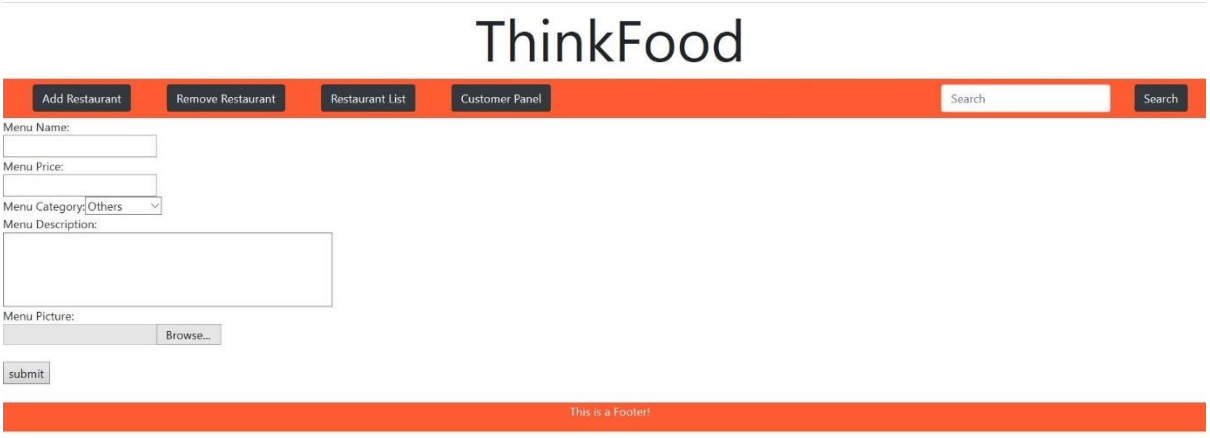

Figure 22. Enroll a product

4. Problems and Challenges

4.1 Managing tasks on schedule

We faced some problems to manage the schedule for delayed tasks, missing tasks, new tasks by changement
- We handled it by modified the priorities of some tasks and changed the schedule.

4.2 Standards

We didn't define naming rules on filename, variable, table & column, so it was hard to understand the other tasks. It was time consuming to us to figure out the sources and documents for deliverables.
- We defined the base files for standard and tried to follow the names in those documents.

4.3 GitLab

It was hard to use GitLab for managing resources, we lost some code files and faced conflict files between GitLab master and deepblue servers.
- We studied a GitLab and practiced some commands like pull, fetch and push command after making guideline.

4.4 Analysing requirements

We implemented the sites based on the written requirements but didn't figure out all of them in detail, so we got some unexpected checklist on checking by Doug and John.
- We got the check at the beginning of the class to prepare a missing requirement and immediate changement, so we could make a proper action on time.

5. Skill Learnt

5.1 Cooperation

We had regular meetings to check the tasks' progress and cooperate. We used Slack for quick communication and set the alarm using it for new changement on our GitLab. We could practice how to work together in efficient ways.

5.2 GitLab

Using GitLab encouraged us in manage our sources consistent. We used only one way from our GitLab master to deepblue server whenever it comes to deploy new versions of sources for version management.

5.3 MySQL and Workbench

We learned Oracle database and SQL Developer during past database classes. One of the non-functional requirement for this project is using an open source software, so we used MySQL for our data. The syntax is very similar with Oracle, We also used MySQL Workbench for modeling and reverse , forward engineer function for extracting metadata and creating objects each.

5.4 PHP, HTML, CSS

We could review web technologies such HTML, CSS and PHP through this project. Bootstrap is used for predefined styles for UI. PHP is used for server side implementation between user interface and database. It was a meaningful experience to us to practice all knowledge we learned at 1st year.

6. Proposed Changes

6.1 Time management

We will focus on assigning resources and tasks based on the priority with enough time in case some tasks are delayed by technical issues and requirement changement. This will be helpful to finish the project on time.

6.2 Dedicated role assign

We had some difficulties when we switch to the other roles every weekend. We could learn from the others' tasks and practice through roles. But it was time consuming to explain one's tasks to other team members. It will be more efficient to assign one team member on role.

7. Code Snippets

7.1 Batch for recommend products

```
[ics19906@deepblue ~]$ crontab -l
00 * * * * sh /home/student/ics19906/script/batch01.sh

CREATE DEFINER=`ICS199Group06_prod`@`%` PROCEDURE `SP_RECOMMEND_PRODUCT`()
BEGIN
 UPDATE PRODUCT SET RECOMMEND_YN = 'N';
 UPDATE PRODUCT SET RECOMMEND_YN = 'Y'
 WHERE PRODUCT_ID IN (
              SELECT AA.PRODUCT_ID
               FROM (
                           SELECT A.PRODUCT_ID, A.RATE_POINT, A.PRODUCT_CATEGORY
                          , (CASE @vproduct_category WHEN A.PRODUCT_CATEGORY
                                    THEN @rownum:=@rownum+1 ELSE @rownum:=1 END) RNUM
                          , (@vproduct_category:=A.PRODUCT_CATEGORY) CATEGORY
                           FROM  (
                                                SELECT P.PRODUCT_ID,
                                                            SUM(S.QUANTITY * CASE S.RATE_POINT
                                                                                    WHEN 1 THEN -1 ELSE 1 END) RATE_POINT,
                                                            P.PRODUCT_CATEGORY
                                                 FROM ORDERINFO O, SOLD_PRODUCT S, PRODUCT P
                                                WHERE O.ORDER_STATUS = '03'
                                                  AND O.ORDER_ID = S.ORDER_ID
                                                  AND S.PRODUCT_ID = P.PRODUCT_ID
                                                GROUP BY P.PRODUCT_ID, P.PRODUCT_CATEGORY
                                       ) A , (SELECT @vproduct_category:=0, @rownum:=0 FROM DUAL) B
               ORDER BY A.PRODUCT_CATEGORY ASC, A.RATE_POINT DESC
               ) AA
              WHERE AA.RNUM < 4
);
 INSERT INTO LOG VALUES('01',NOW());
END
```

7.2 Add product to cart

```
// 03: same CUST add same product at same restaurant
// 02: same CUST add diff product at same restaurant
// 01: same CUST add diff product at diff restaruant OR first add in to cart

 $query = "SELECT MAX(K.KIND) kind FROM (
                                        SELECT '03' KIND FROM ORDERINFO O, SOLD_PRODUCT S
                                        WHERE O.ORDER_ID = S.ORDER_ID
                                          AND O.REST_ID = $rest_id  AND O.CUST_ID = $cust_id
                                          AND O.ORDER_STATUS = '05 AND S.PRODUCT_ID = $product_id
                                        UNION ALL
```

```
                                    SELECT '02' KIND FROM ORDERINFO O, SOLD_PRODUCT S
                                    WHERE O.ORDER_ID = S.ORDER_ID
                                     AND O.REST_ID = $rest_id AND O.CUST_ID = $cust_id
                                     AND O.ORDER_STATUS = '05'
                                    UNION ALL
                                    SELECT '01' FROM  dual
                                    ) K;";

if ($row['kind'] == '03') {
            $query_03_01 = "UPDATE SOLD_PRODUCT SET QUANTITY = QUANTITY + 1
                                    WHERE PRODUCT_ID = $product_id
                                    AND ORDER_ID = (SELECT ORDER_ID
                                                    FROM ORDERINFO
                                                    WHERE  ORDER_STATUS='05'
                                                    AND CUST_ID = $cust_id AND REST_ID = $rest_id);";

 else if ($row['kind'] == '02') {
                        $query_02_02 = "INSERT INTO SOLD_PRODUCT (PRODUCT_ID, ORDER_ID, QUANTITY) VALUES
                                            ($product_id, (SELECT ORDER_ID FROM ORDERINFO where  ORDER_STATUS='05'
                                                                    and cust_id = $cust_id AND REST_ID = $rest_id), 1);";
                        $result_02_02 = mysqli_query($connectData, $query_02_02);

 else {
                        $query_01_01 = "INSERT INTO ORDERINFO (ORDER_STATUS, CUST_ID, REST_ID) VALUES ('05', $cust_id, $rest_id);";
                        $result_01_01 = mysqli_query($connectData, $query_01_01);
                        $query_01_02 = "INSERT INTO SOLD_PRODUCT (PRODUCT_ID, ORDER_ID, QUANTITY) VALUES
                                        ($product_id, (SELECT ORDER_ID ORDER_ID FROM ORDERINFO where  ORDER_STATUS='05'
                                          and cust_id = $cust_id AND REST_ID = $rest_id), 1);";
                        $result_01_02 = mysqli_query($connectData, $query_01_02);
 }
```