Phoenix

JELP: Restaurant Recommendation and Analysis Website

Seoyoung Jo†, ***1976371 and*** **Jiyoon Park**††, ***1876166***

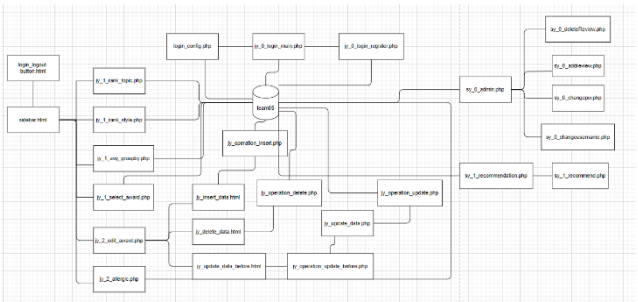
key words: *Food Data. Restaurant Analysis*

1. Project Description

1.1 Project Overview

Our Project JELP is a restaurant recommendation service that also provides useful information on restaurants. Our website cannot be accessed unless a user logs into the website. Once logged in, the user has access to all rank viewing, selecting, and viewing wanted data and data manipulation!

1.2 Project Structure



1.3 Project File Description

jy_0_login_main.php: This is the login page users will see as they enter our service. Only when a user is logged in, can they see the website. This page initializes the sessions.

jy_0_login_register.php: If the user does not already have an account, then they go to this page to create their accounts.

jy_1_rank_topic.php: This is the first page people see when logging into the website. This page let's the user select by which column they want the rank to be ordered. They have a choice between Average price per person, average rating per person, total amount of reviews per restaurants.

jy_1_rank_style.php: This page allows the users to select and see only restaurants that serve particular style of menu. They have a choice between Korean, Japanese, Chinese, Western and Dessert.

jy_1_select_award.php: This page allows users to see awards given by each organization. Users can choose among four organizations.

jy_1_avg_groupby.php: This page shows analytical data of restaurants. This is the only page in our website that does not take in user input. The page shows average price and rating per topic. Such as the average price for Korean food or average price of food in Texas.

jy_2_edit_award.php: This page allows the users to manipulate award data. The user can choose between Insert new data, update current data, delete current data. Each function relies on other scripts that will be further explained later in the paper.

jy_2_allergic.php: This is a special page made for people with egg or peanut allergies. When the user selects an ingredient, they are allergic to, then the page shows restaurants that are safe from the food.

sy_0_addview.php: User can add reviews. The items to be written are restaurant name, location, food classification (Korean food, Chinese food, etc.), average price for two people, rating, and review in writing. These items are related to the Review table in Database. INSERT, SELECT statements were used.

sy_0_admin.php: User can see the number of reviews she has written, the average star rating of her own review, and the list of her reviews. These items are updated and displayed on the screen whenever the user adds a review. SELECT, WINDOW, and INNER JOIN were used.

sy_0_changepw.php, sy_0_changeusername.php: User can change username and password. If user change username, the username display in the upper right corner will also change. UPDATE statement was used.

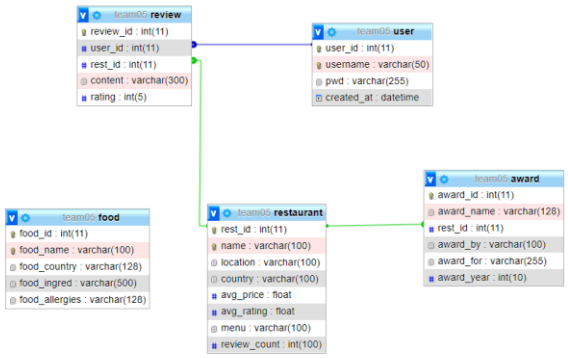
sy_1_recommend.php, sy_1_recommend.php: Among Korean, Chinese, Japanese, Western, and Desserts, this page shows the category with the highest average rating and the lowest price. When the user enters the category she wants

† The authors contributed to the project equally

and price range, the page shows which category of food should be selected in consideration of it. The input value is applied as a filter to show restaurants that match the filter value in the order of rating. SELECT, GROUP BY, RANK, ORDER BY, etc. were used.

2. Database Schema & ER diagram

2.1 ER Diagram



2.2 Table Description

Restaurant Table (315 entries): Saves the restaurant id(rest_id), the name of the restaurant(name), the location(location), average price per person(avg_price), average review per restaurant(avg_rating) and the total number of reviews per restaurant(review_count).

Award Table (28 entries): Saves the award's unique id (award_id), the id of restaurant that received the award(rest_id), name of organization that gave the award (award_by), reason for receiving the award (award_for) and the year it was awarded in (award_year).

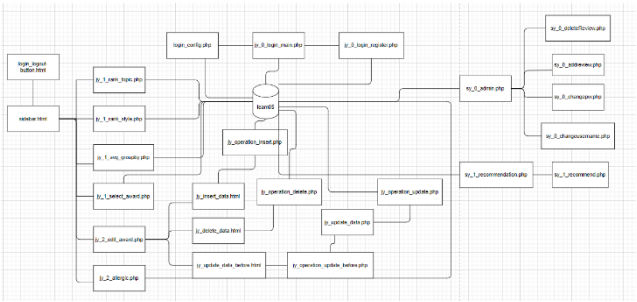
Food Table (23 entries): Saves the auto generated food id(food_id), the style of food(food_counrty), the food's main ingredients(food_ingred) and the ingredients that may cause food allergies (food_allergies).

User Table (150 entries): Saves users' login information. The colomns are user's id(user_id, PRIMARY KEY), username(input by user), password(pwd), and the time record was created(created_at).

Review Table (276 entries): Saves review data. The columns are review_id(PRIMARY KEY), user_id(FOREIGN KEY), rest_id(FOREIGN KEY), contemt(review content), rating

Filter Table (1 entry): Saves filter data, which is user input. The columns are filter_id(PRIMARY KEY), category(one among Korean, Chinese, Japanese, Western, Dessert), price range(lowerPrice, upperPrice)

3. Overview of PHP Code Structure



3.1 jy_0_login_main.php: Uses sessions and select to get user data

```
session_start();

if (isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] ==== true) {
    header("location: jy_1_rank_topic.php?orderby=avg_price");
    exit;
}
$sql = "SELECT user_id, username, pwd FROM user WHERE username = '" . $username . "'";
```

3.2 jy_0_login_register.php: Uses INSERT and SELECT to add new users to the database

```
$sql = "SELECT user_id FROM
user WHERE username = ?";
$sql = "INSERT INTO user (username, pwd) VALUES (?, ?)";
```

3.3 jy_1_rank_topic.php: Uses RANK to rank data according to user input

```
$sql = "
SELECT RANK() OVER (ORDER BY " . $orderby." DESC) as
ranking, name, location, country, avg_price, avg_rating, review_count
  FROM restaurant";
```

3.4 jy_1_rank_style.php: Uses SELECT WHERE ORDER BY to select according to user input.

```
$sql = "
      SELECT *
      FROM restaurant
      WHERE country = ?
      ORDER BY avg_rating DESC, avg_price;";
```

3.5 jy_1_select_award.php: Uses SELECT WHERE to get data according to what the customers selected. It also uses INNER JOIN to get data from not only the award database but also

```
$sql = "
   SELECT *
   FROM award
   INNER JOIN restaurant
   ON award.rest_id = restaurant.rest_id
   WHERE award_by = ?
   ORDER BY award_name;";
```

3.6 jy_1_avg_groupby.php: Uses GROUP BY, AVG, SUM to provide useful information to users for them to analyze restaurants.

```
$sql = "
SELECT country, AVG(avg_price) as avg_price, AVG(avg_rating) as avg_rating, SUM(review_count) as review_count
FROM restaurant
GROUP BY country
ORDER BY country;";
```

3.7 jy_2_edit_award.php: There are many functions that take action in this file.

First is INSERT.

```
$sql = "INSERT INTO award (award_name, rest_id, award_by, award_for, award_year)
VALUES ('" . $_POST['award_name'] . "'," . $_POST['rest_id'] . ",'" . $_POST['award_by'] . "','" . $_POST['award_for'] . "'," . $_POST['award_year'] . ")";
```

Second is UPDATE.

```
$sql = "UPDATE award SET award_name='" . $_POST['award_name'] . "', award_by = '" . $_POST['award_by'] . "', award_for = '" . $_POST['award_for'] . "',award_year = " . $_POST['award_year'] . " WHERE award_id=" . $_POST['award_id'] . ";";
```

Third is DELETE.

```
$sql = "DELETE FROM award WHERE award_id =".$_POST["award_id"];
```

3.8 jy_2_allergic.php: Uses SELECT WEHRE to get data from database.

```
$sql = "
    SELECT *
    FROM food
    INNER JOIN restaurant
    ON food.food_name = restaurant.menu
    WHERE food.food_allergies ='none';";
```

3.8 sy_0_admin.php: User can see the count of her reviews, average rating of her reviews and can add or delete reviews. Also, user can change her username and password. SELECT, AVG, WINDOW, INNER JOIN, SUB QUERY were used.

```
SELECT review.review_id, restaurant.rest_id, restaurant.name, review.content, review.rating, AVG(review.rating) OVER (PARTITION BY review.user_id) as rating_avg FROM review INNER JOIN restaurant ON review.rest_id = restaurant.rest_id WHERE review.user_id=(SELECT user_id FROM user WHERE user.username = '$username');
```

3.9 sy_0_addreview.php: This is backend code of sy_0_admin.php. User can add review. INSERT and TRANSACTION were used.

```
try {
    $mysqli->autocommit(FALSE);
    $mysqli->begin_transaction();
    $mysqli->query("INSERT INTO restaurant(name, location, country, avg_price) values('$name','$location','$country',$price);");
    //$mysqli->query("SELECT rest_id from restaurant WHERE name='$name';");
    $mysqli->query("INSERT INTO review (user_id, rest_id, content, rating) VALUES ((SELECT user_id FROM user WHERE username = '$username'), (SELECT rest_id from restaurant WHERE name='$name'), '$content', $rating);");

    $mysqli->commit();
    $mysqli->autocommit(TRUE);
    $mysqli->close();
    header('Location: sy_0_admin.php');
} catch (mysqli_sql_exception $exception) {
    $mysqli->rollback();
    $mysqli->autocommit(TRUE);
    echo $exception;
    echo 'rollback';
}
```

3.10 sy_0_deleteReview.php: This is backend code of sy_0_admin.php. User can delete review. DELETE and WHERE were used.

```
$sql = "DELETE FROM review WHERE review_id = {$filtered['id']}";
```

3.11 sy_0_changepw.php: This is backend code of sy_0_admin.php. User can change the password. UPDATE and WHERE were used.

```
$sql = "UPDATE user SET pwd='{$_POST['changepw']}' WHERE username='$username' ";
```

3.12 sy_0_changeusername.php: This is backend code of sy_0_admin.php. User can change the username. UPDATE and WHERE were used.

```
$sql = "UPDATE user SET username='{$_POST['changeusername']}' WHERE username='$username' ";
```

3.13 sy_1_recommendation.php: User can see the recommended category for the user, best rated category, and category with the lowest price. Also, user can choose the filters to see the restaurants ranked by rating screened by the filters. SELECT, RANK, ORDER BY, SUB QUERY, AVG, GROUP BY, COMPLEX GROUPING were used.

```
SELECT *, rank() over (order by avg_rating desc) as rating_rank
FROM restaurant WHERE restaurant.country = (select category from filter order by filter.filter_id desc limit 1) AND avg_price >= (select lowerPrice from filter order by filter.filter_id desc limit 1) AND avg_price <= (select upperPrice from filter order by filter.filter_id desc limit 1);

SELECT country, AVG(avg_rating) as avgrating FROM restaurant GROUP BY country order by avgrating;

SELECT country, avg_price, avg(avg_rating), count(*), avg_rating+count(*) as standard FROM restaurant GROUP BY country, avg_price ORDER BY standard desc;
```

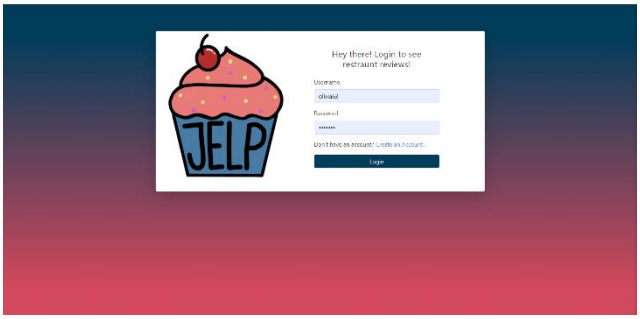
3.14 sy_1_recommend.php: This is backend code of sy_1_recommendation.php. Saves filter data chosen by the user. INSERT was used.

```
INSERT INTO filter(category, lowerPrice, upperPrice)
values('$category', $lowerPrice, $upperPrice);
```

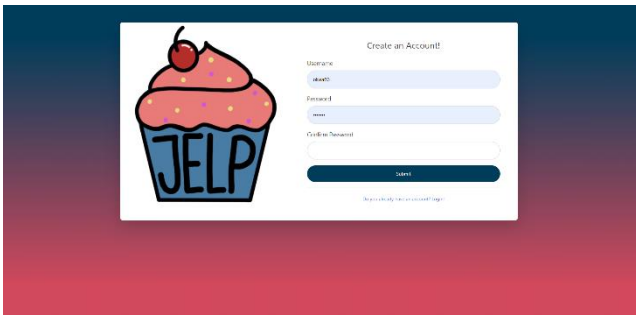
4. Implementation Results

Jiyoon Park

jy_0_login_main.php

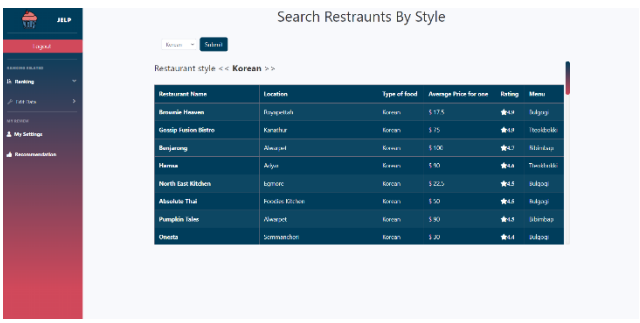


jy_0_login_register.php



jy_1_rank_topic.php

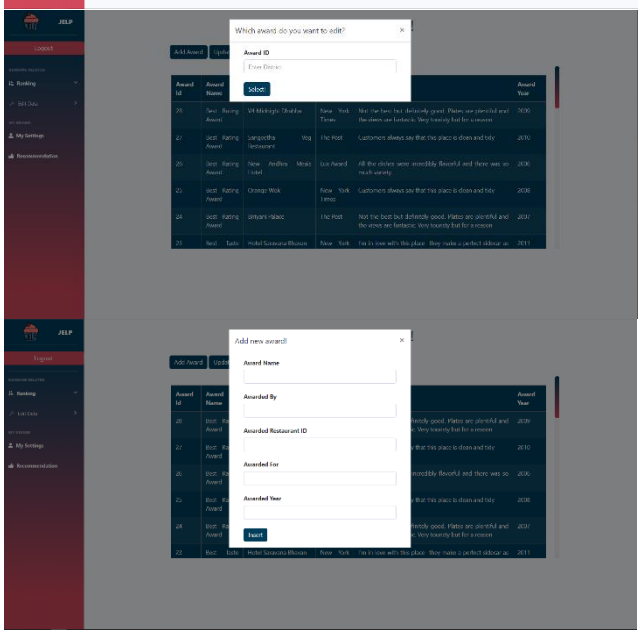
jy_1_rank_style.php

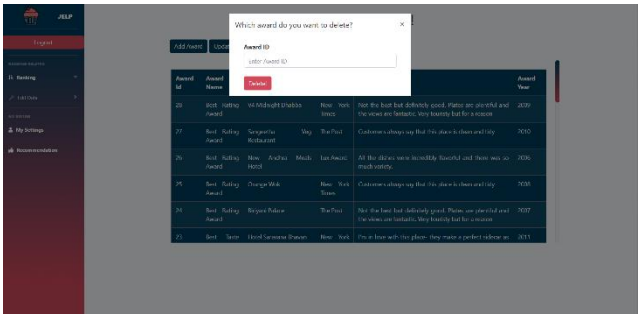


jy_1_select_award.php

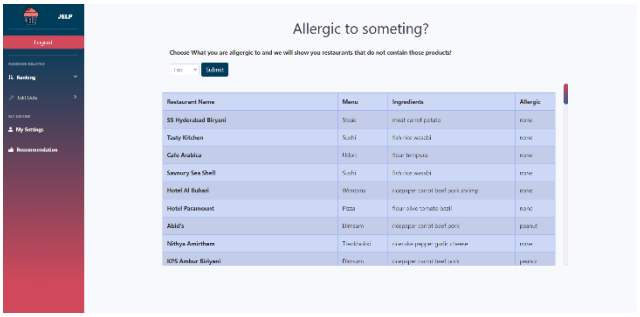
jy_1_avg_groupby.php

jy_2_edit_award.php



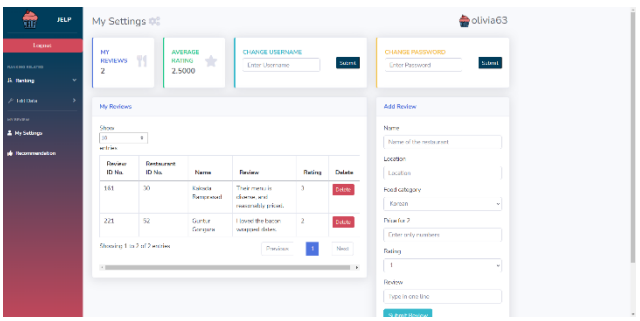


jy_2_allergic.php



Seoyoung Jo

sy_0_admin.php



sy_1_recommendation.php