HO CHI MINH CITY NATIONAL UNIVERSITY

INTERNATIONAL UNIVERSITY

----------⁂----------

PRINCIPLES OF DATABASE MANAGEMENT

Semester 2, 2020-2021

Instructor: Assoc. Prof. Nguyễn Thị Thúy Loan

PROJECT REPORT:

SENTIMENT ANALYSIS FOR PRODUCT RATING (Topic 2)

Group members:

Nguyễn Lãm – ITITIU18255
Nguyễn Thị Hoài An – ITDSIU18024
Phan Ánh Ngọc – ITITWE19018

Contents

List of tables

List of figures

I. Acknowledgement

We would like to thank Assoc. Prof. Nguyen Thi Thuy Loan for the knowledge we learned from this course. This course has been incredibly informative; we believe that the knowledge we obtained from it will be practically used as we grow to be computer science and data science bachelors. We wish Ms. Nguyen Thi Thuy Loan all the best in future as we know that she wishes us the same.

Nevertheless, we consider this project our accomplishment since we successfully managed to finish it in less than 3 months. Our group originally consists of 4 students (Nguyễn Lãm, Nguyễn Thị Hoài An, Phan Ánh Ngọc and Phan Nguyễn Gia Bảo); however, Phan Nguyễn Gia Bảo – ITITIU19006 had confirmed with the team that he would no longer work as a member from May 8th, 2021. Phan Nguyễn Gia Bảo had not yet to make any contribution from the start of the project until his departure from the team. Us three remaining had efficiently worked as a united team, and we are proud to say that each of the three had made equal contribution to the project.

No.	Full name	Student ID	Contribution
1	Nguyễn Lãm	ITITIU18255	$\frac{1}{3}$ of workload
2	Nguyễn Thị Hoài An	ITDSIU18024	$\frac{1}{3}$ of workload
3	Phan Ánh Ngọc	ITITWE19018	$\frac{1}{3}$ of workload

Table 1: Table of contribution

II. Introduction

In this modern time of Industry 4.0, it is common sense for consumers to leave their thoughts on products or search for other users' reviews and product ratings online before purchasing a product. Many of the product ratings online are conducted using sentiment words in users' comments.

Took notice of the current trend, the team members came into consent after discussions that the topic ***"Sentiment Analysis for Product Rating"*** should be chosen for this course group project. Specifically, a website for ***Food and Drinks Destination Rating*** was developed. This project serves as a website where users upload their point of view on cafeterias, view other users' opinions and refer to the destination's rating conducted by the website. The website uses sentimental analysis methodology to rate the restaurants. As a result, users will gain awareness of the cafeterias through the comments and ratings.

1. Project Objective

This project's objective is to create a realistic reviewing website that can be further developed in the future. The system must be well-designed to take control of data storage, including the information on food and drinks destinations, users' comments on them and its rating. A reviewing online platform the basic objective of this project.

2. Project Overview

The main concept of this website is to allow users to leave their thoughts on cafeterias and refer to other users' comments and rating conducted by system. This website is developed with the hope to provide users a better experience using the cafeteria's services. The data about the destinations and users' information are stored on the database system.

The system was designed for two types of users:

- Users: user registers for an account to make comments under posts, to view the cafeterias' ratings and to refer to other users' point of view.
- Admin: admin uploads posts about cafeterias for users to make comments on and controls the database system. Only admin can make updates on the data table that

stores sentiment words and its corresponding weight. When admin adds information about a food and drinks destination, system will create a post based on the input information. Users then can leave their thoughts on experiencing the destination's service, or they can just view others' past opinions as references if they have not yet been to the place and are looking for reliable reviews. Based on comments uploaded by users, system will use the sentiment analysis methodology to rate the cafeterias.

The sentiment analysis methodology mentioned above can be easily understood as:

- Admin adds sentiment words and its corresponding weights into data table "sentimentalword"
- Comments made by users are saved in the data table "comment".
- System queries the comments to check if it contains sentiment words in table "sentimentalword". If there exist sentiment words in the comments, those sentiment words will be saved in column "word" ofthe comment data tuple.

3. Tools

Front-end: HTML, CSS, JavaScript

- HTML:

  HTML stands for Hypertext Markup Language. It helps users create and structure elements of web pages or applications, divide paragraphs, headings, links, blockquotes, and so on. HTML is not a programming language, meaning it cannot create "dynamic" functions. It is just like Microsoft Word, used to design layout and format web pages.

- CSS:

  CSS is the style language for websites - Cascading Style Sheet language. It is used to style and style elements that are written in markup languages, such as HTML. It can control the format of multiple websites at the same time to save effort for web writers. It distinguishes the look of a web page from its main content by controlling the layout, colors, and fonts.

- JavaScript:

  It gives the site additional functionality that isn't otherwise achievable with HTML and CSS alone. JavaScript allows webpages to respond to user activity and dynamically update themselves, and all without requiring a page reload to change its appearance.

Back-end: PHP, XAMPP

- PHP:

PHP (stand for Personal Home Page) is an open source scripting language that is commonly used to create web applications that run on the server. PHP code can be embedded in HTML pages by using PHP.

- XAMPP:

XAMPP is basically a local host or a local server. It meant to be used as a development tool, to permit software developers to run the website on their own computers or personal laptop or test the clients before uploading it to the remote web server. This XAMPP server software provides the suitable environment for testing MYSQL, PHP, Apache and Perl projects on the local computer.

1. We download XAMPP and install it.
2. Open XAMPP :

   Apache -> Start

   MySQL -> Start
3. Access localhost port 80 with link: localhost/hctphone and access phpMyAdmin with link: localhost/dashboard -> click phpMyAdmin

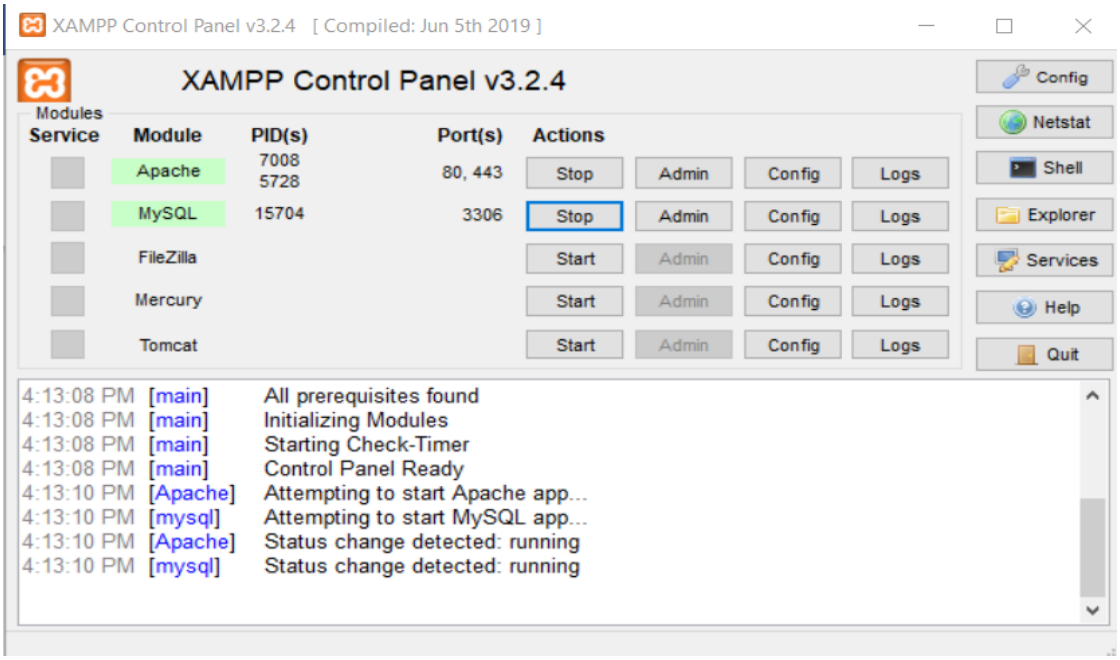


Figure 1: XAMPP Control Panel

III. ERD Model and Relational Algebra

1. ERD Model

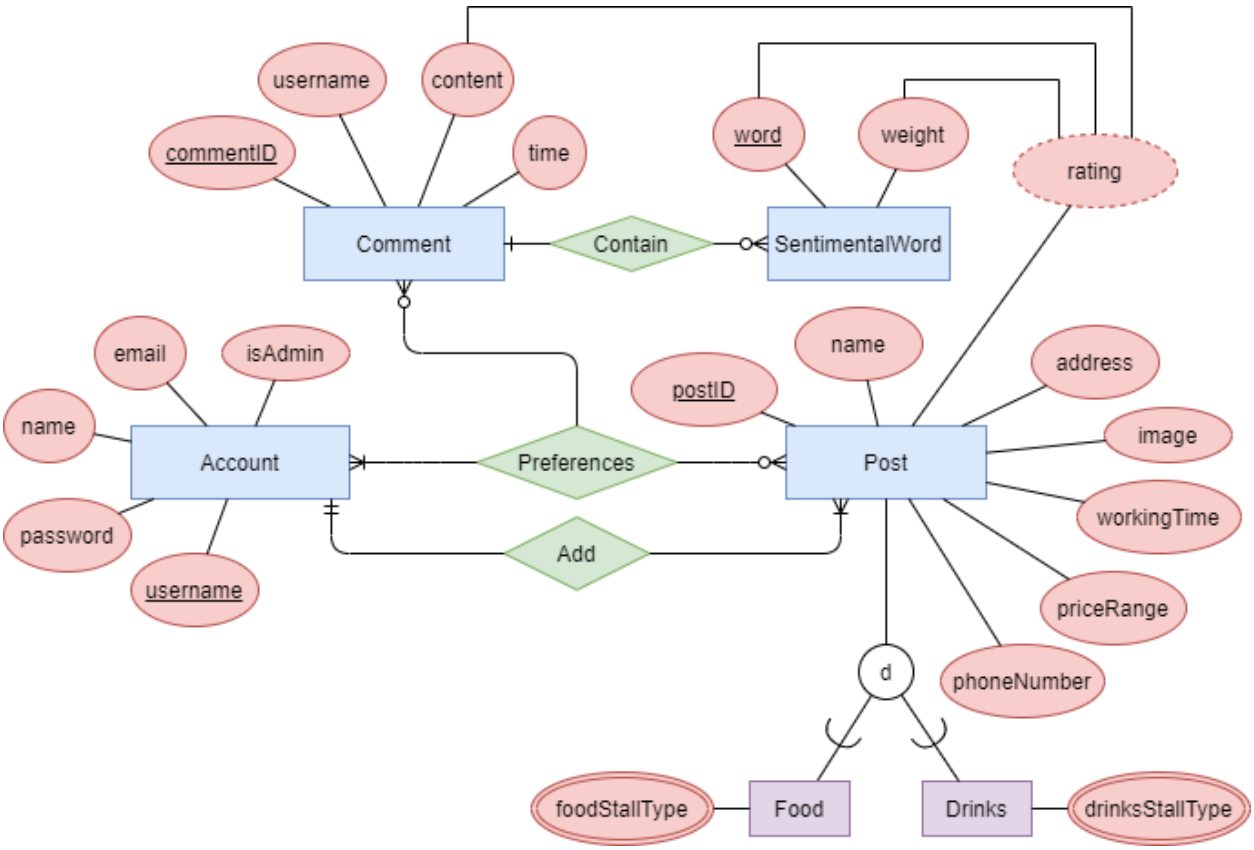


Figure 2: ERD Model

2. Relationship Explanation:

a. Account – Comment: One-or-Many to Zero-or-Many:

There must be at least one account (the first user of this website is undoubtedly the website admin). An account can make zero to many comments: those which do not make any comment indicate that the users created the accounts only for the purpose of viewing the ratings conducted by the website and thoughts of experienced users on the destinations.

b. Account – Post: One-or-Many to Zero-or-Many

There must be at least one account (the first user of this website is undoubtedly the website admin). An account can leave their comments on zero to many posts: those which do not make any comment on any post indicate that the users created the accounts

only for the purpose of viewing the ratings conducted by the website and thoughts of experienced users on the destinations.

c. Account – Post: One-and-Only-One to One-or-Many

Only one account can add posts to the website for users to leave their thoughts on: that one special account is undoubtedly the admin. The admin can decide not to make any post addition or to make as many of them as they want to. Systematically, the website can still exist even if there is not any post for users to view or make comments on; however, that would indicate that the website is not in use and there is no reason why such website should exist. That is why this relationship is a One-and-Only-One to One-to-Many: there should be at least one default post for user to leave their comments on.

d. Comment – Post: Zero-or-Many to One-or-Many:

Explained above in the relationship "*Account – Post: One-and-Only-One to One-or-Many"*, there should be at least one post on the website. On the other hand, it is not compulsory for a post to have comments on it.

e. Comment – SentimentalWord: One to Zero-or-Many

A comment made by user can basically consists of meaningless words, emotion icons or even random characters. A comment can also include many adjectives indicate sentiments.

3. Relational Algebra:

List total restaurants with all food type from starts to know:

$$\pi_{foodName,\ foodStallType,\ postID}(food \bowtie_{food.postID=foodstalltype.postID} foodstalltype)$$

List restaurants name which has rating higher from 5 points:

$$\pi_{foodName}(\sigma_{rate>5}(food))$$

List the comment which has the same username:

$$\pi_{content}(\sigma_{comment\ a.username=comment\ b.username}\ (food))$$

List the sentimental word that has the same weight equal to 8:

$$\pi_{word}(\sigma_{weight=8}\ (sentimentalword))$$

IV. System

4.1. Database

4.1.1. Create database system on phpMyAdmin

In order to create a database system for the website, choose **New** and input **Database Name.** Next, choose **utf8_unicode_ci**.

In this project, database named '*foodratingweb*' is generated

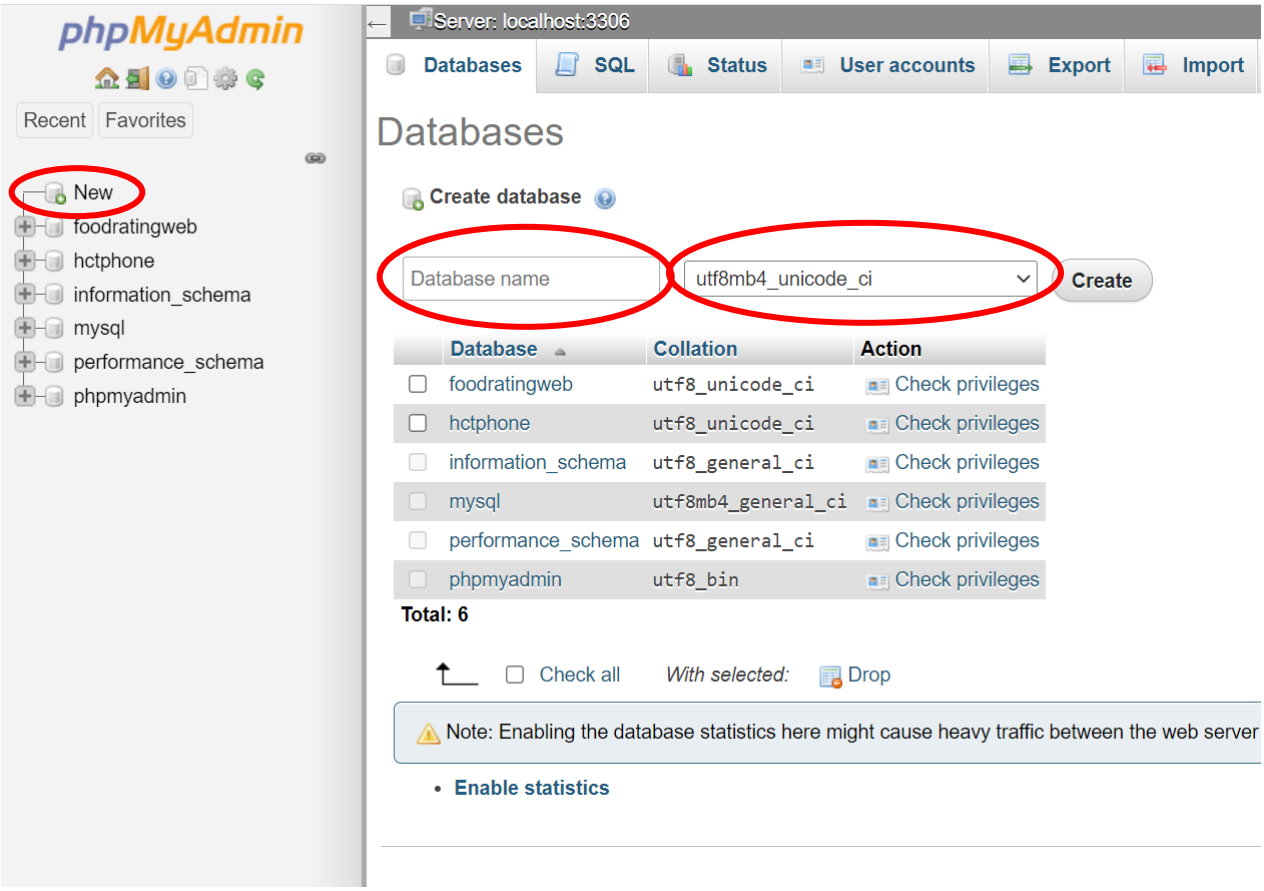


Figure 3: Create database system

When database system has already been created, click **New button** or **import** existing database file. All the tables will be shown on the left-hand side.

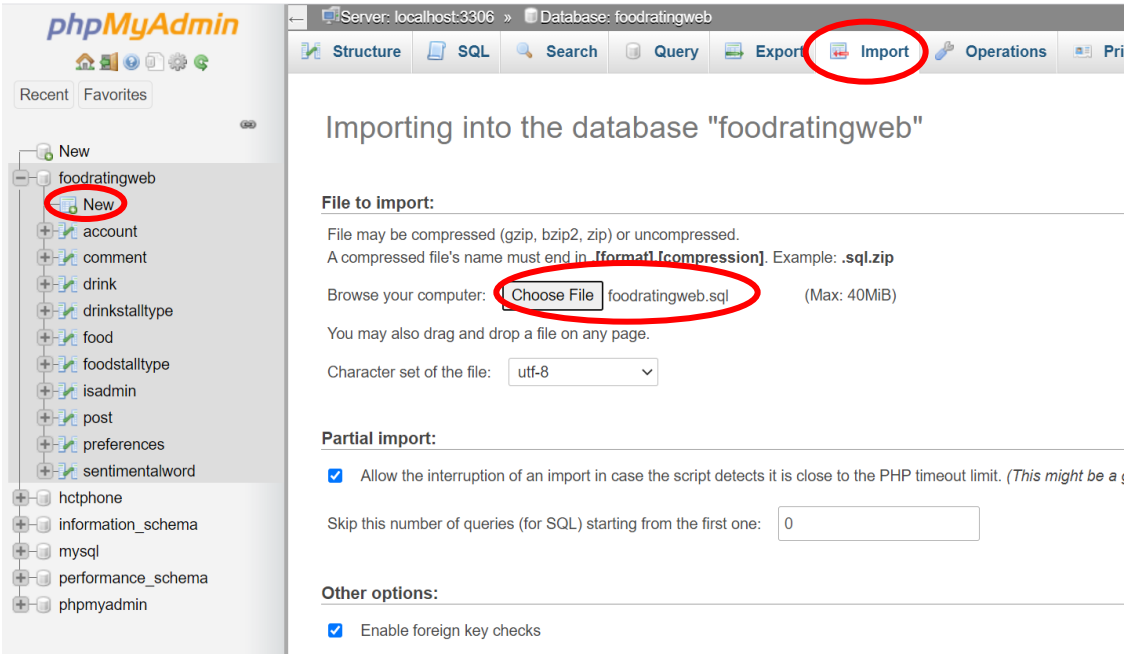


Figure 4: Create tables for database

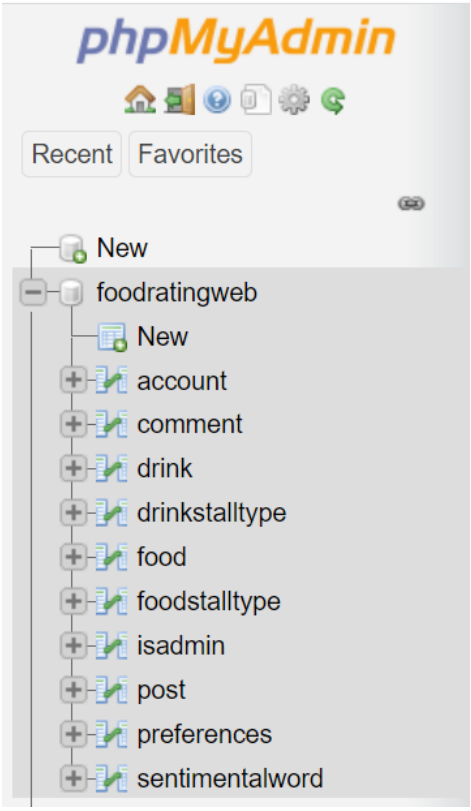


Figure 5: Tables in database

To see and edit all the information, including relationship of database, of the choosen table and edit, choose **Structure** section and **Table Structure**.

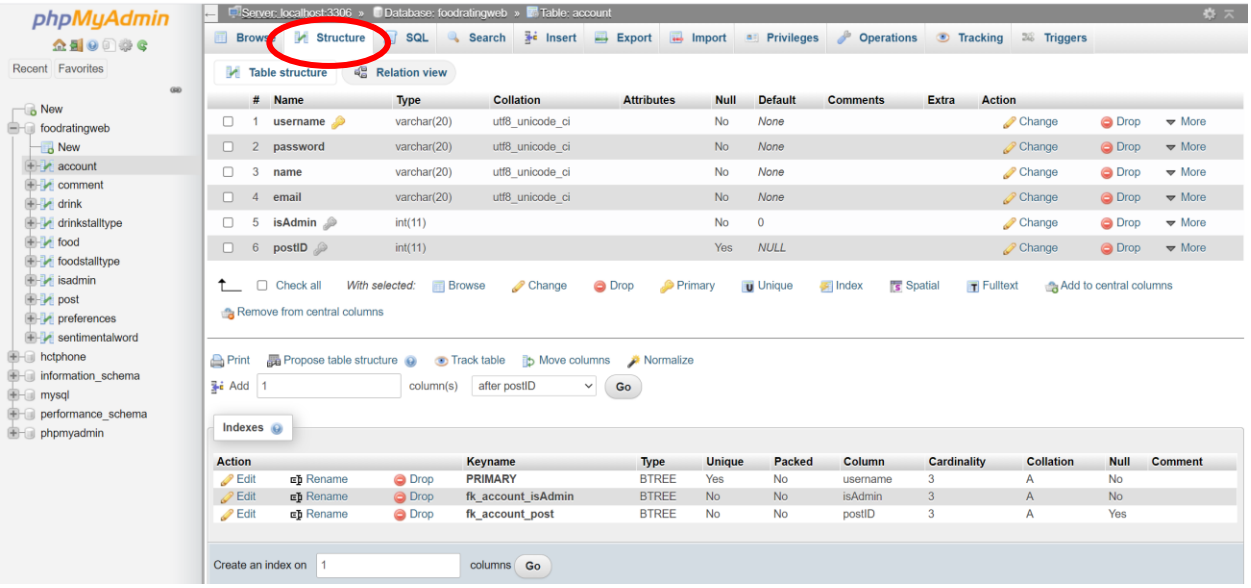


Figure 6: Detail on table

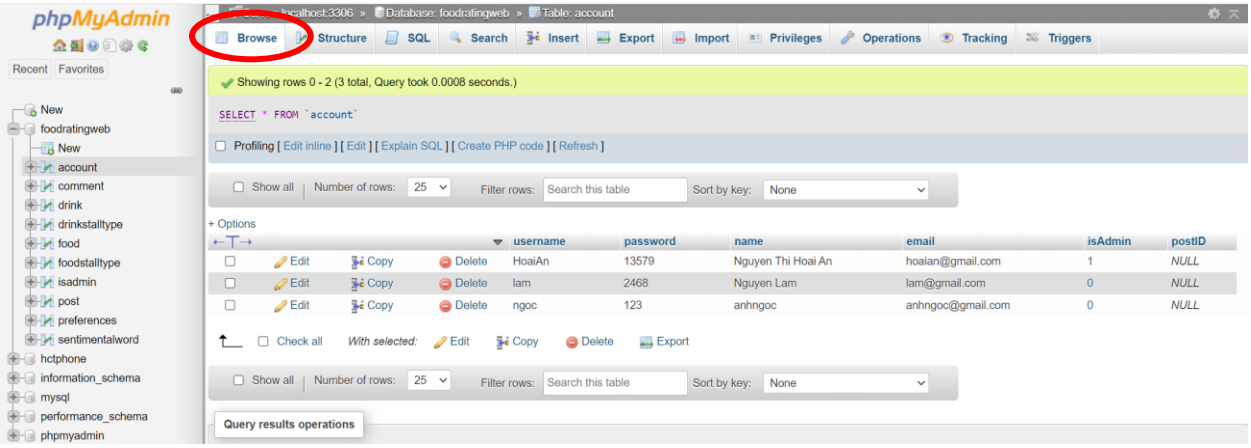


Figure 7: Value of tables

To see all the values of each column, choose Browse section

4.1.2. Table detail

a. *account* table

Figure 8: account table

- username: username is set as the primary key because users are requested to have a distinct username so that others users can distinguish easily
- password: users need to input password to log in
- name: users' full name for full information
- email: needs for sending notification about the post, comment or websit
- isAdmin: to check whether that person is a user or an admin. If a person is an admin, he/she enables to use addition function. The isAdmin has the foreign key to **isadmin** table
- postID: foreign key relation to **post** table so that system can control whose post is.

b. *isadmin* table

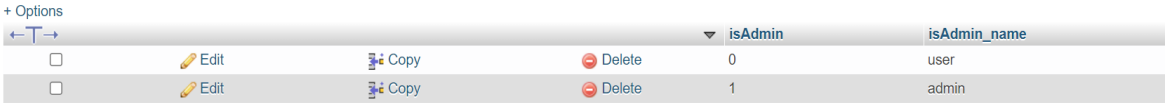


Figure 9: isadmin table

- isAdmin: determine which account is admin or user from account table, 0 for user, 1 for admin
- isAdmin_name: type of user name
- commentID: primary key
- username: to show whose comment is
- content: store the comments of users

- time: store the time users post their comment to the posts
- word: abstract words inside the comments based on the existing list of words in **sentimental** table

c. *post* table

	postID	name	rating	address	image	workingTime	priceRange	phoneNumber
Edit Copy Delete	1	D'maris	8	Nguyen Dinh Chieu, District 3	img/dmaris.jpg	10am - 9pm	45-90	023145
Edit Copy Delete	2	Midori Coffee	7	Mac Dinh Chi, District 1	img/midori.jpg	7am-9pm	10-20	94758564

Figure 10: post table

- postID: primary key
- name: name of places
- rating: based on words of users' comments to compute the score automatically by coding function (sentimental table content words and score for each word)
- image: store image url that users choose from their source
- workingTime: time that places open and close
- priceRange: range of price
- phoneNumber: phone number of that place

d. *food* table

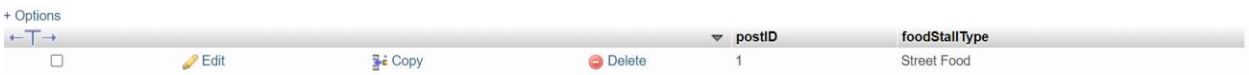
	postID	rating	address	image	workingTime	priceRange	phoneNumber	foodName
Edit Copy Delete	1	8	Nguyen Dinh Chieu, District 3	img/dmaris.jpg	10am - 9pm	45-90	023145	D'maris

Check all With selected: Edit Copy Delete Export

Figure 11: food table

- Because food is a subclass of post so food has every column that the post table has

e. *Foodstalltype* table



	postID	foodStallType
Edit Copy Delete	1	Street Food

Figure 12: foodstalltype table

- postID: belong to post
- foodStallType: store types of food places

f. *drink* table

Figure 13: drink table

- Because drink is a subclass of post so food has every column that the post table has

g. *drinkstalltype* table

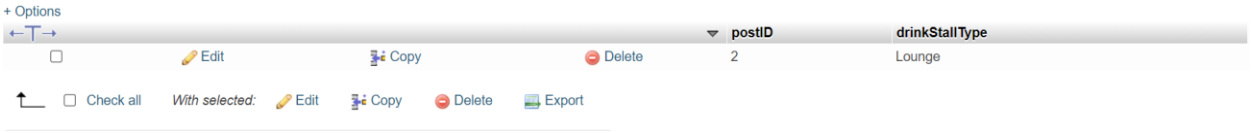


Figure 14: drinkstalltype table

- postID: belong to post
- drinkStallType: store types of food places

h. *comment* table

Figure 15: comment table

- commentID: primary key
- username: username to show which users commented
- content: store the comments of users
- time: the time users posted comments
- word: abstract words from content (abstacting based on the word list in **sentimental** table) by coding function

i. *preferences* table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	postID	int(11)			No	*None*			Change Drop More
2	username	varchar(20)	utf8_unicode_ci		No	*None*			Change Drop More
3	commentID	int(11)			No	*None*			Change Drop More

Figure 16: preferences table

- 'preferences' is the table that link table comment, account and post together so it contains primary keys of those three tables, which are postID, username, and comment ID. Primary keys also be the foreign key link to the corresponding tables

j. *sentimental* table

+ Options

				word	weight
	Edit	Copy	Delete	acceptable	8
	Edit	Copy	Delete	adorable	10
	Edit	Copy	Delete	agreeable	10
	Edit	Copy	Delete	amazing	10
	Edit	Copy	Delete	appetizing	10
	Edit	Copy	Delete	attractive	10
	Edit	Copy	Delete	awesome	10
	Edit	Copy	Delete	beautiful	10
	Edit	Copy	Delete	charming	10
	Edit	Copy	Delete	clean	10
	Edit	Copy	Delete	considerate	10
	Edit	Copy	Delete	cool	8
	Edit	Copy	Delete	courteous	10
	Edit	Copy	Delete	cute	10
	Edit	Copy	Delete	delicious	10
	Edit	Copy	Delete	delightful	10

Figure 17: sentimental table

This table contains list of **words** and the **weight** that made for counting the score for rating.

4.2. Main function

4.2.1. User pages

a. Home page

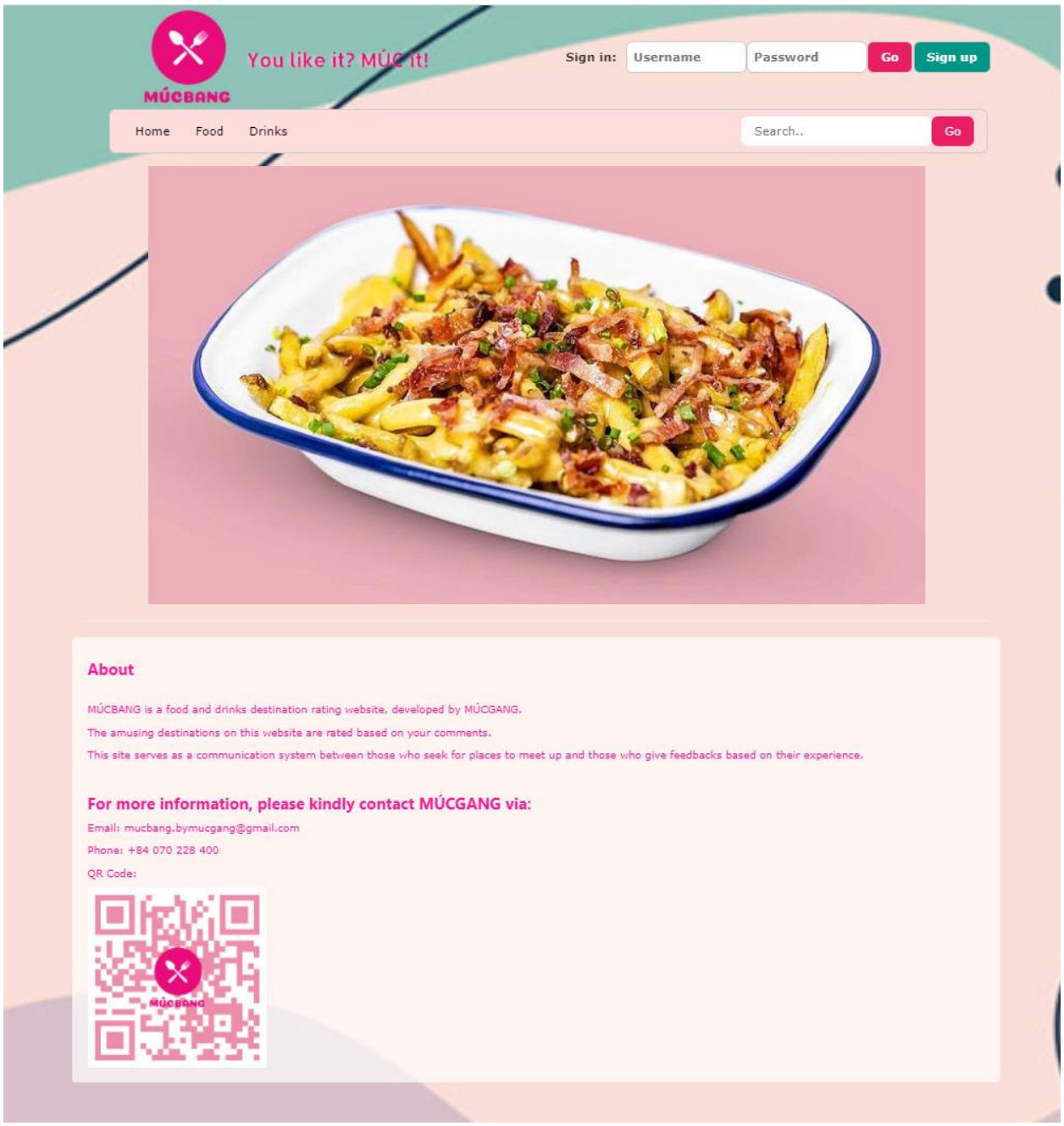


Figure 18: Home page

This is the first user interface users will see when they come to the website. Unless the user has had the account, they need to **sign up**. Otherwise, they just need to sign in by their username and password they have registered before.

b. Sign up

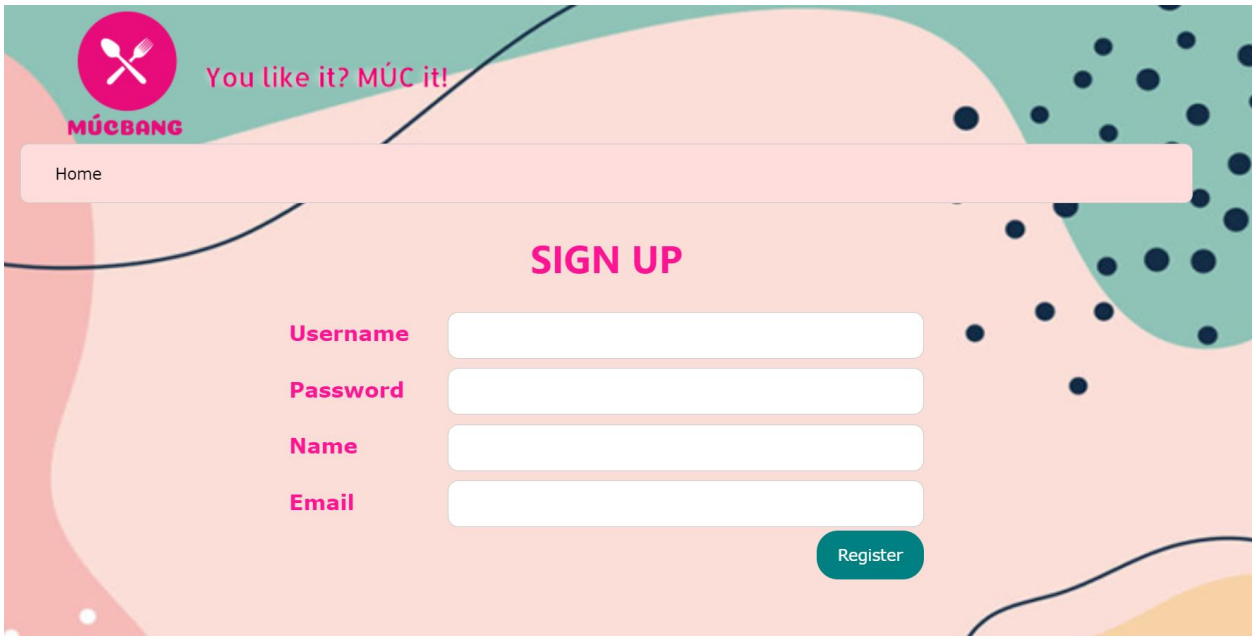


Figure 19: Sign up form

The users need to provide informations including username, password, name and email. The system will automatically insert the information to *account* table

```php
include_once "lib/config.php";
include_once "lib/DataProvider.php";
global $db_host, $db_username, $db_password, $db_name;

$connection = new mysqli($db_host, $db_username, $db_password, $db_name);
/* check connection */
if ($connection->connect_error) {
    die("Failed to connect: " . $connection->connect_error);
}

if(isset($_POST["username"]) && isset($_POST["password"]) && isset($_POST["name"]) && isset($_POST["email"])){
        $username = $_POST["username"];
        $password = $_POST["password"];
        $name = $_POST["name"];
        $email = $_POST["email"];

        $sql = "insert into account (username, password, name, email)
        values ('$username','$password','$name','$email')";
        if($connection->query($sql) == true)
        {
        }
        DataProvider::ChangeURL("index.php");
    }
```

Figure 20: Source code for sign up

c. Login

The system requires user to input his/her username and password.

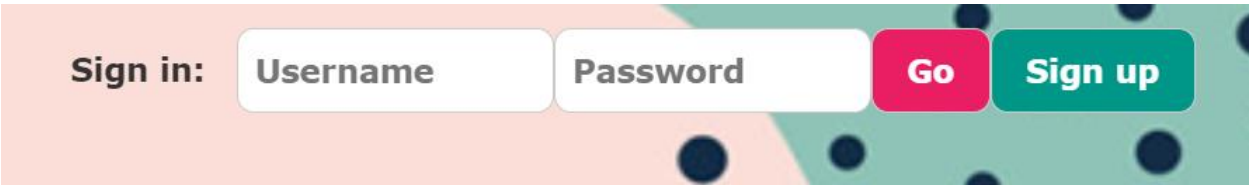


Figure 21: Login form

If the user input valid and correct information, it will come up like the below picture:

Figure 22: Succeed login

```
<div id="accountId">
    <?php
        echo "Hello, ".$_SESSION["username"]." | ";
    ?>
    <a href="index.php?a=11">
        <button>Logout</button>
    </a>
</div>
```

Figure 23: Source code for succed login

If the user does not input one of two, username or password and input different information from the database, the system will pop up the alert message:

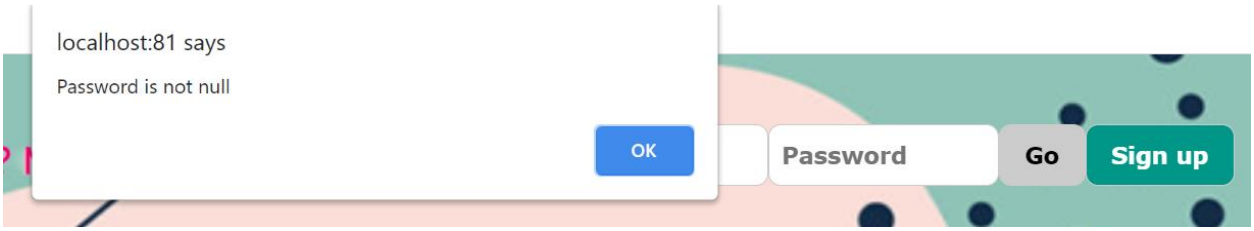


Figure 24: Alert message for not null login

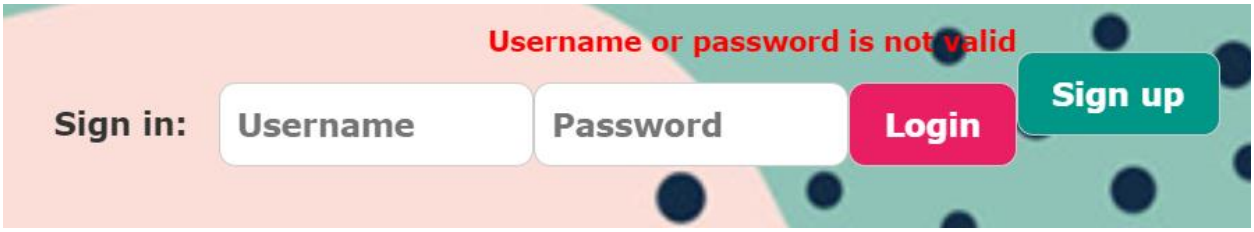


Figure 25: Alert message for invalid input

```php
<?php
        include_once "lib/config.php";
        include_once "lib/DataProvider.php";
        global $db_host, $db_username, $db_password, $db_name;

        $connection = new mysqli($db_host, $db_username, $db_password, $db_name);
        /* check connection */
        if ($connection->connect_error) {
            die("Failed to connect: " . $connection->connect_error);
        }

        $error = '';
        if(isset($_POST["username"]) && isset($_POST["password"])){
            $username = $_POST["username"];
            $password = $_POST["password"];


            $sql = "select username, name
            from account
            where username = '$username' and password = '$password' ";

            $result = DataProvider::ExecuteQuery($sql);
            $row = mysqli_fetch_array($result);

            if($row != null)
            {
                $_SESSION["username"] = $row["username"];
                $_SESSION["name"] = $row["name"];
                DataProvider::ChangeURL("./Food.php?user=".$row["username"]."");
            }
            else
            {
                $error .= '<p class = "error" style = "color: red" >Username or password is not valid</p>';
            }
        }
    ?>
```

Figure 26: Source code for checking login

When the user chooses Logout, it will return back to home page

d. View of Food and Drinks pages

When option “Food” on the header bar is chosen, the website changes its path to *http://localhost/SentimentalAnalysisForProductRating/Food.php* as shown above. The page displays all of the food destinations added by admin. Each of the destination is in a box declaring if it belongs to Restaurants, Buffet or Street Food category. All of the destination’s information is shown in the box.

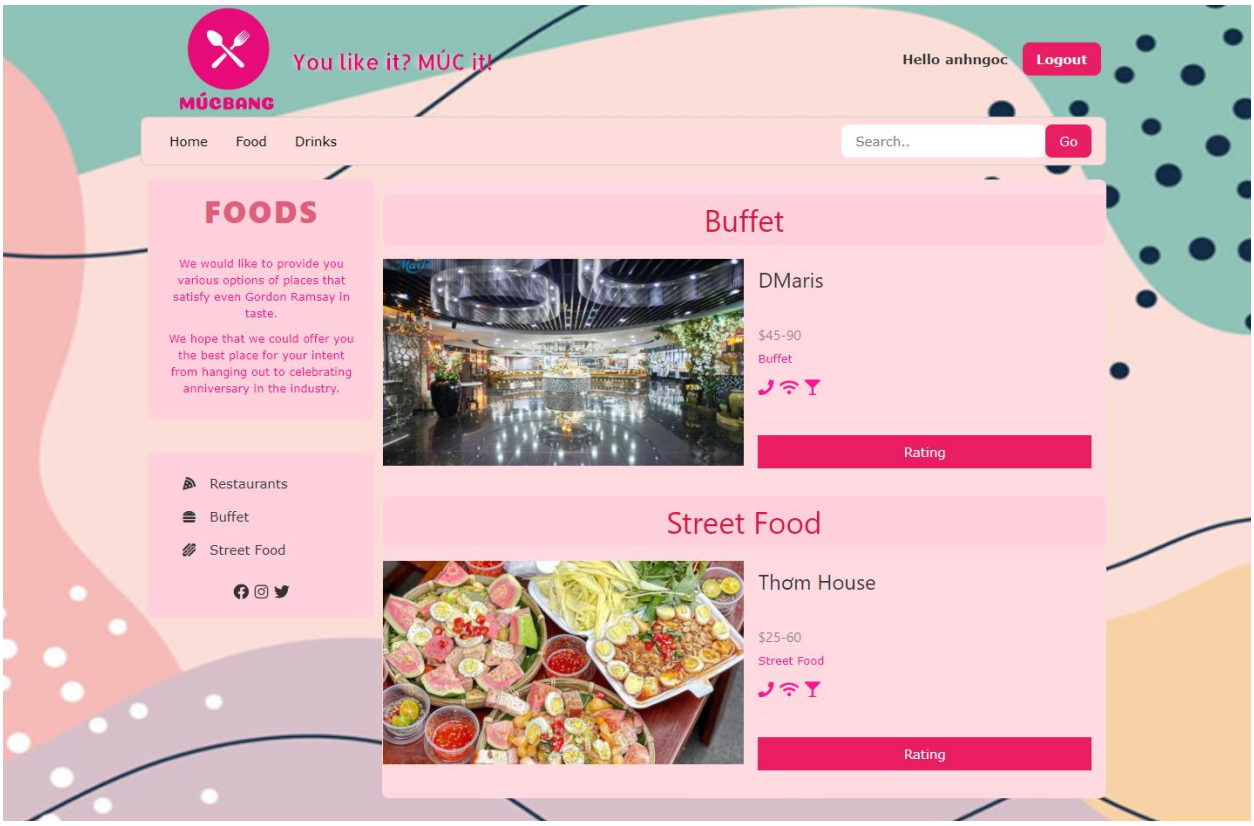


Figure 27: Food page

```
--Restaurant-->
iv class="w3-col m9 w3-container w3-row-padding w3-margin-top w3-round-large" style ="background-color: #ffdbe1">
  <?php
    //Take data from database and show on the web
    //Restaurant
    $sqlRestaurant = "SELECT foodstalltype.foodStallType,
    food.postID,
    food.rating,
    food.address,
    food.image,
    food.workingTime,
    food.priceRange,
    food.phoneNumber,
    food.foodName,
    account.username
    FROM food, foodstalltype, account
    WHERE foodStallType = 'Restaurant'
    AND food.postID = foodstalltype.postID";
    $resultRestaurant = mysqli_query($connection, $sqlRestaurant);
    if ($resultRestaurant){
        if(mysqli_num_rows($resultRestaurant) > 0){
            while($row = mysqli_fetch_array($resultRestaurant)){
                echo"
                <!--Title-->
                <div class='row w3-round-large w3-margin-top w3-margin-bottom' id='restaurant' style ='background-color: #ffd1dc'>
                        <h3 class ='title'>".$row['foodStallType']."</h3>
                    </div>

                    <!--Image-->
                    <div class='row w3-margin-bottom w3-margin-top'>
                        <div class='w3-half'>
                            <div class='w3-half w3-display-container w3-hover-opacity' style='transition:0.5s;width:100%'>
                                <img src=".$row['image']." alt='food' style='width:100%'>
                                <div class='w3-display-topleft w3-display-hover w3-large'>
                                    <button type='button' class='w3-animate-opacity w3-btn w3-margin w3-round' style='left:65px; background-color: #ffdbe1' title='Like'>
                                        <i class='fa fa-heart w3-text-pink'></i>
                                    </button>
                                </div>

                                <div class='w3-display-topleft w3-display-hover w3-large' style='left:65px'>
                                    <button type='button' class='w3-animate-opacity w3-btn w3-margin w3-round' style='left:65px; background-color: #ffdbe1' title='Share'>
                                        <i class='fa fa-paper-plane w3-text-pink'></i>
                                    </button>
                                </div>

                                <div class='w3-display-bottomleft w3-display-hover w3-large w3-text-white'>
                                    <div class='w3-padding w3-text-pink w3-animate-opacity'>Rating point: ".$row['rating']."</div>
                                </div>

                                <div class='w3-display-middle w3-display-hover w3-large'>
                                    <a href='./Rating.php?id=".$row['postID']."&user=".$row['username']."'><button type='submit' name = 'submit'
                                    class='w3-animate-opacity w3-btn w3-round w3-text-pink' style='background-color: #ffdbe1'>Show Rating</button></a>
                                </div>
                            </div>
```

Figure 28: Source code for displaying destinations which belong to Restaurant category (1)

```
                    <!--Information Description-->
                        <div class='w3-half w3-margin-bottom' style ='background-color: #ffdbe1'>
                            <div class='w3-container'>
                                <h3>".$row['foodName']."</h3>
                                <br>
                                <h6 class='w3-opacity'>$".$row['priceRange']."</h6>
                                <p>".$row['foodStallType']."</p>
                                <p class='w3-large'></i>
                                    <i class='fa fa-phone'></i>
                                    <i class='fa fa-wifi'></i>
                                    <i class='fa fa-glass'></i>
                                    <i class='fa fa-cutlery'></i>
                                </p>
                                <hr>
                                <a href='./Rating.php?id=".$row['postID']."&user=".$row['username']."'><button type='submit' name = 'submit'
                                class='w3-button w3-block w3-pink'>Rating</button></a>
                            </div>
                        </div>
                    </div>
```

Figure 29: Source code for displaying destinations which belong to Restaurant category (2)

```
<a href='./Rating.php?id=".$row['postID']."&user=".$row['username']."'><button type='submit'
name = 'submit' class='w3-button w3-block w3-pink'>Rating</button></a>
```

Figure 30: Source code for identifying postID and username

The source codes for displaying destinations which belong to Buffet and Street Food categories are similarly written comparing to the codes in the two given figures above.

The Drinks page (http://localhost/SentimentalAnalysisForProductRating/Drinks.php) and the Food page looks identically alike since they have similar source codes.

e. Rating

When user push the button Rating of the previous page, they can view all the information in detail. At this page, they can upload and leave their comments below the information, the system will automatically score the rating point based on their comments.

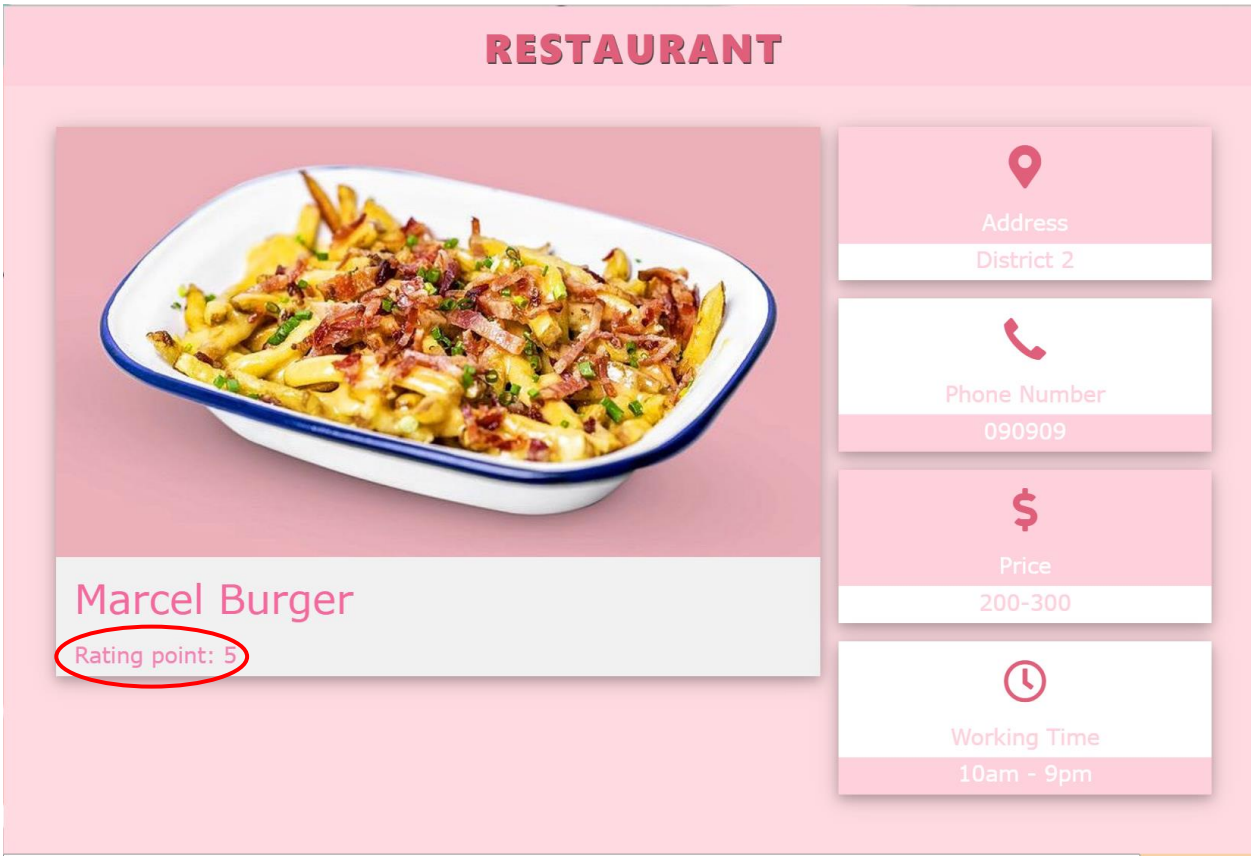


Figure 31: The information is shown in detail

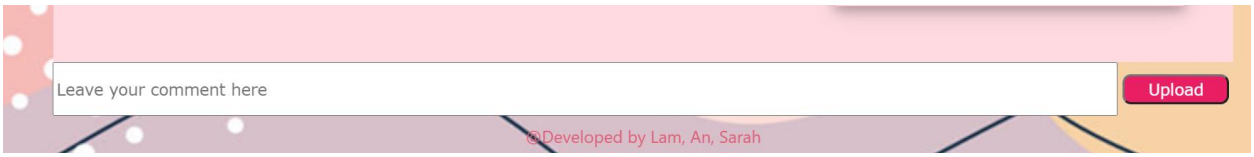


Figure 32: Place to leave the comment

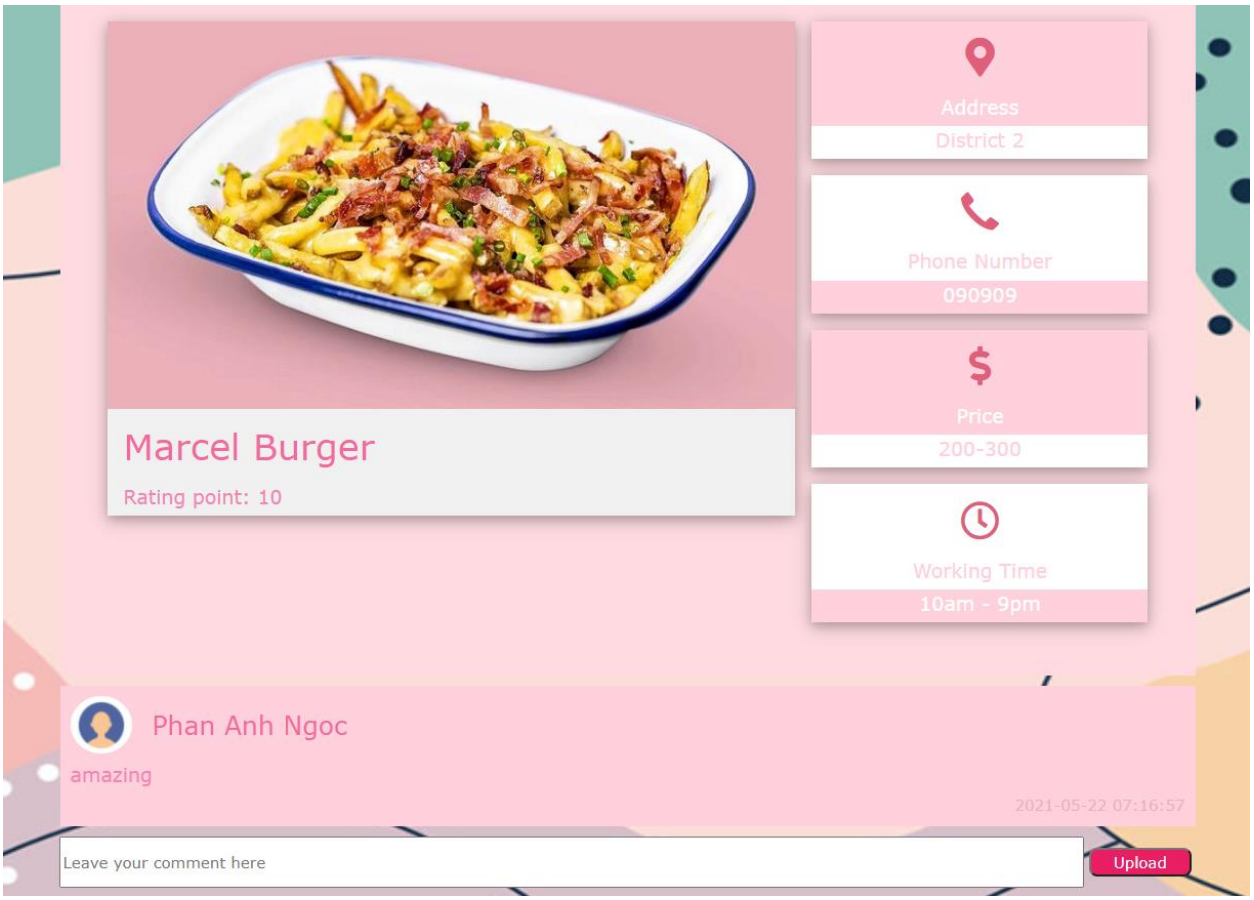


Figure 33: The comment is shown after posting

```php
//add comment and word into database
if(isset($_POST["txtComment"])){
    $content = $_POST["txtComment"];
    $validWord = (string) NULL;
    $weightTotal = 0;
    $index = 0;

    $sqlComment = "INSERT into comment SET
    commentID = NULL,
    username = '$username',
    content = '$content',
    word = NULL,
    point = NULL,
    time = DEFAULT";

    $addComment = mysqli_query($connection, $sqlComment) or die (mysqli_connect_errno()."Cannot insert comment");;
    $sqlSententalWordFromSentimentalWord = "SELECT word FROM sentimentalword";
    $resultSententalWordFromSentimentalWord = mysqli_query($connection, $sqlSententalWordFromSentimentalWord);

    while($arraySentimetalWord = mysqli_fetch_array($resultSententalWordFromSentimentalWord)){
        $sentimentalWord = $arraySentimetalWord["word"];
        $HASsentimentalWordFromComment = strpos($content, $sentimentalWord);

        if ($HASsentimentalWordFromComment !== false){
            $validWord = append_string($validWord, $sentimentalWord);
            $sqlComment = "UPDATE comment
            SET word = '$validWord'
            WHERE content = '$content'";
            $addComment = mysqli_query($connection, $sqlComment) or die (mysqli_connect_errno()."Cannot insert word");;

            $weightOfWord = mysqli_query($connection, "select weight from sentimentalword where word = '$sentimentalWord'");

            while ($array_weightOfEachWord = mysqli_fetch_array($weightOfWord)){
                $string_weightOfEachWord = $array_weightOfEachWord['weight'];
                $float_weightOfEachWord = floatval($string_weightOfEachWord);
                $weightTotal = floatval($weightTotal + $float_weightOfEachWord);
            }
            $index = $index + 1;
        }
    }
    $weightComment = $weightTotal/$index;
```

Figure 34: Source code add comment from users to database and calculate weight of words (1)

```
$UpdatePoint = mysqli_query($connection,
"UPDATE comment
SET point = '$weightComment'
WHERE content = '$content'");


$commentID = mysqli_query($connection, "select max(commentID) from comment");
$getCommentID = mysqli_fetch_array($commentID);
$COMMENTID = $getCommentID[0];

$addPreferences = mysqli_query($connection, "insert into preferences(postID, username, commentID) values ('$ID', '$USER', '$COMMENTID')");
$countComment = mysqli_query($connection, "SELECT COUNT(comment.commentID) FROM comment, preferences, post
WHERE comment.commentID = preferences.commentID AND post.postID = preferences.postID AND post.postID = '$ID'");

while ($array_countComment = mysqli_fetch_array($countComment)){
    $string_counComment = $array_countComment[0];
    $float_countComment = intval($string_counComment);

    if ($float_countComment == 0){
        $TotalWeight = 0;
    }
    else{
        $totalRating = mysqli_query($connection, "SELECT AVG(comment.point) FROM comment, preferences, post
        WHERE comment.commentID = preferences.commentID AND post.postID = preferences.postID AND post.postID = '$ID'");
        while ($array_totalRating = mysqli_fetch_array($totalRating)){
            $string_totalRating = $array_totalRating[0];
            $float_totalRating = floatval($string_totalRating);
        }
    }
}
$postRating = mysqli_query($connection, "update post set rating = '$float_totalRating' where postID = $ID");
$foodRating = mysqli_query($connection, "update food set rating = '$float_totalRating' where postID = $ID");
$drinkRating = mysqli_query($connection, "update drink set rating = '$float_totalRating' where postID = $ID");
```

Figure 35: Source code add comment from users to database and calculate weight of words (2)

```
<!-- Show comment -->
<div class='container' >
    <div class = 'w3-container'>
        <?php

            $sqlComment = "SELECT comment.content, comment.time, account.name
            FROM post, preferences, comment, account
            WHERE preferences.postID = post.postID
            AND post.postID = '$ID'
            AND preferences.username = account.username
            AND preferences.commentID = comment.commentID";

            $resultComment = mysqli_query($connection, $sqlComment) or die(mysqli_connect_errno()."wooooooo");;
            if ($resultComment){
                if(mysqli_num_rows($resultComment) > 0){
                    while ($showComment = mysqli_fetch_array($resultComment)){
                        echo"
                            <div class='container_chat'>
                                <img src='img/user1.jpg' alt='Avatar' style='width:100%;'>
                                    <p class = 'user'>".$showComment['name']."</p>
                                    <p class = 'w3-margin-top'>".$showComment['content']."</p>
                                <span class='time-right'>".$showComment['time']."</span>
                            </div>
                        ";
                    }
                }
            }
        ?>
    </div>
```

Figure 36: Reveal the comment

4.2.2. Admin pages

a. Admin Home Page

Admin pages will have the same function as User pages like Home page, view list of food, and drink, view the comments and rating. However, admin user can able to add for places for food and drink.

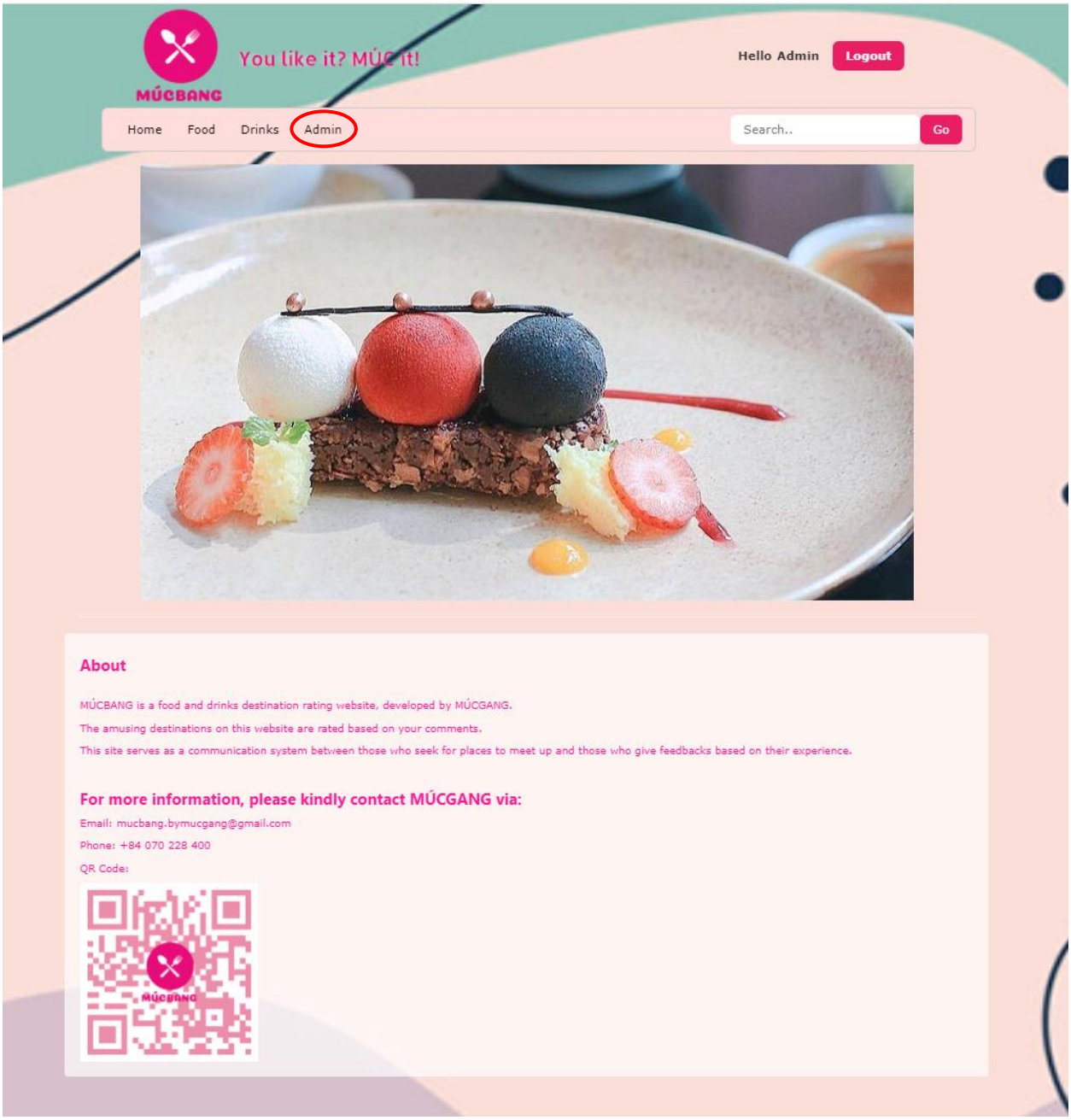


Figure 37: Admin section for Admin user

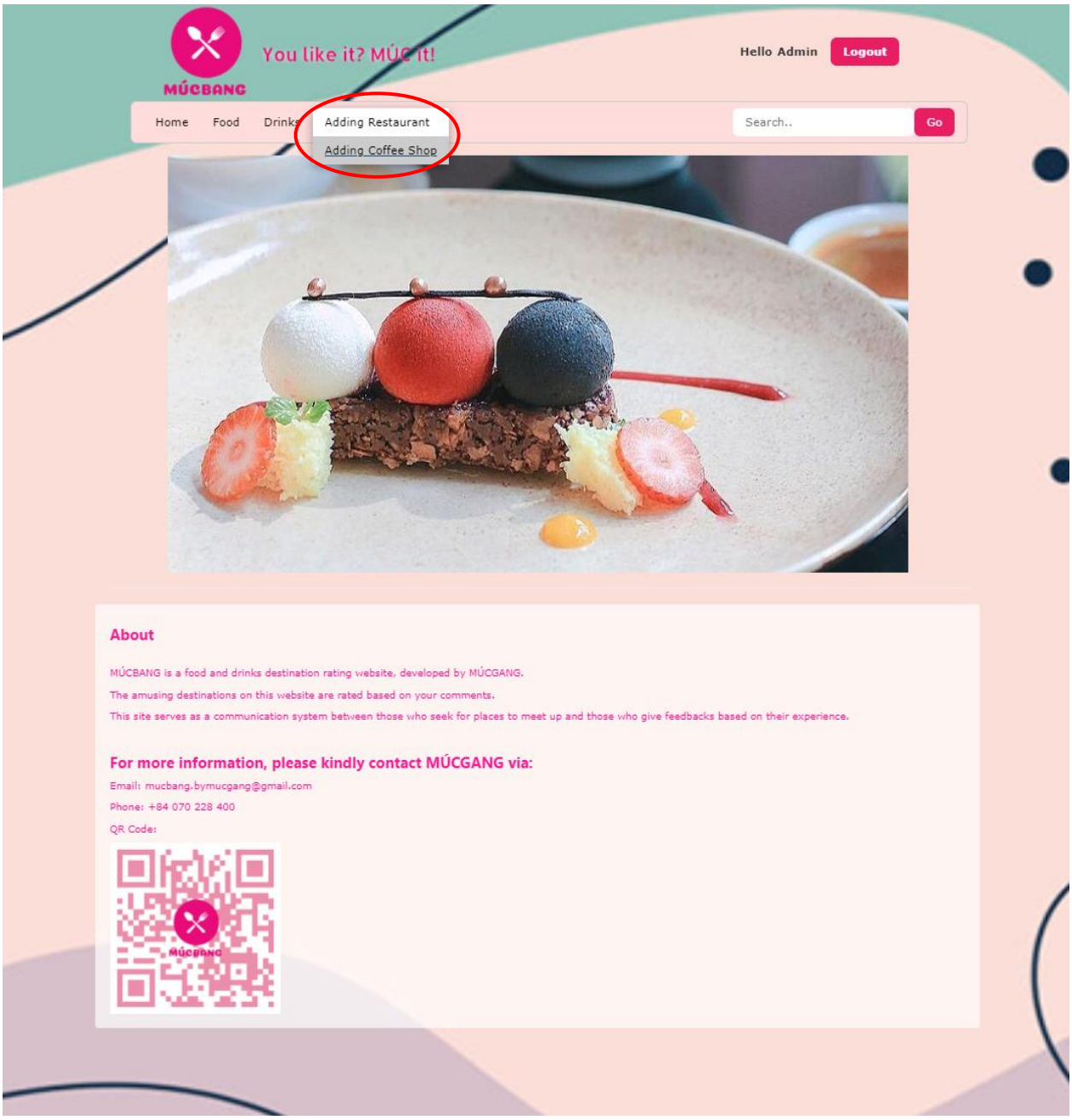


Figure 38: Adding fuction for admin

b. Adding food

Admin need to give information about the post like name, image, address, working time, price range, phone number and types of food. For the types of food, admin will need to choose one of three: restaurant, street food, and buffet. After adding, the added place will

be shown in food section like in user pages so admin do not have to change to user pages for checking

The information (except for types of food) will be added to post and food table. For types of food, they will be added to foodstalltype table.

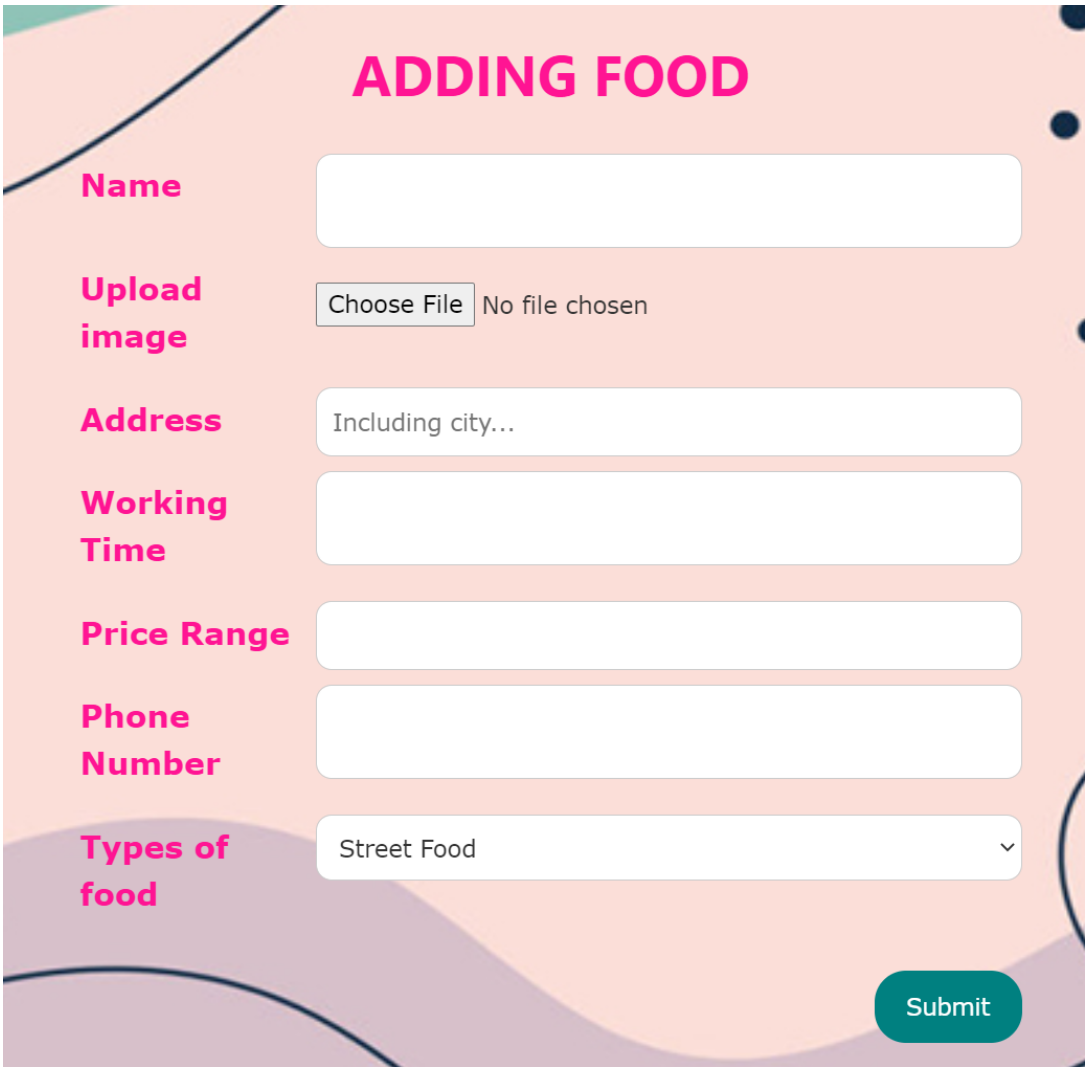


Figure 39: Adding restaurant form

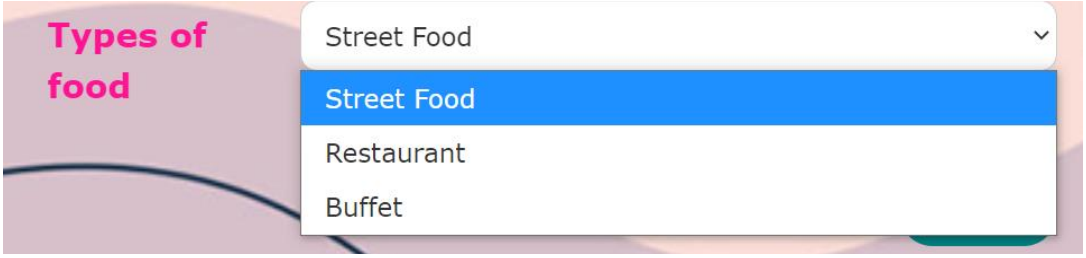


Figure 40: Types of food

```php
<?php

    include_once "lib/config.php";
    include_once "lib/DataProvider.php";
    global $db_host, $db_username, $db_password, $db_name;

    $connection = new mysqli($db_host, $db_username, $db_password, $db_name);
    /* check connection */
    if ($connection->connect_error) {
        die("Failed to connect: " . $connection->connect_error);
     }

    if(isset($_POST["name"]) && isset($_POST["image"]) && isset($_POST["address"]) && isset($_POST["workingtime"]) && isset($_POST["pricerange"])
      && isset($_POST["phonenumber"]) && isset($_POST["drinkstypes"])){
            $name = $_POST["name"];
            $image = $_POST["image"];
            $address = $_POST["address"];
            $workingtime = $_POST["workingtime"];
            $pricerange = $_POST["pricerange"];
            $phonenumber = $_POST["phonenumber"];
            $drinkstypes = $_POST["drinkstypes"];

            $sql = "insert into post (name, address, image, workingTime, priceRange, phoneNumber)
            values ('$name', '$address', 'img/$image', '$workingtime', '$pricerange', '$phonenumber')";

            if($connection->query($sql) == true)
            {
            }

            $sql_1 = "insert into drink (postID, address, image, drinkName, phoneNumber, priceRange, workingTime)
            select post.postID, post.address, post.image, post.name, post.phoneNumber, post.priceRange, post.workingTime
            from post
            where post.postID = (select max(p1.postID) from post as p1)";
            $sql_2 = "insert into drinkstalltype (postID, drinkStallType)
            values ((select max(postID) from post), '$drinkstypes')";

            if(($connection->query($sql_1) == true) && ($connection->query($sql_2) == true))
            {
            }

            DataProvider::ChangeURL("Admindrink.php");
        }
?>
```

Figure 41: Source code for adding food

c. Adding drink

Adding post of drink just the same as adding food function. The only difference is drink they have another three different types of drink.

After adding, the information and the types of drink will be added to post and drink table and drinkstalltype table, respectively.

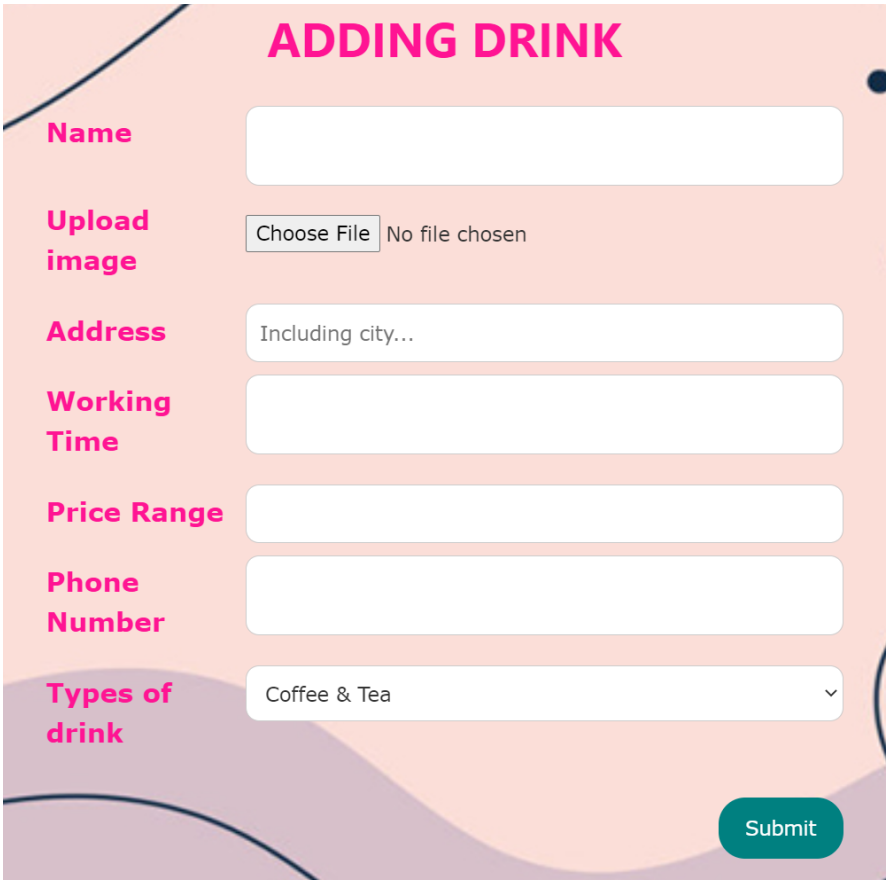


Figure 42: Adding drink form

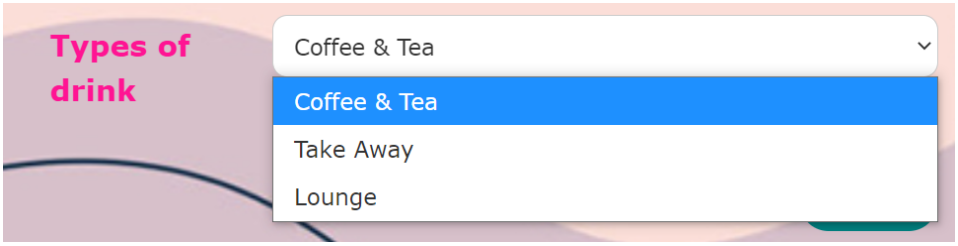


Figure 43: Types of drink

```php
<?php
    include_once "lib/config.php";
    include_once "lib/DataProvider.php";
    global $db_host, $db_username, $db_password, $db_name;

    $connection = new mysqli($db_host, $db_username, $db_password, $db_name);
    /* check connection */
    if ($connection->connect_error) {
        die("Failed to connect: " . $connection->connect_error);
    }

    if(isset($_POST["name"]) && isset($_POST["image"]) && isset($_POST["address"]) && isset($_POST["workingtime"]) && isset($_POST["pricerange"])
    && isset($_POST["phonenumber"]) && isset($_POST["drinkstypes"])){
            $name = $_POST["name"];
            $image = $_POST["image"];
            $address = $_POST["address"];
            $workingtime = $_POST["workingtime"];
            $pricerange = $_POST["pricerange"];
            $phonenumber = $_POST["phonenumber"];
            $drinkstypes = $_POST["drinkstypes"];

            $sql = "insert into post (name, address, image, workingTime, priceRange, phoneNumber)
            values ('$name', '$address', 'img/$image', '$workingtime', '$pricerange', '$phonenumber')";

            if($connection->query($sql) == true)
            {
            }

            $sql_1 = "insert into drink (postID, address, image, drinkName, phoneNumber, priceRange, workingTime)
            select post.postID, post.address, post.image, post.name, post.phoneNumber, post.priceRange, post.workingTime
            from post
            where post.postID = (select max(p1.postID) from post as p1)";
            $sql_2 = "insert into drinkstalltype (postID, drinkStallType)
            values ((select max(postID) from post), '$drinkstypes')";

            if(($connection->query($sql_1) == true) && ($connection->query($sql_2) == true))
            {
            }

            DataProvider::ChangeURL("Admindrink.php");
        }
?>
```

Figure 44: Source code for adding drink

4.2.3. Query

This page will provide users a tool to access to database system so that they can keep track their information at a glance

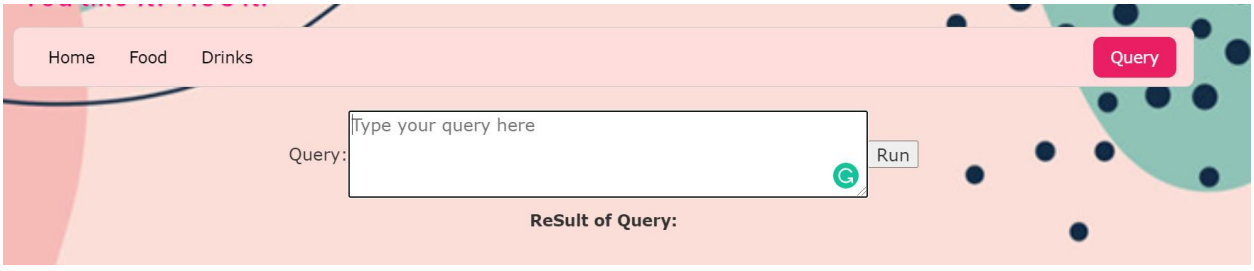
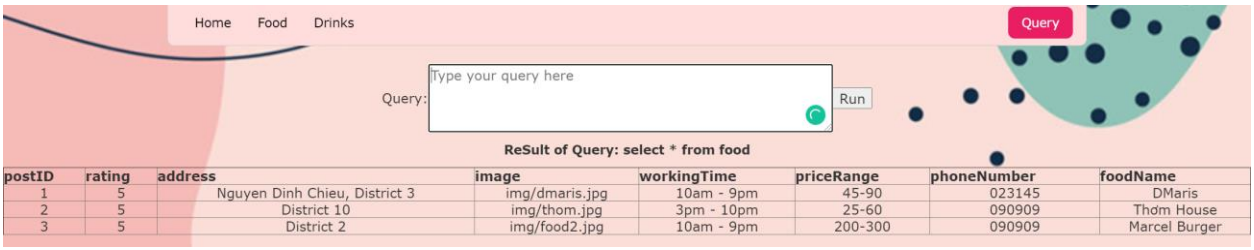


Figure 45: Query form



postID	rating	address	image	workingTime	priceRange	phoneNumber	foodName
1	5	Nguyen Dinh Chieu, District 3	img/dmaris.jpg	10am - 9pm	45-90	023145	DMaris
2	5	District 10	img/thom.jpg	3pm - 10pm	25-60	090909	Thơm House
3	5	District 2	img/food2.jpg	10am - 9pm	200-300	090909	Marcel Burger

Figure 46: Reveal the information from database

V. Conclusion

The project objective stated in the beginning of this report has been satisfied as this website of Food and Drinks Rating was implemented systematically successful yet visually captivating. Throughout stages from coming up with ideas to realizing them, the team members have gained a lot of practical knowledge such as managing database system using SQL servers and designing web pages using HTML, CSS and PHP. How different development phases of a project stand in line and the life cycle of development itself were the lessons we unexpectedly learned from this project assignment.

There are many more fascinating features that can be implemented in this system, such as enabling users to “save” destinations that draw their attention, or a financial-supporting feature, let’s say, enabling stakeholders to hang up their advertisement banners on the website. The project holds so much potential that we wish to have a chance to further develop it in the future.