



Final Report

for

ThinkFood

By

ICS Group 06

Contents

1. Project Overview	4
2. Project Specifications	5
2.1 Function Requirements	5
2.2 Non-Function Requirements	5
2.3 Group Roles	6
3. Usability Guide	7
3.1 Customer	7
3.2 Restaurant	7
3.3 Administrator	16
4. Problems and Challenges	18
4.1 Managing tasks on schedule	18
4.2 Standards	18
4.3 GitLab	18
4.4 Analying requirements	18
5. Skill Learned	18
5.1 Cooperation	18
5.2 GitLab	18
5.3 MySQL and Workbench	18
5.4 PHP, HTML, CSS	18
6. Proposed Changes	19
6.1 Time management	19
6.2 Dedicated role assign	19
6.3 Avoid duplicate statements	Error! Bookmark not defined.
6.4 Avoid duplicate statements	Error! Bookmark not defined.
6.5 Avoid duplicate statements	Error! Bookmark not defined.
7. Code Snippets	19
7.1 Batch for recommend products	19
7.2 Add product to cartt	19

7.3 ppppppp Error! Bookmark not defined.

Table of Figures

Figure 1. System Architecture	4
Figure 2. Privacy Term Accpet.....	7
Figure 3. Login.....	7
Figure 4. Login	8
Figure 5. Product.....	8
Figure 6. Add to cart.....	9
Figure 7. Product Review	9
Figure 8. Order History.....	10
Figure 9. Review	10
Figure 10. Cart.....	11
Figure 11. Cart.....	11
Figure 12. Update a cart.....	12
Figure 14. Order	13
Figure 15. Payment	13
Figure 16. Order Detail.....	14
Figure 17. Order History.....	14
Figure 18. Restaurant manage	15
Figure 19. Administrator Page	16
Figure 20. Enroll a restaurant	16
Figure 21. Restaurant Page	17

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.

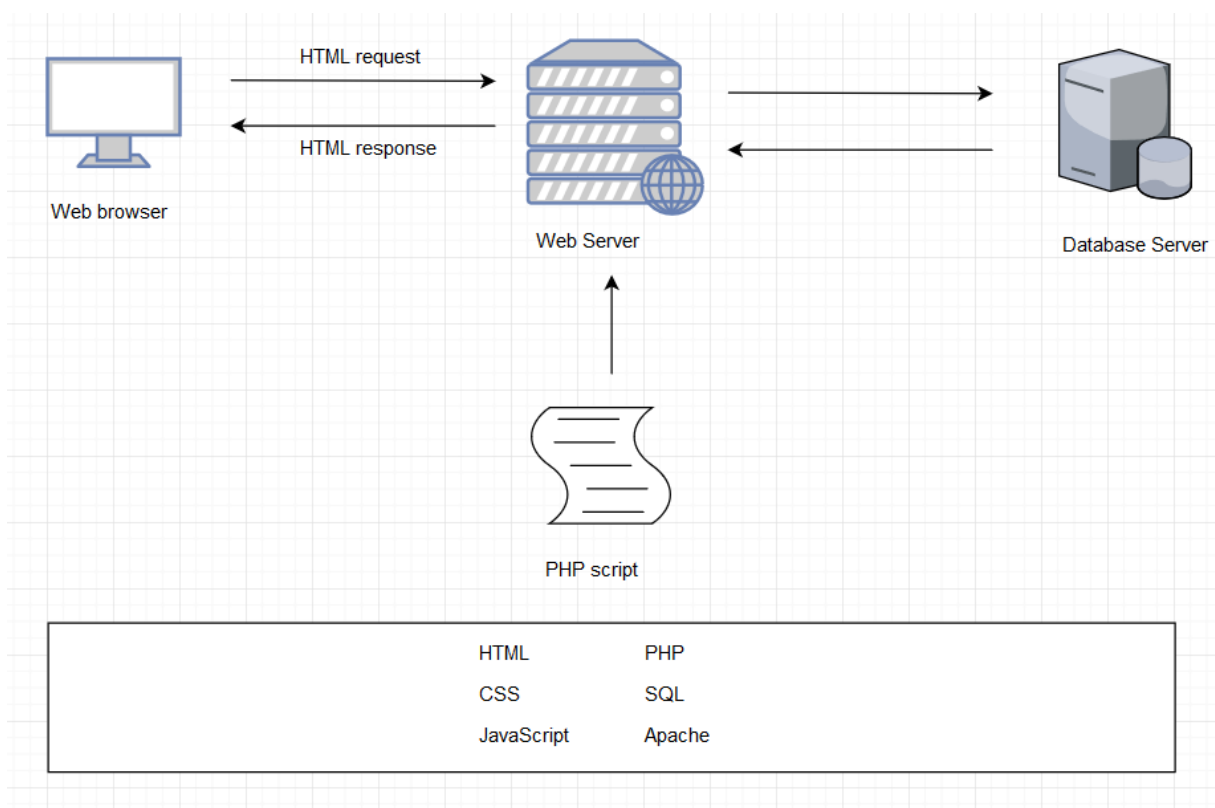


Figure 1. System Architecture

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.

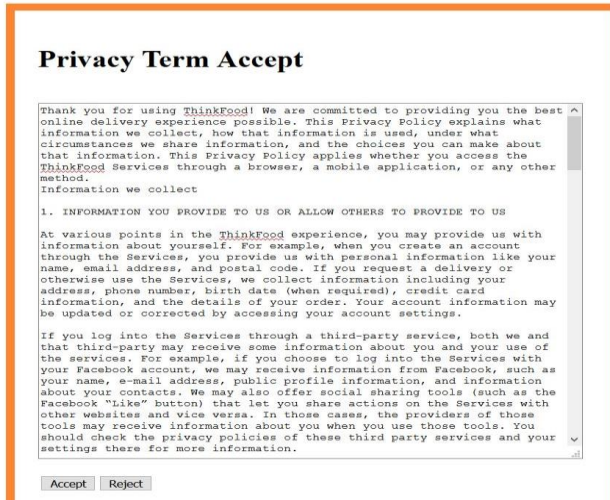


Figure 2. Privacy Term Accpet

Step 2. Fill the form and submit

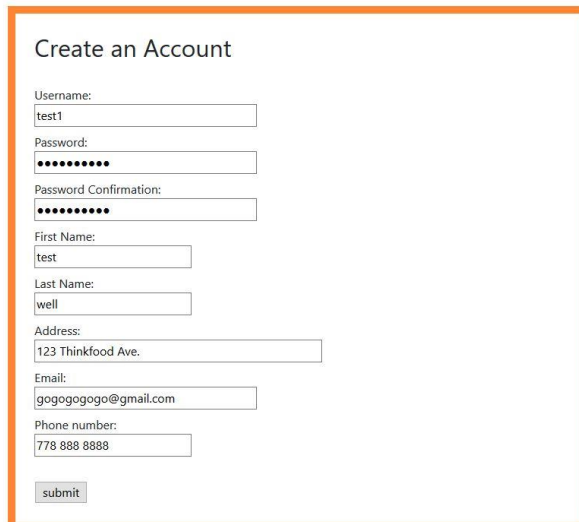


Figure 3. Login

3.1.2 Login

Step 1. Go to login page.

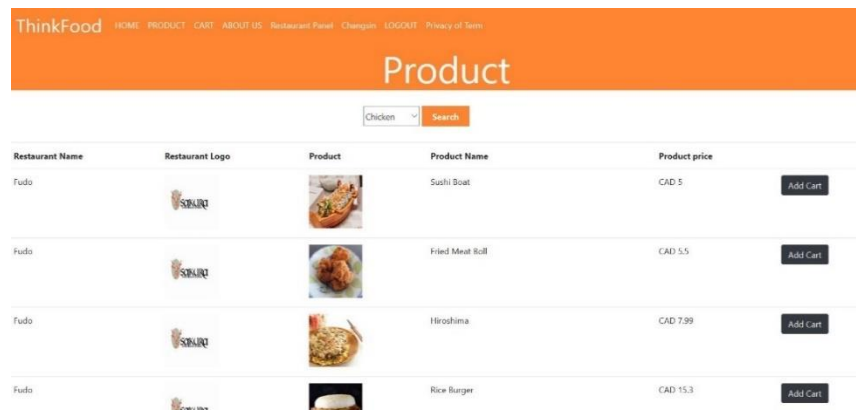
Step 2. Input username, password and then login.



Figure 4. Login

3.1.3 Product choose

Step 1. Go to product page











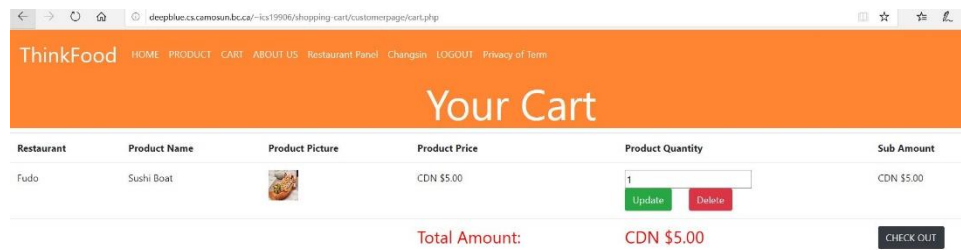
Restaurant Name	Restaurant Logo	Product	Product Name	Product price	
Fudo			Sushi Boat	CAD 5	Add Cart
Fudo			Fried Meat Roll	CAD 5.5	Add Cart
Fudo			Hiroshima	CAD 7.99	Add Cart
Fudo			Rice Burger	CAD 15.3	Add Cart

Figure 5. Product

Step 2. Add items to cart




Restaurant	Product Name	Product Picture	Product Price	Product Quantity	Sub Amount
Fudo	Sushi Boat		CDN \$5.00	1 Update Delete	CDN \$5.00
Total Amount:			CDN \$5.00		CHECK OUT

Figure 6. Add to cart

3.1.4 Review

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

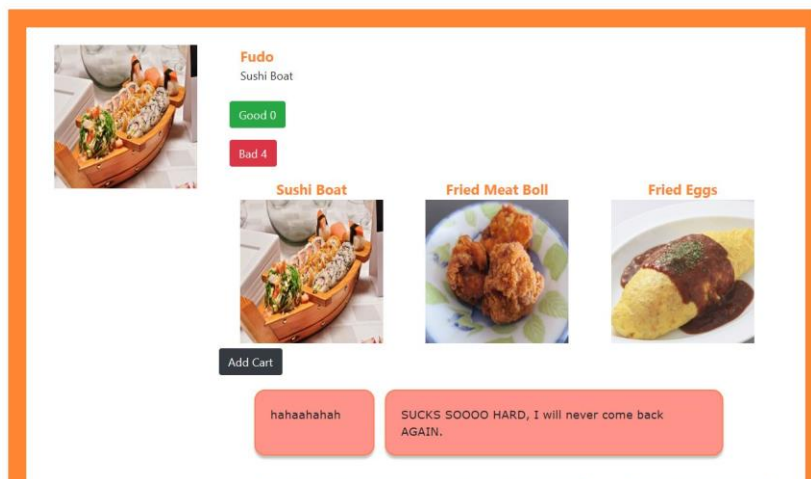
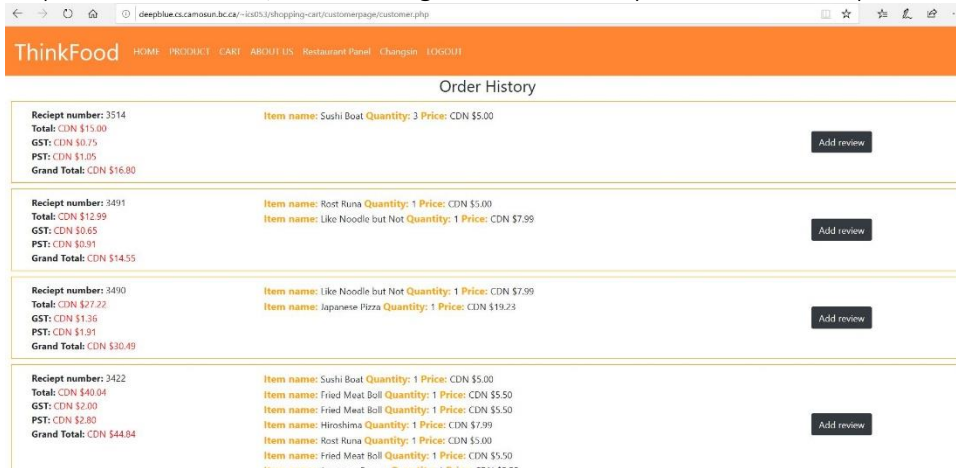


Figure 7. Product Review

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

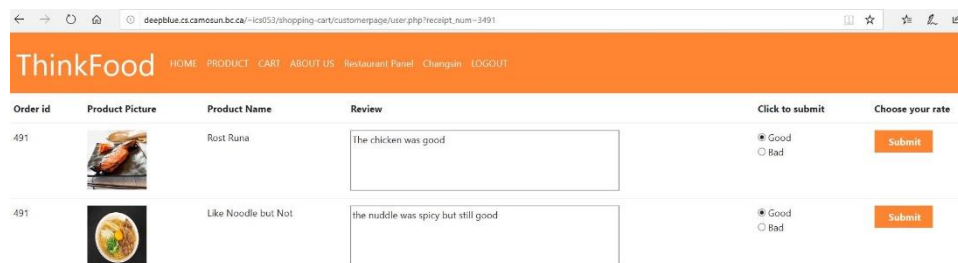


ThinkFood HOME PRODUCT CART ABOUT US Restaurant Panel Change Logout

Order History

Receipt number: 3514 Total: CDN \$15.00 GST: CDN \$0.75 PST: CDN \$1.05 Grand Total: CDN \$16.80	Item name: Sushi Boat Quantity: 3 Price: CDN \$5.00	Add review
Receipt number: 3491 Total: CDN \$12.99 GST: CDN \$0.65 PST: CDN \$0.91 Grand Total: CDN \$14.55	Item name: Rost Runa Quantity: 1 Price: CDN \$5.00 Item name: Like Noodle but Not Quantity: 1 Price: CDN \$7.99	Add review
Receipt number: 3490 Total: CDN \$27.22 GST: CDN \$1.36 PST: CDN \$1.91 Grand Total: CDN \$30.49	Item name: Like Noodle but Not Quantity: 1 Price: CDN \$7.99 Item name: Japanese Pizza Quantity: 1 Price: CDN \$19.23	Add review
Receipt number: 3422 Total: CDN \$40.04 GST: CDN \$2.00 PST: CDN \$2.80 Grand Total: CDN \$44.84	Item name: Sushi Boat Quantity: 1 Price: CDN \$5.00 Item name: Fried Meat Roll Quantity: 1 Price: CDN \$5.50 Item name: Fried Meat Roll Quantity: 1 Price: CDN \$5.50 Item name: Hiroshima Quantity: 1 Price: CDN \$7.99 Item name: Rost Runa Quantity: 1 Price: CDN \$5.00 Item name: Fried Meat Roll Quantity: 1 Price: CDN \$5.50	Add review

Figure 8. Order History



ThinkFood HOME PRODUCT CART ABOUT US Restaurant Panel Change Logout


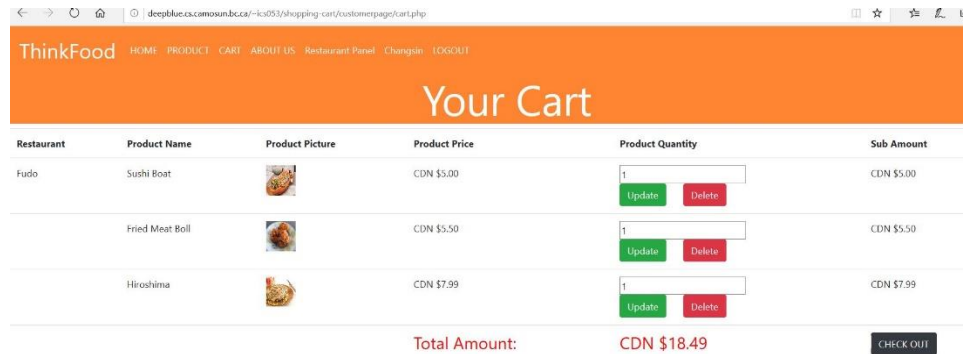
Order id	Product Picture	Product Name	Review	Click to submit	Choose your rate
3491		Rost Runa	The chicken was good	<input checked="" type="radio"/> Good <input type="radio"/> Bad	Submit
3490		Like Noodle but Not	the niddle was spicy but still good	<input checked="" type="radio"/> Good <input type="radio"/> Bad	Submit

Figure 9. Review

3.1.5 Cart

Step 1. Go to cart to check the selected product






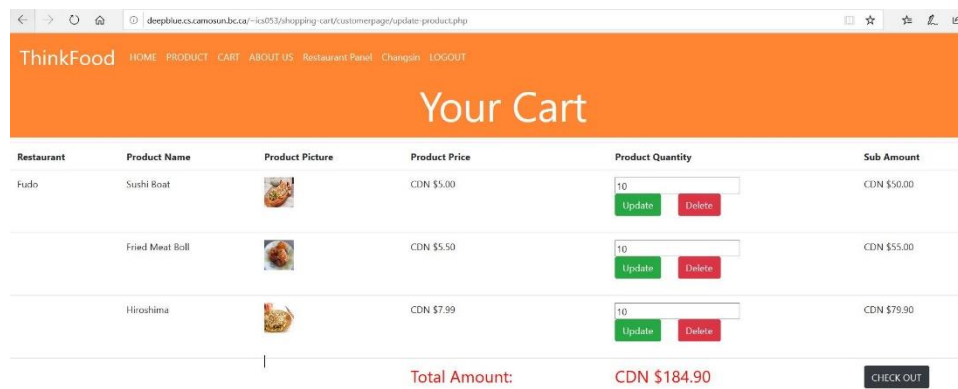
Restaurant	Product Name	Product Picture	Product Price	Product Quantity	Sub Amount
Fudo	Sushi Boat		CDN \$5.00	1	CDN \$5.00
	Fried Meat Roll		CDN \$5.50	1	CDN \$5.50
	Hiroshima		CDN \$7.99	1	CDN \$7.99
Total Amount:			CDN \$18.49	CHECK OUT	

Figure 10. Cart

Step 2. Change each item's quantity






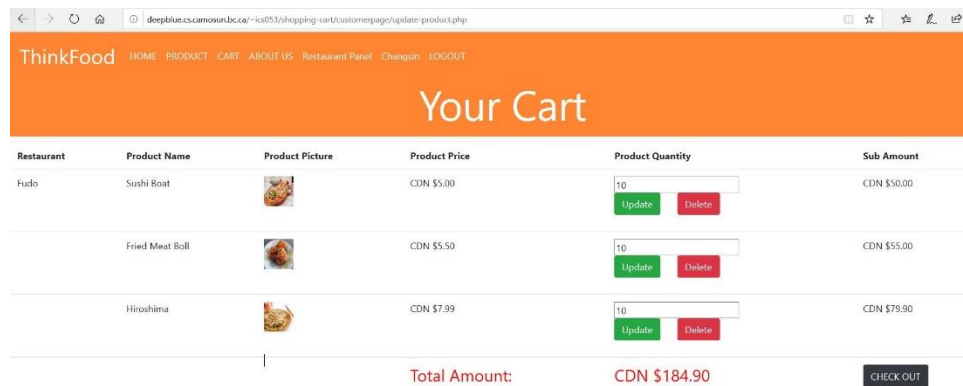
Restaurant	Product Name	Product Picture	Product Price	Product Quantity	Sub Amount
Fudo	Sushi Boat		CDN \$5.00	10	CDN \$50.00
	Fried Meat Roll		CDN \$5.50	10	CDN \$55.00
	Hiroshima		CDN \$7.99	10	CDN \$79.90
Total Amount:			CDN \$184.90	CHECK OUT	

Figure 11. Cart

Step 3. Update quantity or delete a product






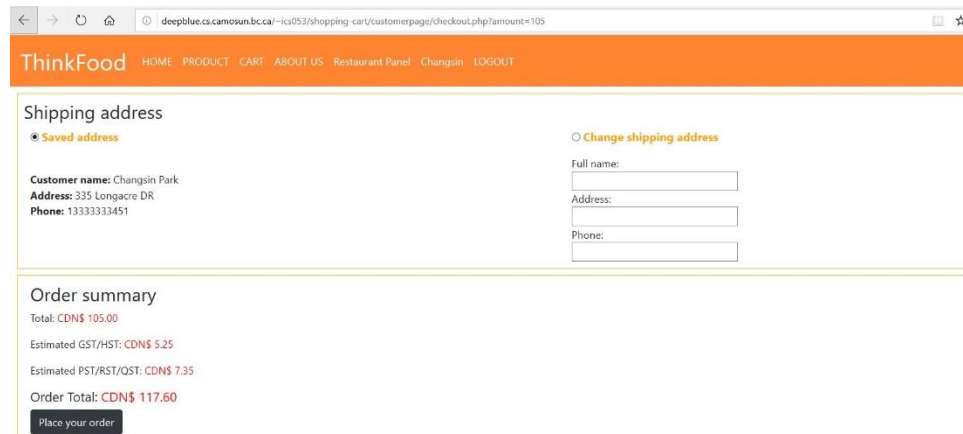
Restaurant	Product Name	Product Picture	Product Price	Product Quantity	Sub Amount
Fudo	Sushi Roast		CDN \$5.00	10 Update Delete	CDN \$50.00
	Fried Meat Bowl		CDN \$5.50	10 Update Delete	CDN \$55.00
	Hiroshima		CDN \$7.99	10 Update Delete	CDN \$79.90
			Total Amount:	CDN \$184.90	CHECK OUT

Figure 12. Update a cart

Step 4. Press check out button



Shipping address

• **Saved address**

Customer name: Changsin Park
Address: 335 Longacre DR
Phone: 13333333451

○ **Change shipping address**

Full name:
Address:
Phone:

Order summary

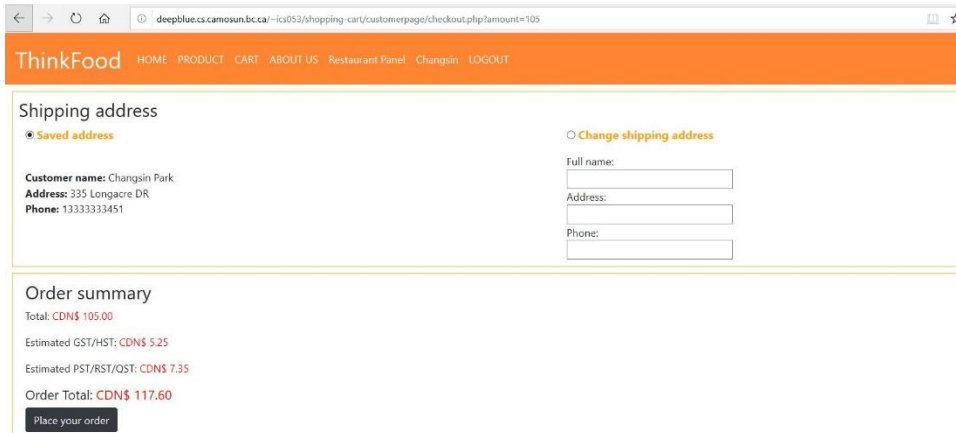
Total: CDN\$ 105.00
Estimated GST/HST: CDN\$ 5.25
Estimated PST/RST/OST: CDN\$ 7.35
Order Total: **CDN\$ 117.60**

[Place your order](#)

Figure 13. Check out

3.1.6 Check out

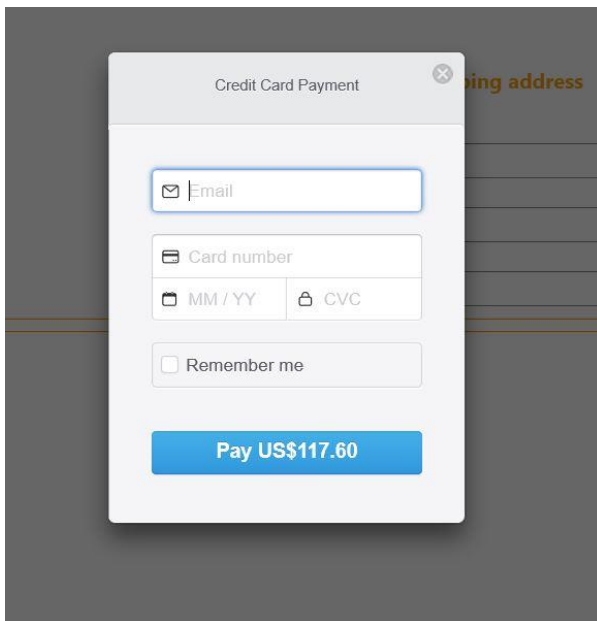
Step 1. In the checkout page, users should be able to modify their shipping address and check the info



The screenshot shows the ThinkFood checkout page. The top navigation bar includes links for HOME, PRODUCT, CART, ABOUT US, Restaurant Panel, Changsin, and LOGOUT. The main content area is divided into two sections. The first section, titled 'Shipping address', has a radio button for 'Saved address' (selected) and a radio button for 'Change shipping address'. Below the 'Saved address' radio button, the customer's information is displayed: 'Customer name: Changsin Park', 'Address: 335 Longacre DR', and 'Phone: 13333333451'. To the right of this information are input fields for 'Full name:', 'Address:', and 'Phone:'. The second section, titled 'Order summary', displays the following information: 'Total: CDN\$ 105.00', 'Estimated GST/HST: CDN\$ 5.25', 'Estimated PST/RST/OST: CDN\$ 7.35', and 'Order Total: CDN\$ 117.60'. At the bottom of the order summary is a button labeled 'Place your order'.

Figure 14. Order

Step 2. Click 'Place your order' button



The screenshot shows a 'Credit Card Payment' modal form. The form has a title bar with a close button. It contains the following fields: an 'Email' field with an envelope icon, a 'Card number' field with a card icon, and two fields for 'MM / YY' and 'CVC' with calendar and lock icons respectively. There is a 'Remember me' checkbox. At the bottom of the form is a blue button labeled 'Pay US\$117.60'.

Figure 15. Payment

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

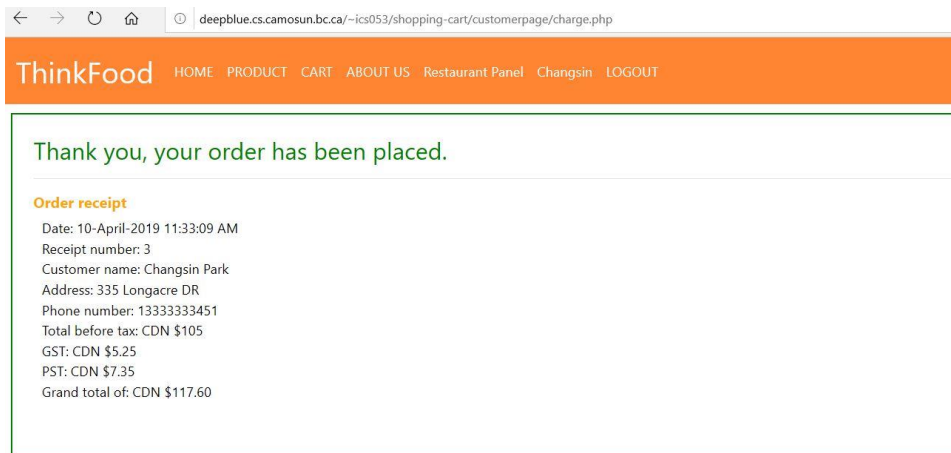


Figure 16. Order Detail

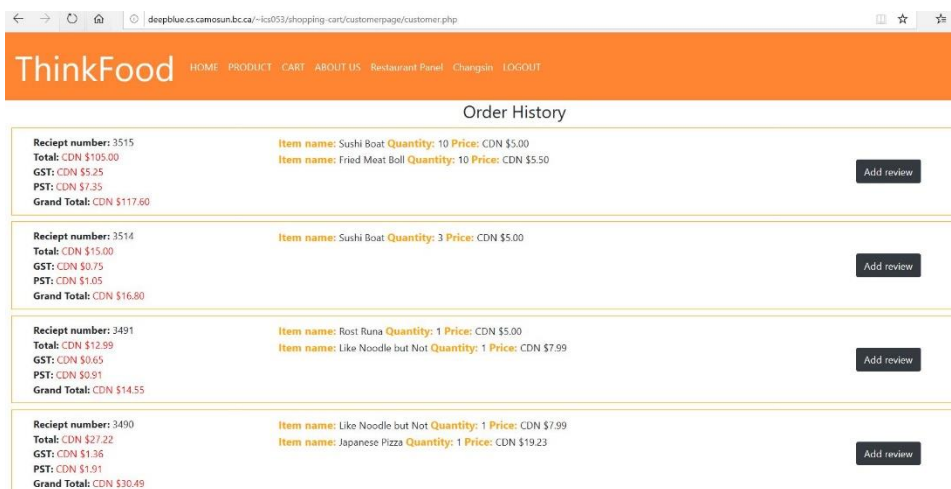


Figure 17. Order History

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



Order number	Order	Quantity	Time recieve	Customer name	Address	Phone number	Order status
514	Sushi Boat	3	2019-04-10 11:08:37	Changsin Park	335 Longacre DR	13333333451	In progress
515	Sushi Boat	10	2019-04-10 11:31:01	Changsin Park	335 Longacre DR	13333333451	In progress
515	Fried Meat Boll	10	2019-04-10 11:31:01	Changsin Park	335 Longacre DR	13333333451	In progress

Figure 18. Restaurant manage

3.3 Administrator

3.3.1 Administrator page

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

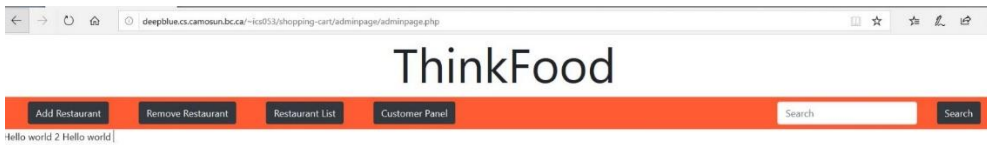


Figure 19. Administrator Page

3.3.2 Add a restaurant

Step1. Fill out mandatory items

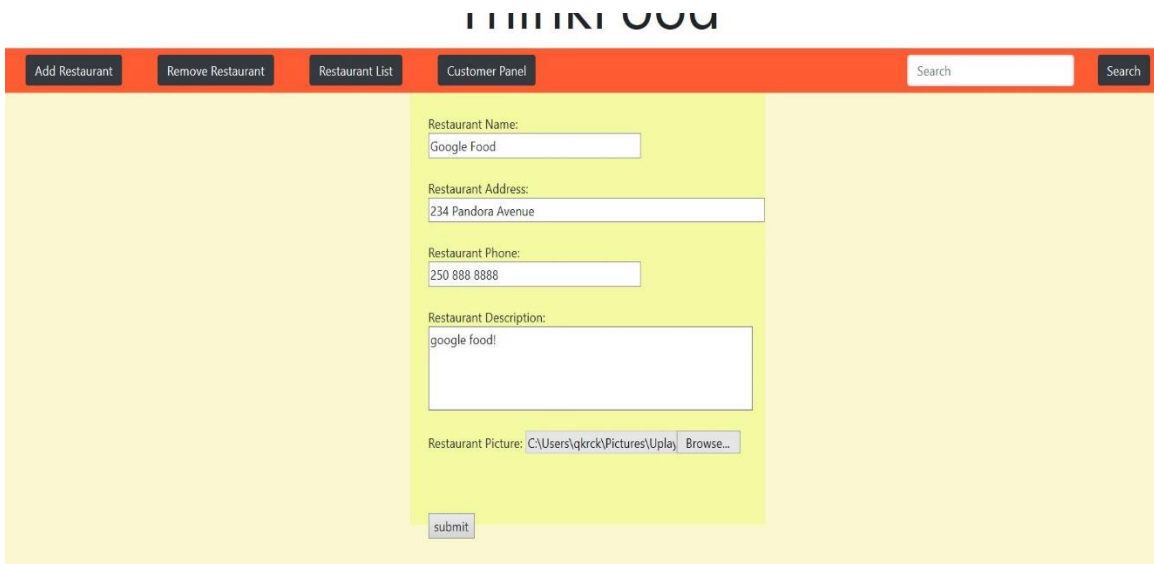
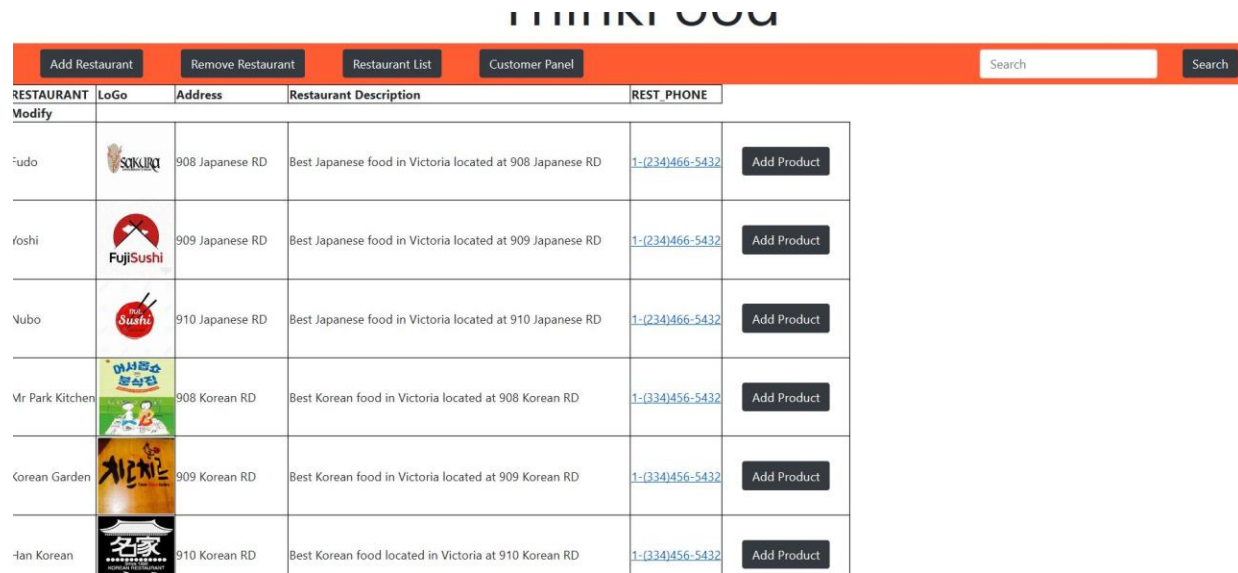


Figure 20. Enroll a restaurant

3.3.3 Add a product

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









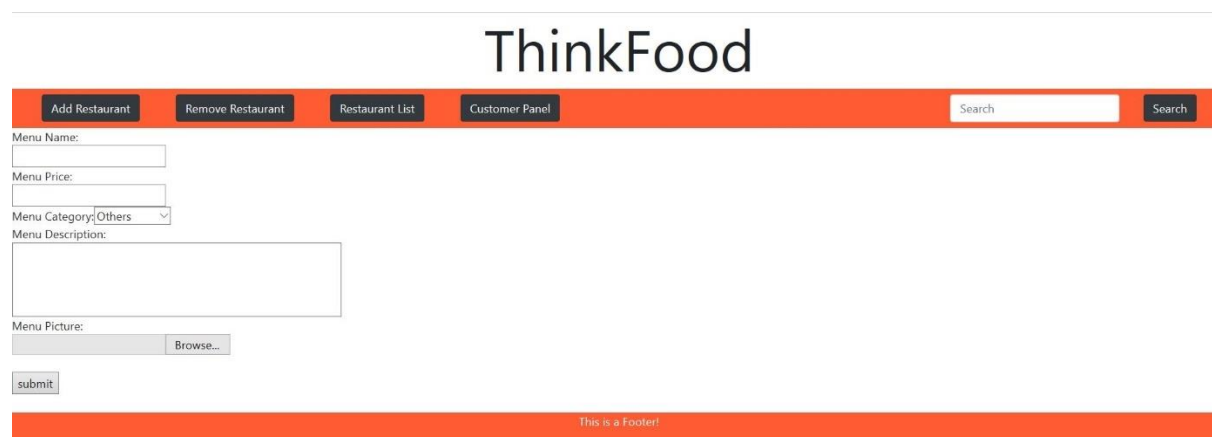
RESTAURANT	LoGo	Address	Restaurant Description	REST_PHONE	
Modify					
Yudo		908 Japanese RD	Best Japanese food in Victoria located at 908 Japanese RD	1-(234)466-5432	Add Product
Yoshi		909 Japanese RD	Best Japanese food in Victoria located at 909 Japanese RD	1-(234)466-5432	Add Product
Yubo		910 Japanese RD	Best Japanese food in Victoria located at 910 Japanese RD	1-(234)466-5432	Add Product
Mr Park Kitchen		908 Korean RD	Best Korean food in Victoria located at 908 Korean RD	1-(334)456-5432	Add Product
Korean Garden		909 Korean RD	Best Korean food in Victoria located at 909 Korean RD	1-(334)456-5432	Add Product
Han Korean		910 Korean RD	Best Korean food located in Victoria at 910 Korean RD	1-(334)456-5432	Add Product

Figure 21. Restaurant Page

Step2. Fill out mandatory items



ThinkFood

[Add Restaurant](#)
[Remove Restaurant](#)
[Restaurant List](#)
[Customer Panel](#)

[Search](#)

Menu Name:

Menu Price:

Menu Category: Others

Menu Description:

Menu Picture:

[Browse...](#)

[submit](#)

This is a Footer!

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 change

- 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 change, 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 change 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 chagement. 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);
}
}

```