

Project 2 Report

Abstract

1. Introduction

In project 1, we designed a relational database that can serve as a relational backend for a cooking website that focuses on cooking and recipes as required. In this part, we built a real website with the backend database and front-end frameworks.

Briefly speaking, the website provides the following functions. It allows people to register, create a profile and log in. When logged in, the user can search cooking recipes by keyword, post cooking recipes, review and grade posted cooking recipes, as well as attach additional suggestions to a posted recipe. Also, users can create cooking groups and organize cooking meetings with other users. Users in a specific group can set a meeting or send RSVP to attend a meeting.

To implement these functions, we use PHP with MySQLi to implement our backend services. For the front-end, we use Bootstrap to help with us. We will show how we implement all the functions in the following.

In this report, we will first describe the entire design of the database, including ER diagrams, relational model and constraints, queries, and explain the detail about the design. Then we will explain our backend implementation of PHP. We will use some sample data to justify the website and show how all these things work together.

2. Method

2.1 Programming Language and Tools

Since I am not familiar with web development, I chose PHP 5.5+ (which is the most familiar one) as back-end programming language . I used a very easy front-end framework Bootstrap(based on javascript) to implement front-end. Database is MySQL. Details are here:

Database: MySQL

Back-end: PHP with MySQLi interface

Front-end: Bootstrap , javascript.

The website is host in my local machine. I used Apache HTTP server as the server, and set my working directory to the default Apache directory on Mac (/Library/WebServer/Documents/project2). The browser I used is Google Chrome.

2.2 Database Design

2.2.1 ER Diagram

Here is the assumptions about the website.

1. A user can register a user account with a username as the key, a password and an email. When the user log in to the website, he/she will only use the email as the log in name. This is very common in nowadays website. When the users logs in, they can update their profiles in the accounts. The profile includes first name, last name, address and an introduction about themselves.
2. A user can search different recipes by enter a keyword. They can also post, review, grad and comment recipes. Each recipe contains a description about how to make the food and many ingredients. The recipe can also have some photos with it and can be set with some tags.
3. Users can create cooking groups and held meetings in a group. Any user can create a group without being authorized and set a meeting in that group.
4. Any user in a group can organize meetings and other members in the group can send RSVP to join.

In project1 , we design a table ‘ReviewPhotos’ , ‘RecipePicture’ and ‘MeetingPhotos’ . However , in this part, we deleted these tables and add an attribute ‘picture’ to table ‘Review’ , ’Recipe’ and ‘Meeting’ . This is because multiple photos are not easy to arrange and list clearly in a web page. So we just allow people to upload one picture for an item.

We also deleted the ‘ConvertUnit’ table. In my opinion, convert one unit to another unit is not that useful. People will not search a recipe with a specific measure unit.

According to the assumptions as well as the difficulty of implementation, we design a ER diagram as follows(make some changes on ER diagram in Project 1).

2.2.2 Relational Schema

We convert the above ER diagram into relational schema as follows.

User(username, password, email, firstname, lastname, address, gender, photo, description)

(username) is the primary key.

Recipe(rid, username, title, description, serving, direction, cooktime, picture, posttime)

(rid) is the primary key.

(username) is the foreign key references User(username).

RecipeIngredient(rid, iname, amount, unit)

(rid, iname) is the primary key.
(rid) is the foreign key references Recipe(rid).
(unit) is the foreign key references ConvertUnit(unit).

RecipeTag(rid, tname)

(rid, tname) is primary key.
(rid) is the foreign key references Recipe(rid)

Review(reviewid, username, rid, rtime, rtitle, reviews, rating, suggestions, picture)

(reviewid) is primary key.
(username) is the foreign key references User(username).
(rid) is the foreign key references Recipe(rid).

CookingGroup(gid, gname, description, creator)

(gid) is primary key.
(creator) is the foreign key references User(username).

GroupMember(gid, username)

(gid, username) is primary key.
(gid) is the foreign key references CookingGroup(gid).
(username) is the foreign key references User(username).

GroupMeeting(mid, mname, gid, organizer, starttime, endtime, mdescription, location)

(mid) is primary key.
(gid) is the foreign key references CookingGroup(gid).
(organizer) is the foreign key references User(username).

MeetingMember(mid, username, rating, report, sendRSVP)

(mid, username) is primary key.
(mid) is the foreign key references GroupMeeting(mid).
(username) is the foreign key references User(username).

The design of the schema can be divided into 4 parts:

1. User account —— User.
2. Recipe information —— Recipe, RecipeIngredient, RecipeTag
3. Review information —— Review

4. Group and meeting —— CookingGroup, GroupMember, GroupMeeting, MeetingMember

Here we will explain the schema in detail.

1. User

1) User(username, password, email, firstname, lastname, address, gender, photo, description)

For each user, we will store a unique username(as login name), password, email. The email is used to log in. All other fields are used as profile attributes.

In all tables, we only stored pictures' url. All the photos are stored in our working directory and the url can be used to load the picture.

2. Recipe information

1) Recipe(rid, username, title, description, serving, direction, cooktime, picture, posttime)

A user can post many recipes in the website. We use a unique rid to identify a recipe and store the username who posts the recipe as well as the post time. Generally, a recipe contains a title, ingredients and its amount, directions about how to prepare the food, number of servings, cook time and a cover photo. So we store all these information in Recipe except ingredients.

2) RecipeIngredient(rid, iname, amount, unit)

Ingredients were stored individually in RecipeIngredient table. Each recipe must contain multiple ingredients. It is a one to many relationship between recipe and ingredients. Therefore, storing ingredients in Recipe table will cause redundancy and waste lots of space. Therefore, we use another RecipeIngredient table to store recipe and its ingredients.

3) RecipeTag(rid, tname)

Users can add some tags for a recipe. Same as ingredients, we store tags in an individual RecipeTag table to decrease redundancy and save space.

3. Review Information

1) Review(reviewid, username, rid, rtime, rtitle, reviews, rating, suggestions, picture)

A user can review a recipe. A review contains a unique reviewid, username of the review user, review time(rtime) , review title(rtitle), reviews, a rating from 1 ~ 5 star, and suggestions. A picture can also uploaded.

4. Group and meeting

1) CookingGroup(gid, gname, description, creator)

2) GroupMember(gid, username)

Any users can create and join informal groups that organize cooking events without being authorized. A group has a unique id, a group name, a description about it and a creator who create the group.

Users can join groups as members. The GroupMember table keeps the relationship between a group and its member's username.

- 3) GroupMeeting(mid, mname, gid, organizer, starttime, endtime, mdescription, location)
- 4) MeetingMember(mid, username, rating, report, sendRSVP, picture)

A group can held meetings or events for its members. A meeting is organized by a user and contains a unique meeting id, a meeting name, the group id, username of the organizer, start and end time, a short description and location.

Group members can join group meeting via sending RSVP. After the meeting, meeting members can rating the meeting and write a report about it. So in the table MeetingMember, each row represents a member who want to join the meeting. It contains the meeting id, username, rating for the meeting from 1 ~ 5 star and a optional report.

2.2.3 Sample SQL Queries

1. Create a record for a new user account, with a name, a login name, and a password.

Insert into User values

```
('Mr.Cool', '234567', 'mc@gmail.com', 'Jason', 'Yang', 'Male', "photo/url", 'My name is Jason. I live in New York. I love curry food. I am studying computer science. My daddy is a chef.');
```

2. List all recipes with tag “italian” that contain the keyword ``broccoli”.

Select rid, title from

```
Recipe natural join RecipeTag  
where tname = 'italian'  
and description like '%broccoli%' ;
```

3. List all members of the group “Park Slope Cake Club” that have given a positive RSVP to more than three events of the group.

Suppose the group id of “Park Slope Cake Club” is 1.

Select mm.username From

```
GroupMeeting gm natural join MeetingMember mm  
where gm.gid = 1 and mm.sendRSVP = 1  
Group By mm.username  
Having count(*) >= 3;
```

4. Add a review with title "Yummy!", text "Really, really, tasty!", and a rating of 5 stars to the recipe for "Grandma's Fettuccini Alfredo".

Suppose the recipe id of recipe "Grandma's Fettuccini Alfredo" is 4.

insert into Review values

```
(1, 'basketball_lord', 4, '2016-10-19 18:00:00', 'Yummy!', 'Really, really, tasty!', 5,  
'no', 'picture/url');
```

5. List all recipes containing the word "tuna", sorted from highest to lowest average rating.

select rc.rid, rc.title, Avg(rv.rating) as rating from

 Recipe rc left join Review rv on rc.rid = rv.rid

 where rc.description like '%tuna%'

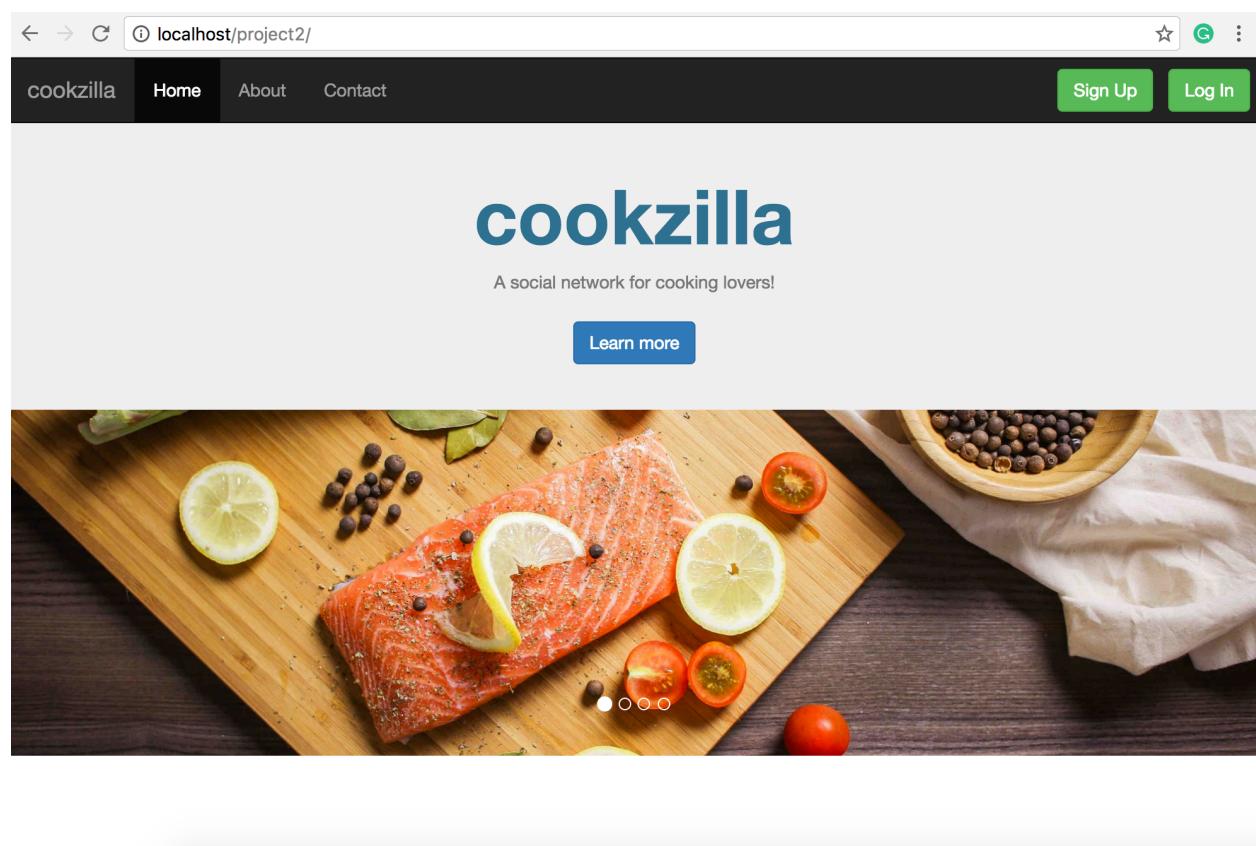
 Group By rc.rid

 Order By Avg(rv.rating) Desc

2.2.3 Back-end Implementation

1. default homepage(index.html)

The user need to sign up or log in to further steps.



2. User Register(register.php)

The user need to file a form with information of the login. All the fields are needed to fill out and then go to next step.

when the user click the button, all the information will be POST to the server, and the server can get these information by `$_POST`, and store them in MySQL.

localhost/project2/register.php

Join now.

Start your trip to cookzilla.

Enter Username
Please enter your full name.

Enter Your Email
Please enter valid email address.

Enter Password
Please enter password.

Sign Up

Already have an account? [Log In](#)

3. Log in(login.php)

User need to log in by entering his email and password. All fields are needed to fill out.

localhost/project2/login.php

Welcome!

Begin to explore cookzilla.

Your Email
Please enter your email address.

Your Password
Please enter your password.

Remember Me | [Forgot Password?](#)

Log In

Don't have an account? [Sign up.](#)

4. Homepage (home.php)

When logged in, the website will jump to homepage of the user. In the navigation bar, it will show a search bar to search recipes. In the right is the drop down list. It provide some useful functions to the user.

It will show a slide with random food pictures.

Then it will show top 5 rated recipes with rating scores(because the sample data is not enough, we just get top5).It will also show top 5 latest recipes.

In the bottom, it will show all the recipe tags. Each tag is a link which can jump to a page list all recipes with that tag.

The screenshot shows a web browser window with the URL `localhost/project2/home.php`. The page has a dark header with the logo "cookzilla" and navigation links for "Home", "About", and "Contact". A search bar contains the word "pizza". On the right, a user profile for "sweetyHeart" is shown with options like "Add a Recipe", "My Recipes", "Profile", "Favorate", "Group", "Friends", and "Log out". Below the header is a large image of two wraps filled with meat, lettuce, and radishes. At the bottom, a message says "Hi, sweetyHeart! Now it's time to play in Cookzilla!".

Top 5 rated Recipes!

ID	Made By	Title	Rating Score	Post Time	Picture
3	Mr.Cool	curried chicken	5.0000	2016-02-18 03:50:39	

Recent Recipes					
ID	Made By	Title	Rating Score	Post Time	Picture
8	Mr.Cool	Roasted Tuna	4.0000	2016-05-29 13:21:39	
2	akb48	tuna sushi	5.0000	2016-03-08 16:50:39	

Top 5 Latest Recipes!

ID	Made By	Title	Rating Score	Post Time	Picture
48	Mr.Cool	last	5.0000	2016-12-14 16:04:26	
4	sweetyHeart	Grandma's Fettuccini Alfredo	5.0000	2016-10-18 10:50:39	
8	Mr.Cool	Roasted Tuna	4.0000	2016-05-29 13:21:39	

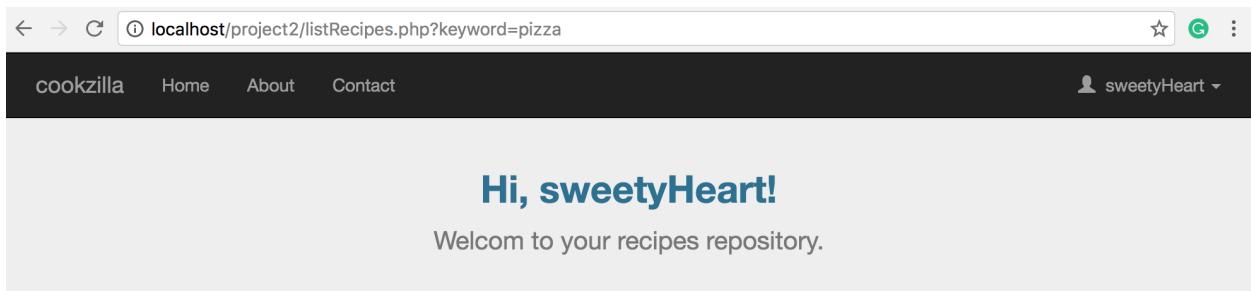
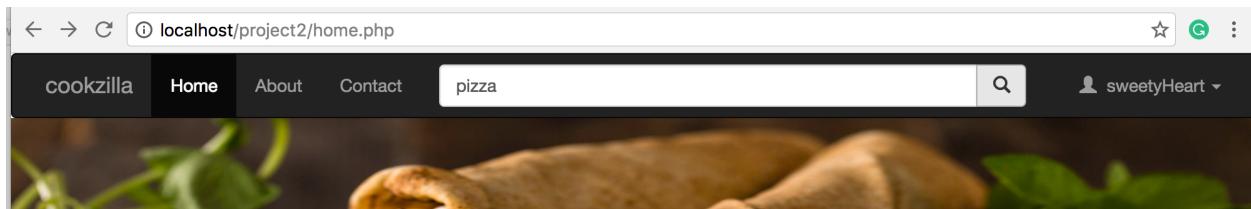
ID	Made By	Title	Rating Score	Post Time	Picture
48	Mr.Cool	last	5.0000	2016-12-14 16:04:26	
4	sweetyHeart	Grandma's Fettuccini Alfredo	5.0000	2016-10-18 10:50:39	
8	Mr.Cool	Roasted Tuna	4.0000	2016-05-29 13:21:39	
2	akb48	tuna sushi	5.0000	2016-03-08 16:50:39	
3	Mr.Cool	curried chicken	5.0000	2016-02-18 03:50:39	

Tags

italian, pizza, veggie, fish, japanese, sushi, Tuna, chicken, curry, Indian, Chili, Chinese, Spicy, spaghetti, American, Burger, roast, seafood, butter, cake, dessert, sweet, dfa, ga, df, adfa, adf, ff, f, a, adg,

5. Search a Recipe(listRecipes.php)

The user can search a recipe in the home page of the search bar. Here show a result.



User	Title
akb48	Veggie Pizza
EatMonster	Veggie Pizza

6. Show a recipe.(listRecipes.php)

In a search result or other result page, the recipe names are links to show its content. It will pass the recipe ID to the new page. The listRecipes.php will use _GET and show results.

Here is the results of a Recipe: "Roasted Tuna". It will show the name and maker of the recipe. It will also provide an averaging score of the recipe.

The user's name is also a link which will show all the user's recipes(show this in point 10.)

Ingredients and directions are listed line by line.

Of course , user can check out the reviews and write a review.

← → C ⓘ localhost/project2/Recipe_display.php?rid=8 ⋮

cookzilla Home About Contact ⋮

👤 sweetyHeart

Hi, sweetyHeart!

Check out your interesting recipes.

Roasted Tuna

-----By Mr.Cool

Average rating score: 4.0000

(The most famous home made style Italian custom pizza.)



← → C ⓘ localhost/project2/Recipe_display.php?rid=8 ⋮

List reviews

Cooktime

40 min

Serving

3

Ingredient

Black pepper:

olive oil: 2 teaspoons

sea salt: 1 teaspoon

tuna fillets: 4 pieces

Direction

1. Heat oven to 400 degrees.

2. Drizzle tuna with olive oil, sprinkle with salt and pepper and place on a rimmed baking sheet, skin side down if you have left the skin on.

3. Roast fish for 10 minutes per inch of thickness. Serve with lemon wedges, drizzled with more good olive oil.

Write a review

7. List Reviews(listReview.php)

User can click 'list reviews' and get to this page. It get the recipe id (rid) and name and search in the database about the data. Then list reviews in a table.

The screenshot shows a web browser window with the URL `localhost/project2/listReviews.php?rid=8&&rname=Roasted%20Tuna`. At the top, there's a header bar with a back/forward button, a search bar containing the URL, and some icons. Below the header is a navigation bar with links for 'cookzilla', 'Home', 'About', 'Contact', and a user profile 'sweetyHeart'. The main content area has a greeting 'Hi, sweetyHeart!' and a sub-greeting 'Reviews can help you better.' Below this, the title 'Reviews for Roasted Tuna' is displayed. A table titled 'Reviews' lists one review entry:

Reviewed By	title	contents	rating score	suggestion
RockTheEarth	Tuna!1!	Really good experience. Thanks for sharing. Tuna is my favorite! Go go tuna!	4	

8. Write Reviews(addReviews)

The user can write his/her own reviews. They need to grade it with score 1-5. The score was then add to database and used to get average score for a recipe.

The screenshot shows a web browser window with the URL `localhost/project2/addReviews.php?rid=8`. The page has a header with a back/forward button, a search bar, and some icons. The main content area starts with the text 'Add a review.' Below this, the title 'Roasted Tuna' is displayed, followed by the text '----By Mr.Cool'. There is a form for writing a review:

Rating: 5
4
3
2
1

Title: Goo

Review: I like it.

Suggestion: make it longer

9. Add Recipes(addRecipes.php)

User can add recipes from the navigation bar with needed information.

localhost/project2/addRecipes.php

cookzilla Home About Contact sweetyHeart

Hi, sweetyHeart!
Welcom to your recipes repository.

Add Your Recipe

Title:

Description:

Ingredient	Amount	Unit
<input type="text" value="Enter ingredient name."/>	<input type="text" value="Amount(ex.,1,many)"/>	<input type="text" value="Unit or leave blank"/>
More		

Direction:

localhost/project2/addRecipes.php

Number of serving:

Serving Number is needed.

cookeime:

Cookeime is needed.

Tags:
 More

upload a photo
 No file chosen
picture is needed.

Submit

10. Show my recipes or a user's recipe lists.(myRecipes.php, userRecipes.php)

localhost/project2/myRecipes.php

cookzilla Home About Contact sweetyHeart

Hi, sweetyHeart!

Welcome to your recipes repository.

Title	Post Time
Grandma's Fettuccini Alfredo	2016-10-18 10:50:39
fad	2016-12-10 00:36:00
fad1	2016-12-10 00:36:00

localhost/project2/userRecipes.php?username=Mr.Cool

cookzilla Home About Contact sweetyHeart

Hi, sweetyHeart!

Mr.Cool's recipes repository.

Title	Post Time
curried chicken	2016-02-18 03:50:39
Hamburgers	2016-07-18 15:50:39
Roasted Tuna	2016-05-29 13:21:39

11. Update profile(profile.php)

User can update their profile from the right above user button.

localhost/project2/profile.php

cookzilla Home About Contact sweetyHeart

localhost says:
Successfully Updated ...

OK

Hi, sweetyHeart!

Update your public profile.

This is your photo.

Choose File No file chosen

First Name: Bear

Last Name: nike

Gender: Male

Address: ggg

About me: I heard Chinese eat everything?! How brave and strange it is!! But just lets eat everything!

Save

12. Show groups.(group.php)

The user can go to Group module. It will list all groups in this website(now in total 3 groups). The user can join one or create his/her own group(createGroup.php). ' My groups' button show all groups current user is in. If the user click the link of the group id, it will redirect to a page for that specific group.

This screenshot shows the 'group.php' page. At the top, there's a navigation bar with links for Home, About, Contact, and a user profile for 'sweetyHeart'. A dropdown menu for 'sweetyHeart' is open, showing options like 'Add a Recipe', 'My Recipes', 'Profile', 'Favorate', 'Group' (which is highlighted in blue), and 'Friends'. Below the navigation, a message says 'Hi, sweetyHeart! Check out your interesting group.' A table lists three groups:

groupId	name	creater	Action
1	Park Slope Cake Club	peace	Join
2	Brooklyn Burgers	Heyman	Join
3	Chicken is best	sweetyHeart	Join

At the bottom, there are two buttons: 'Create a Group' (red) and 'My Groups' (blue).

This screenshot shows the 'createGroup.php' page. The top navigation bar is identical to the first screenshot. The main content area has a heading 'Hi, sweetyHeart!' and a sub-instruction 'Create a group here.' Below this is a 'Create a Group' section with two input fields: 'Group Name:' and 'Description:', both with placeholder text. A green 'Submit' button is at the bottom.

This screenshot shows the 'myGroups.php' page. The top navigation bar is identical. The main content area has a heading 'Hi, sweetyHeart!' and a sub-instruction 'Check out groups your joint.' A table lists the groups the user is part of:

groupId	name	creater
1	Park Slope Cake Club	peace
2	Brooklyn Burgers	Heyman

At the bottom, there is a red 'Create a Group' button.

13. Meetings in a group(groupMeeting.php)

The user can join a meeting in his/her own group. For example, there are 4 meetings to join in group 1 for the current user. The user can also set a meeting or list all meetings.

localhost/project2/groupMeeting.php?gid=1

Group1: Park Slope Cake Club

----By peace

description: Love cakes, make cakes!

Members:

akb48. basketball_lord. HeyMan. peace. RockTheEarth. SingleMom.
sweetyHeart.

Meetings

Meeting ID	Meeting Name	Organizer	Action
1	Cheese Cake	peace	Join
2	Egg Cake	peace	Join
3	Fruit Cake	akb48	Join
4	Coffee Cake	akb48	Join

localhost/project2/createMeeting.php?gid=1

Hi, sweetyHeart!

Set a meeting here.

Set a Meeting

Meeting title:

Enter a Meeting Title

Start Time:

Enter a Start Time

End time:

Enter a End Time

Description:

Enter group description.

Location:

Enter a Location

Submit

14. Check a meeting(MeetingDisplay.php)

User can check information of a meeting. That will show all information including the participants and RSVPs.

The screenshot shows a web browser displaying a meeting details page. The URL in the address bar is `localhost/project2/meeting_display.php?mid=1`. The main title is "Meeting1: Cheese Cake" and it is "By peace". The description is "please join on time." The start time is "2016-09-01 15:00:00" and the end time is "2016-09-01 18:00:00". The location is "460 Ocean Park". Below this, there is a section titled "Meeting Members" with a table showing the user names and their RSVP status:

User Name	RSVP
akb48	1
HeyMan	1
peace	1
SingleMorn	1
sweetyHeart	0

15. RSVP for a meeting

The user can send RSVP also for a meeting.

The screenshot shows a web browser displaying an RSVP form. The URL in the address bar is `localhost/project2/join_meeting.php?mid=1`. The top navigation bar includes links for "Home", "About", "Contact", and a user profile "Mr.Cool". The main content area displays a greeting "Hi, Mr.Cool!" and a prompt "RSVP.". Below this, there is a large button labeled "RSVP". Underneath the button, the text "Do you want to sent RSVP?" is displayed. A form field labeled "RSVP:" with the placeholder "Enter a Start Time" is present, along with a green "Submit" button.

3. Conclusion

In this project , we build a website by using PHP and MySQLi. It provides the basic function of a website.

However, there are still lots of aspects which can be improved in the future.

By building the website, I understand SQL more and learn the design of a database is the most important thing. By designing a good relational database, it will more easy to build a website on it.

Also, I learn that when transferring data in internet, we need to use some method to prevent SQL injection. We can use stored procedures and prepared statements . We also need to use encryption to protect safe.