CHAPTER 1

INTRODUCTION

* The Recipe world is a website which gathers a variety of food recipes from all over the world to appreciate the culinary culture spread across various regions.
* It is an interface where users can upload their recipes or make a list of their favorite recipes given by other users from different regions.
* The recipe uploaded will be reviewed and rated by other users and will build the overall rating of the user.
* The users can sort their list of favorite recipes given by other users according to their liking.
* From recipes of exotic food to the popular dishes, will be available to users.
* The users can also explore various food coming from different regions in a country

The recipe world is free to prescribe, generate and use the functioning and maintaining the quality of services.

1.1 Problem Definition

This project makes things easier for people to learn the recipes of different cuisines, this also lets people to share their own recipes to others.

The purpose of the project entitled as THE RECIPE WORLD is to share multiple recipes to multiple number of people around the world. This project enables sharing of recipes of different cuisines around the people. which gathers a variety of food recipes from all over the world to appreciate the culinary culture spread across various regions.

1.1 **Need**

* It is an interface where users can upload their recipes or make a list of their favorite recipes given by other users from different regions.

* From recipes of exotic food to the popular dishes, will be available to users

* The users can also explore various food coming from different regions in a country

* This can also be a place where users can share their own recipes to the other users.

* This enables people to learn and try new variety recipes, popular dishes on their own with the steps for making a dish.

* This makes things easier for users and saves a lot of time

CHAPTER 2

REQUIREMENTS

2.1 Software Requirement Specifications

Operating System Front End Back End Server Documentation :

1.HTML

2.CSS

3.JAVASCRIPT

4.AJAX

5.MYSQL

6. BOOTSTRAP

2.2 Hardware Requirement Specifications

Computer Processor Core i5 Processor
Speed 2.3 GHz Processor
Hard Disk 400 GB or more
RAM Min 2GB

CHAPTER 3

ENTITY RELATIONSHIP DIAGRAM

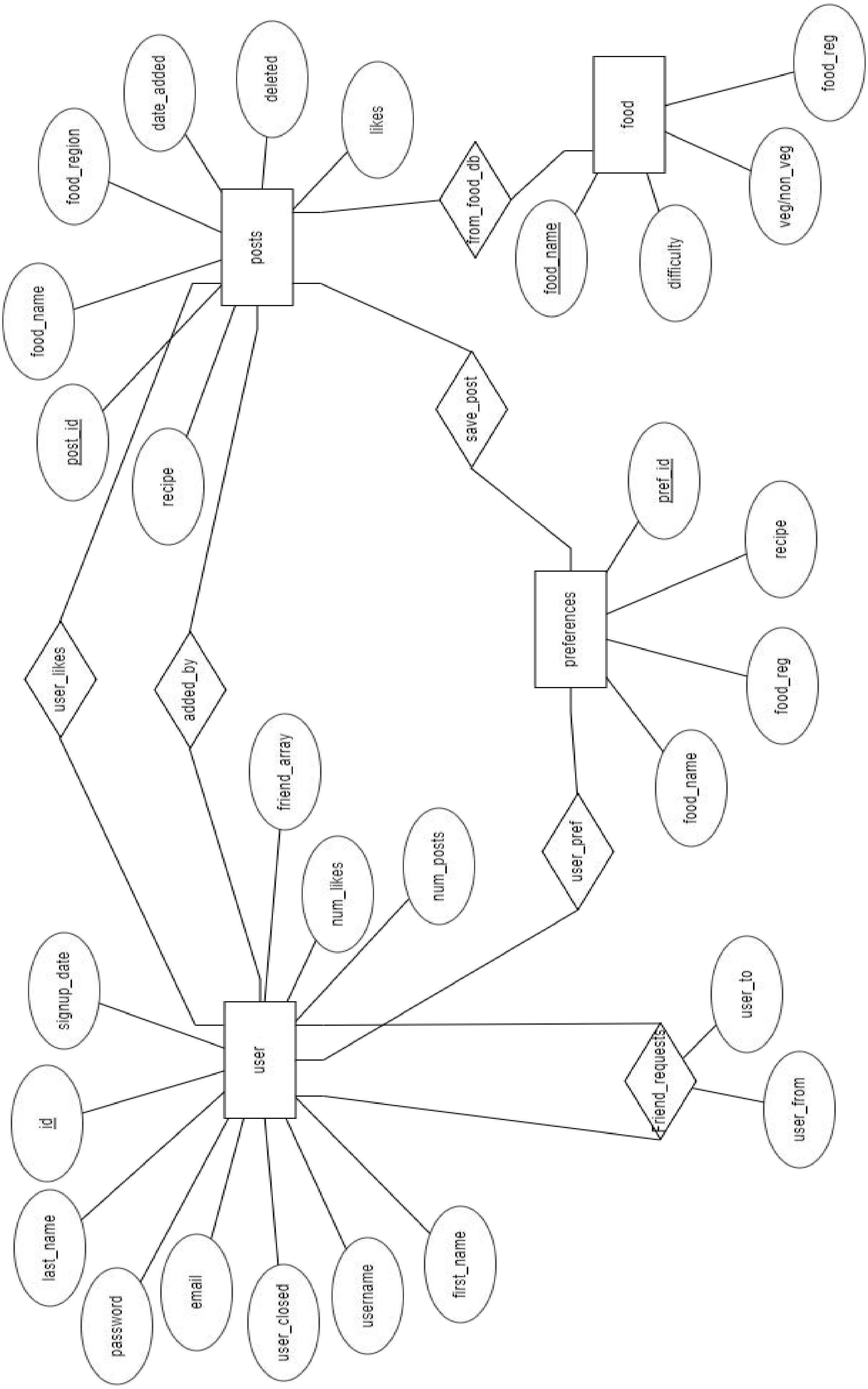
An entity type is represented in ER diagrams as a rectangular box enclosing the entity type name. Attribute names are enclosed in ovals and are attached to their entity type by straight lines. Composite attributes are attached to their component attributes by straight lines. Multivalued attributes are displayed in double ovals.

In ER diagrammatic notation, each key attribute has its name underlined inside the oval

In ER diagrams, relationship types are displayed as diamond-shaped boxes, which are connected by straight lines to the rectangular boxes representing the participating entity types. The relationship name is displayed in the diamond-shaped box

Cardinality ratios for binary relationships are represented on ER diagrams by displaying 1, M, and N on the diamonds

In ER diagrams, total participation (or existence dependency) is displayed as a double line connecting the participating entity type to the relationship, whereas partial participation is represented by a single line


posts
food_region
date_added
deleted
likes
food_name
post_id
recipe
from_food_db
food
food_name
difficulty
veg/non_veg
food_reg
save_post
preferences
pref_id
recipe
food_reg
food_name
user_likes
added_by
user_pref
user
friend_array
num_likes
num_posts
signup_date
id
last_name
password
email
user_closed
username
first_name
friend_requests
user_to
user_from

CHAPTER 4

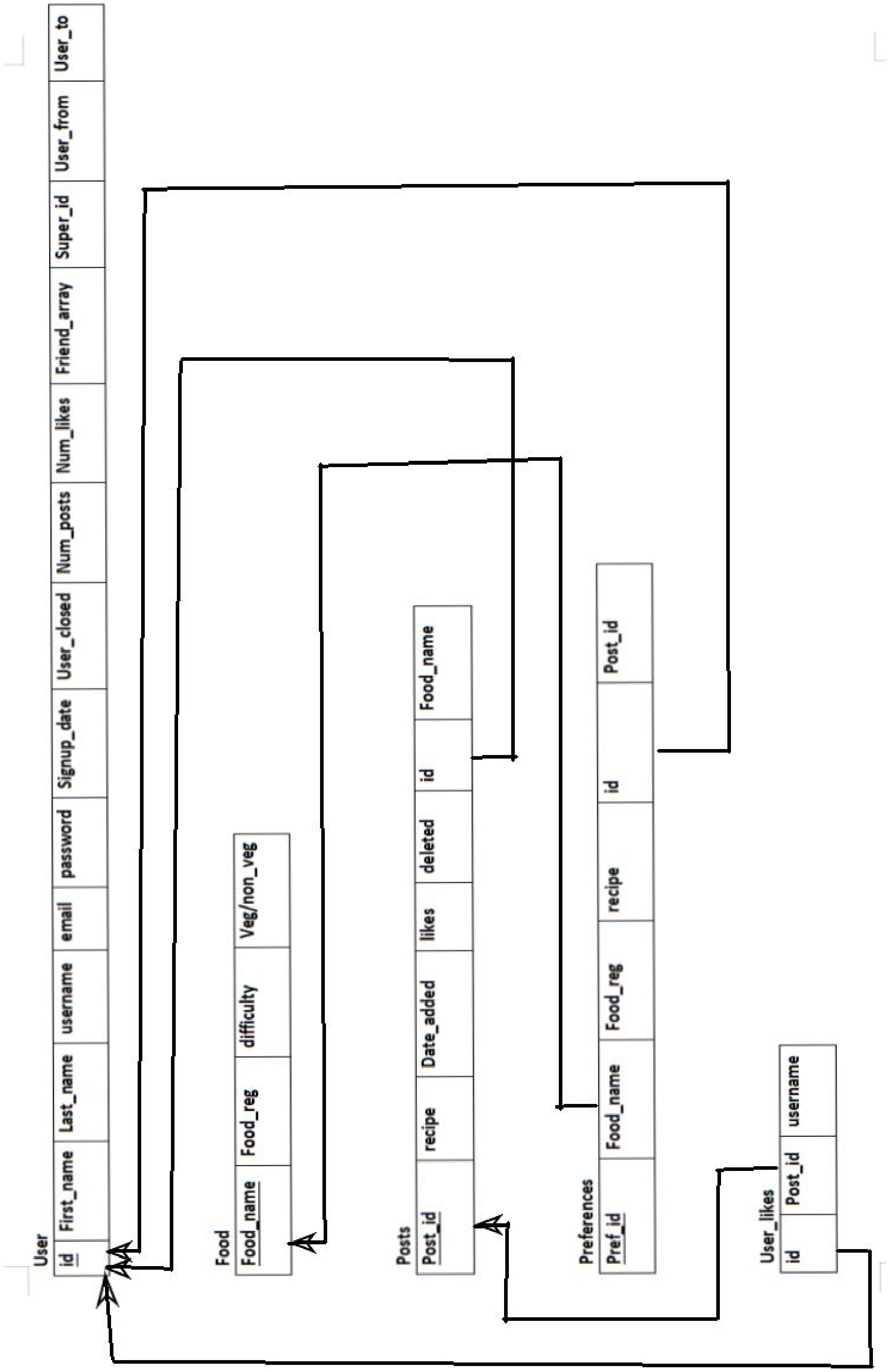
SCHEMA DIAGRAM

Most data models have certain conventions for displaying schemas as diagrams.A displayed schema is called a schema diagram.

A schema diagram displays only some aspects of a schema, such as the names of record types and data items, and some types of constraints. Other aspects are not specified in the schema diagram

A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database.

Schema can be a single table or it can have more than one table which is related. The schema represents the relationship between these tables.


User
id
First_name
Last_name
username
email
password
Signup_date
User_closed
Num_posts
Num_likes
Friend_array
Super_id
User_from
User_to
Food
Food_name
Food_reg
difficulty
Veg/non_veg
Posts
Post_id
recipe
Date_added
likes
deleted
id
Food_name
Preferences
Pref_id
Food_name
Food_reg
recipe
id
Post_id
User_likes
id
Post_id
username

CHAPTER 5

IMPLEMENTATION

5.1 Backend Implementation

```
CREATE TABLE `friend_requests` (
`id` int(11) NOT NULL,
`user_to` varchar(50) NOT NULL,
`user_from` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `likes` (
`id` int(11) NOT NULL,
`username` varchar(60) NOT NULL,
`post_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `likes` (`id`, `username`, `post_id`) VALUES
(1, 'anmol', 100),
(2,’Balaji’,100);

CREATE TABLE `food` (
`food_name` text NOT NULL,
`food_reg` text NOT NULL,
`difficulty` text NOT NULL
`veg/non_veg` varchar(10) NOT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `food` (‘food_name’,’food_reg’,’difficulty’,’veg/non_veg’) VALUES
(‘maggi’,’india’,’easy’,’veg’);



CREATE TABLE `preferences` (
`pref_id` int(3) NOT NULL,
`food_reg` text NOT NULL,
`food_name` text NOT NULL
`recipe` text NOT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `posts` (
`id` int(11) NOT NULL,
`recipe` text NOT NULL,
`food_region` text NOT NULL
‘food_name’ references food(food_name) on delete cascade,
`added_by` varchar(60) NOT NULL,
`user_to` varchar(60) NOT NULL,
`date_added` datetime NOT NULL,
`user_closed` varchar(3) NOT NULL,
`deleted` varchar(3) NOT NULL,
`likes` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
INSERT INTO `posts` (`id`, `recipe`,’food_region,’food_name’, `added_by`, `user_to`, `date_added`,
`user_closed`, `deleted`, `likes`) VALUES

(1,’Take one and a half cup of water in a pan.Heat the pan on medium flame.When the water comes to
boil, add the,Noodles Cake to the pan.Cover it with a lid for a minute.After a minute, uncover the lid
and add the tastemaker to the pan.Mix it well.Without breaking the Noodles.Switch off the flame
when all the Water boils.Enjoy the hot Maggi in this lovely weather!’,’india’,’anmol’,none,’01 jan
2022’,’no’,’no’,0);
```

```
CREATE TABLE `users` (

`id` int(11) NOT NULL,

`first_name` varchar(25) NOT NULL,

`last_name` varchar(25) NOT NULL,

`username` varchar(100) NOT NULL,

`email` varchar(100) NOT NULL,

`password` varchar(255) NOT NULL,

`signup_date` date NOT NULL,

`num_posts` int(11) NOT NULL,

`num_likes` int(11) NOT NULL,

`user_closed` varchar(3) NOT NULL,
`friend_array` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `users` (`id`, `first_name`, `last_name`, `username`, `email`, `password`,
`signup_date`, `num_posts`, `num_likes`, `user_closed`, `friend_array`) VALUES

(1,’anmol’,’vm’,’avm007’,’avm007@gmail.com’,’’,’2022-01-02’,1,0,no,no,balaji);




ALTER TABLE `friend_requests`
ADD PRIMARY KEY (`id`);


--
-- Indexes for table `likes`
--
ALTER TABLE `likes`
ADD PRIMARY KEY (`id`);

ALTER TABLE `posts`
ADD PRIMARY KEY (`id`);


--
-- Indexes for table `users`
--
ALTER TABLE `users`
ADD PRIMARY KEY (`id`);

ALTER TABLE `food`
ADD PRIMARY KEY (`food_name`);

ALTER TABLE `preferences`
ADD PRIMARY KEY (`pref_id`);
```

Frontend to backend queries:

```
$check_database_query = mysqli_query($con, "SELECT * FROM users WHERE email='$email' AND password='$password'");
$check_login_query = mysqli_num_rows($check_database_query);
if($check_login_query == 1) {
$row = mysqli_fetch_array($check_database_query);
$username = $row['username'];
// Update's user_closed='yes' for successfull logging in
$user_closed_query = mysqli_query($con, "SELECT * FROM users WHERE email='$email' AND user_closed='yes'");
if(mysqli_num_rows($user_closed_query) == 1) {
$reopen_account = mysqli_query($con, "UPDATE users SET user_closed='no' WHERE email='$email'");
}
$_SESSION['username'] = $username; // Storing username into session variable
header("Location: index.php"); // Redirecting to index page
exit();
}
else {
array_push($error_array, "Wrong Credentials..!!<br>");
}


$query = mysqli_query($this->con, "INSERT INTO posts VALUES('', '$body', '$added_by', '$user_to', '$date_added', 'no', 'no', '0')");
$returned_id = mysqli_insert_id($this->con); // which returns the id of post
$num_posts = $this->user_obj->getNumPosts();
$num_posts++;
$update_query = mysqli_query($this->con, "UPDATE users SET num_posts='$num_posts' WHERE username='$added_by'");
```

```
$data_query = mysqli_query($this->con, "SELECT * FROM posts WHERE deleted='no' ORDER BY id DESC");
if(mysqli_num_rows($data_query) > 0) {
$num_iterations = 0; // Number of results checked
$count = 1;

while($row = mysqli_fetch_array($data_query)) {
$id = $row['id'];
$body = $row['body'];
$added_by = $row['added_by'];
$date_time = $row['date_added'];
if($row['user_to'] == "none") {
$user_to = "";
}
else {
$user_to_obj = new User($this->con, $row['user_to']);
$user_to_name = $user_to_obj->getFirstAndLastName();
$user_to = "to <a href='" . $row['user_to'] ."'>" . $user_to_name . "</a>";
}
$user_details_query = mysqli_query($this->con, "SELECT first_name, last_name, profile_pic FROM users WHERE username='$added_by'");
$user_row = mysqli_fetch_array($user_details_query);
$first_name = $user_row['first_name'];
$last_name = $user_row['last_name'];
$profile_pic = $user_row['profile_pic'];

$user_details_query = mysqli_query($con, "SELECT * FROM users WHERE username='$username'"); // Query to get the details of user
$user_array = mysqli_fetch_array($user_details_query);

$num_friends = (substr_count($user_array['friend_array'], ",")) - 1; // To count the number of friends using comma

}

// Insert post

$query = mysqli_query($this->con, "INSERT INTO posts VALUES('', '$body', '$added_by',
'$user_to', '$date_added', 'no', 'no', '0')");

$returned_id = mysqli_insert_id($this->con); // which returns the id of post

// Update post count for user

$num_posts = $this->user_obj->getNumPosts();
$num_posts++;
$update_query = mysqli_query($this->con, "UPDATE users SET num_posts='$num_posts' WHERE
username='$added_by'");



public function __construct($con, $user){
$this->con = $con;
$user_details_query = mysqli_query($con, "SELECT * FROM users WHERE username='$user'");
$this->user = mysqli_fetch_array($user_details_query);
}

// To get the username
```

```
public function getUsername() {

return $this->user['username'];

}


public function getNumberOfFriendRequests() {

$username = $this->user['username'];

$query = mysqli_query($this->con, "SELECT * FROM friend_requests WHERE
user_to='$username'");

return mysqli_num_rows($query);

}


// To get the number of posts


public function getNumPosts() {

$username = $this->user['username'];

$query = mysqli_query($this->con, "SELECT num_posts FROM users WHERE
username='$username'");

$row = mysqli_fetch_array($query);

return $row['num_posts'];

}


// To get the firstname and lastname


public function getFirstAndLastName() {

$username = $this->user['username'];

$query = mysqli_query($this->con, "SELECT first_name, last_name FROM users WHERE
username='$username'");

$row = mysqli_fetch_array($query);

return $row['first_name'] . " " . $row['last_name'];
```

```
}

// To check for friend request receieved or not

public function didReceiveRequest($user_from) {

$user_to = $this->user['username'];

$check_request_query = mysqli_query($this->con, "SELECT * FROM friend_requests WHERE
user_to='$user_to' AND user_from='$user_from'");

if(mysqli_num_rows($check_request_query) > 0) {

return true;
}
else {
return false;
}
}

// To check whether request is sent or not

public function didSendRequest($user_to) {

$user_from = $this->user['username'];

$check_request_query = mysqli_query($this->con, "SELECT * FROM friend_requests WHERE
user_to='$user_to' AND user_from='$user_from'");
```

```
if(mysqli_num_rows($check_request_query) > 0) {

return true;
}
else {

return false;
}
}

// To Remove friend

public function removeFriend($user_to_remove) {

$logged_in_user = $this->user['username']; // To get the logged in username

$query = mysqli_query($this->con, "SELECT friend_array FROM users WHERE username='$user_to_remove'"); // To get the friend of user_to_remove

$row = mysqli_fetch_array($query);

$friend_array_username = $row['friend_array']; // User_to_remove friend _array

$new_friend_array = str_replace($user_to_remove . ",", "", $this->user['friend_array']); // Replace with null in logged_in user's friend array by finding the user_to_remove substring

$remove_friend = mysqli_query($this->con, "UPDATE users SET friend_array='$new_friend_array' WHERE username='$logged_in_user'"); // Update friend array after replacing with null of logged_in user
```

```
$new_friend_array = str_replace($this->user['username'] . ",", "", $friend_array_username); // Same
process as above but in user_to_remove friend_array

$remove_friend = mysqli_query($this->con, "UPDATE users SET friend_array='$new_friend_array'
WHERE username='$user_to_remove'"); // Updating new friend array of user_to_remove
}

// To Send request

public function sendRequest($user_to) {

$user_from = $this->user['username']; // To get the username of logged_in user

// Query to insert into friend_requests

$query = mysqli_query($this->con, "INSERT INTO friend_requests VALUES(", '$user_to',
'$user_from')");
}
```

5.2 Frontend Implementation

```
("includes/classes/User.php"); // To include Us("includes/classes/Post.php"); // To include Post.php
file

if (isset($_SESSION['username($con, "SELECT * FROM users WHERE
username='$userLoggedIn'");


$user = mysqli_fetch_array($user_details_query);

}

else {

header("Location: register.php"); // If not set redirects to register.php

}


?>


<html>

<head>

<title>Explore The food world!!</title>


<!-- Javascript -->


<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

<script src="assets/js/bootstrap.js"></script>

<script src="assets/js/bootbox.min.js"></script>

<script src="assets/js/food.js"></script>

<script src="assets/js/jquery.jcrop.js"></script>

<script src="assets/js/jcrop_bits.js"></script>

<!-- CSS -->

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.0.13/css/all.css"
integrity="sha384-
```

```
DNOHZ68U8hZfKXOrtjWvjxusGo9WQnrNx2sqG0tfsghAvtVlRW3tvkXWZh58N9jp"
crossorigin="anonymous">
<link rel="stylesheet" type="text/css" href="assets/css/bootstrap.css">
<link rel="stylesheet" type="text/css" href="assets/css/style.css">
<link rel="stylesheet" href="assets/css/jquery.Jcrop.css" type="text/css" />
</head>
<body>
<!-- Navigation Bar -->
<div class="top_bar">
<!-- Navbar Logo Section -->


<div class = "logo">
<a href="index.php"><h2>RECIPES!</h2></a>
</div>
<div class="search">
<form action="search.php" method="GET" name="search_form">
<input type="text" onkeyup="getLiveSearchUsers(this.value, '<?php echo $userLoggedIn; ?>')"
name="q" placeholder="Search..." autocomplete="off" id="search_text_input">
<div class="button_holder">
<img src="assets/images/icons/magnifying_glass.png">
</div>
</form>
<!-- Search Section -->
<div class="search_results">
</div>
<div class="search_results_footer_empty">
</div>
</div>
<nav>
```

```
<!-- Firstname display in navbar -->
<a href="<?php echo $userLoggedIn; ?>">
<?php echo $user['first_name']; ?>
</a>
<!-- Home -->
<a href="index.php">
<i class="fas fa-home"></i>
</a>
<!-- Messages -->
<a href="messages.php">
<i class="fas fa-comments"></i>
</a>
<!-- Friend Requests -->
<a href="requests.php">
<i class="fas fa-users"></i>
</a>

<!DOCTYPE html>
<html>
<head>
<title>Explore the food world :)</title>
<link rel="stylesheet" type="text/css" href="assets/css/register_style.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
<script src="assets/js/register.js"></script>
</head>
<body>

<?php  // For hiding login/register form
```

```
if(isset($_POST['register_button'])) {

echo '
<script>

$(document).ready(function() {
$("#first").hide();
$("#second").show();
});
</script>
';
}
?>
<div class="wrapper">

<div class="login_box">

<div class="login_header">
<h1>The recipe world</h1>
<p> Login/Register </p>
</div>

</br>

<!-- Login Section -->

<div id="first">
```

```
<!-- Login form -->

<form action="register.php" method="POST">

<!-- Email Section -->

<input type="email" name="log_email" placeholder="Your Email Address" value="<?php
if(isset($_SESSION['log_email'])) {
echo $_SESSION['log_email'];
}
?>" required>
<br>

<!-- Password Section -->

<input type="password" name="log_password" placeholder="Password">
<br>

<!-- Error's Section -->

<?php if(in_array("Wrong Credentials..!!<br>", $error_array)) echo  "Wrong Credentials..!!<br>"; ?>

<!-- Login Button -->

<input type="submit" name="login_button" value="Login">
<br>
```

```
<!-- Link to register form -->

<a href="#" id="signup" class="signup">New to the recipe world? Register here!</a>

</form>

</div>

<!-- Register Section -->

<div id="second">

<!-- Register form -->

<form action="register.php" method="POST">

<!-- First Name Section -->

<input type="text" name="reg_fname" placeholder="Your First Name" value="<?php
if(isset($_SESSION['reg_fname'])) {
echo $_SESSION['reg_fname'];
}
?>" required>

<!-- Last Name Section -->

<input type="text" name="reg_lname" placeholder="Your Last Name" value="<?php
if(isset($_SESSION['reg_lname'])) {
```

```
echo $_SESSION['reg_lname'];
}
?>" required>
<br>

<!-- Error's Section -->

<?php if(in_array("Your First Name should be in between 2-25 characters</br>",$error_array)) echo
"Your First Name should be in between 2-25 characters</br>"; ?>
<?php if(in_array("Your Last Name should be in between 2-25 characters</br>",$error_array)) echo
"Your Last Name should be in between 2-25 characters</br>"; ?>

<!-- Email Section -->

<input type="email" name="reg_email" placeholder="Your Email" value="<?php
if(isset($_SESSION['reg_email'])) {
echo $_SESSION['reg_email'];
}
?>" required>

<!-- Confirm Email Section -->


<input type="email" name="reg_email2" placeholder="Confirm Email" value="<?php
if(isset($_SESSION['reg_email2'])) {
echo $_SESSION['reg_email2'];
}
?>" required>
<br>
```

```
<!-- Error's Section -->

<?php if(in_array("Email already in use</br>",$error_array)) echo "Email already in use</br>"; ?>
<?php if(in_array("Invalid email format</br>",$error_array)) echo "Invalid email format</br>"; ?>
<?php if(in_array("Emails doesn't match</br>",$error_array)) echo "Emails doesn't match</br>"; ?>

<!-- Register Button Section -->

<input type="submit" name="register_button" value="Register">
<br>

<!-- Succesfull Registered Message -->

<?php if(in_array("<span style='color: #14C800;'>You're all set! Go ahead and login!</span><br>",
$error_array)) echo "<span style='color: #14C800; margin-left: -80px;'>You're all set! Go ahead and
login!</span><br>"; ?>

<!-- Link to login form -->

<a href="#" id="signin" class="signin">Already have an account? Sign in here!</a>
</form>
</div>
</div>
</div>
</body>
</html>

<?php
```

```
if(isset($_POST['update_details'])) {

$first_name = $_POST['first_name'];
$last_name = $_POST['last_name'];
$email = $_POST['email'];

$email_check = mysqli_query($con, "SELECT * FROM users WHERE email='$email'");
$row = mysqli_fetch_array($email_check);
$matched_user = $row['username'];

if($matched_user == "" || $matched_user == $userLoggedIn) {
$message = "Details updated!<br><br>";

$query = mysqli_query($con, "UPDATE users SET first_name='$first_name',
last_name='$last_name', email='$email' WHERE username='$userLoggedIn'");
}
else
$message = "That email is already in use!<br><br>";
}
else
$message = "";


//*************************************************

if(isset($_POST['update_password'])) {

$old_password = strip_tags($_POST['old_password']);
$new_password_1 = strip_tags($_POST['new_password_1']);
$new_password_2 = strip_tags($_POST['new_password_2']);

$password_query = mysqli_query($con, "SELECT password FROM users WHERE
username='$userLoggedIn'");
$row = mysqli_fetch_array($password_query);
$db_password = $row['password'];

if(md5($old_password) == $db_password) {
```

```
if($new_password_1 == $new_password_2) {



if(strlen($new_password_1) <= 4) {
$password_message = "Sorry, your password must be greater than 4 characters<br><br>";
}
else {
$new_password_md5 = md5($new_password_1);
$password_query = mysqli_query($con, "UPDATE users SET password='$new_password_md5'
WHERE username='$userLoggedIn'");
$password_message = "Password has been changed!<br><br>";
}



}
else {
$password_message = "Your two new passwords need to match!<br><br>";
}

}
else {
$password_message = "The old password is incorrect! <br><br>";
}

}
else {
$password_message = "";
}



if(isset($_POST['close_account'])) {
header("Location: close_account.php");
}



?>
```

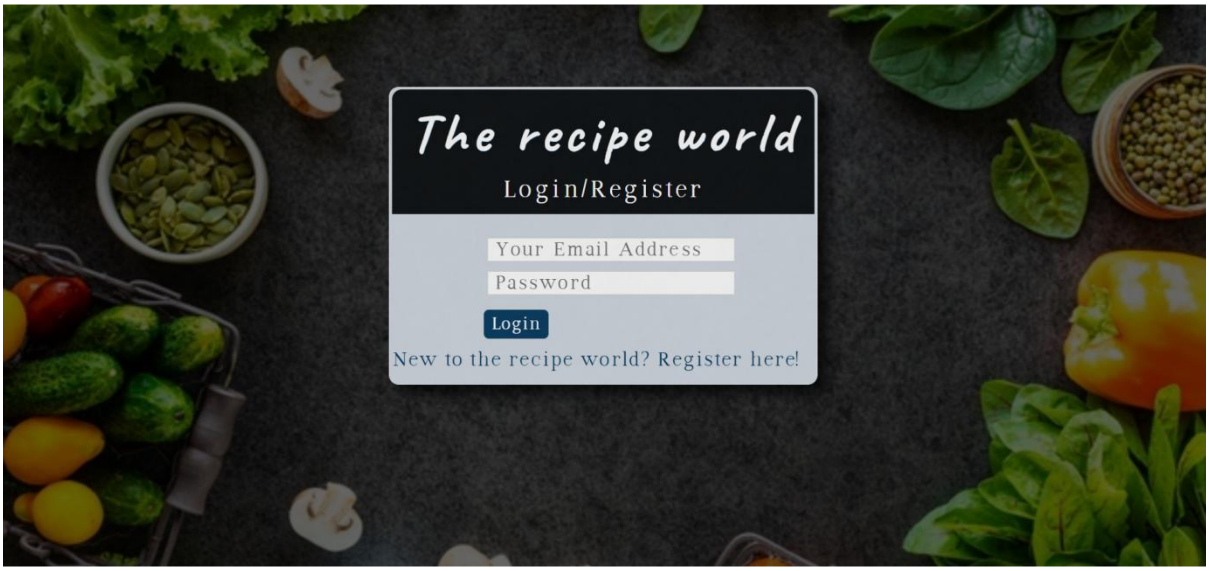
```
//delete post
<?php

require '../../config/config.php';

if(isset($_GET['post_id']))

$post_id = $_GET['post_id'];

if(isset($_POST['result'])) {

if($_POST['result'] == 'true')

$query = mysqli_query($con, "UPDATE posts SET deleted='yes' WHERE id='$post_id'");
}

?>

//create post
<?php

require '../../config/config.php';

if(isset($_GET['post_id']))

$post_id = $_GET['post_id'];

if(isset($_POST['result'])) {

if($_POST['result'] == 'true')

$query = mysqli_query($con, "UPDATE posts SET deleted='yes' WHERE id='$post_id'");
}

?>
```

```
public function submitPost($body, $user_to) {

$body = strip_tags($body); // Removes html tags
$body = mysqli_real_escape_string($this->con, $body); // Which escapes special characters in the
body
$check_empty = preg_replace('/\s+/', '', $body); // Deletes all spaces

if($check_empty != "") {



// Current date and time

$date_added = date("Y-m-d H:i:s");

// Get username

$added_by = $this->user_obj->getUsername();

// If user is on own profile, user_to is 'none'

if($user_to == $added_by)
$user_to = "none";
// Insert post
$query = mysqli_query($this->con, "INSERT INTO posts VALUES('', '$body', '$added_by',
'$user_to', '$date_added', 'no', 'no', '0')");

$returned_id = mysqli_insert_id($this->con); // which returns the id of post

// Update post count for user

$num_posts = $this->user_obj->getNumPosts();
$num_posts++;
$update_query = mysqli_query($this->con, "UPDATE users SET num_posts='$num_posts' WHERE
username='$added_by'");

}
}
```

CHAPTER 6

SNAPSHOTS

Login page



The recipe world

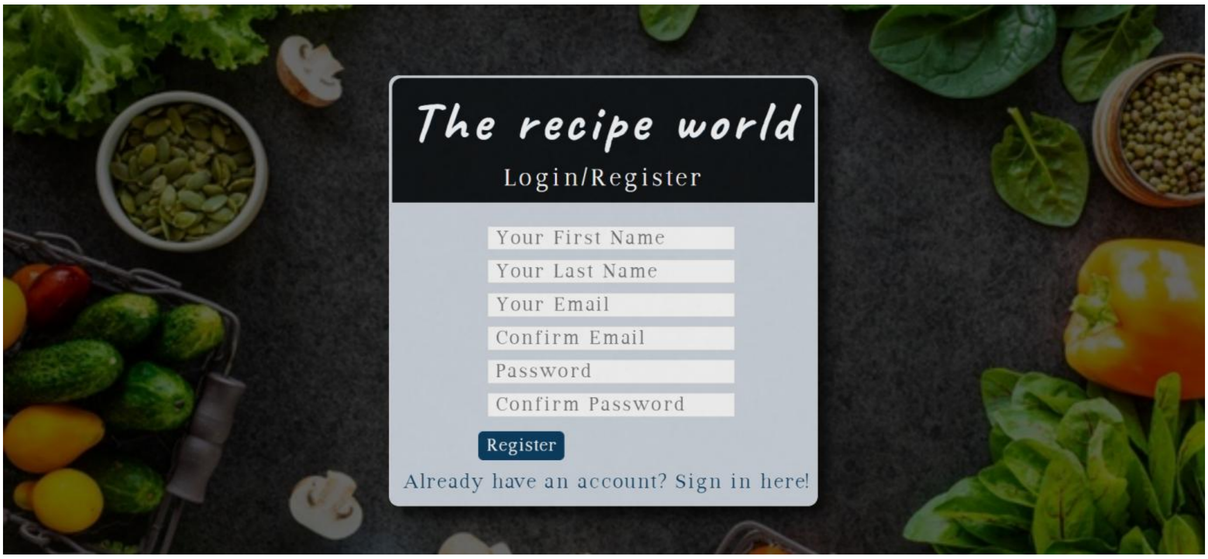
Login/Register

Your Email Address

Password

New to the recipe world? Register here!

Registration page



The recipe world

Login/Register

Your First Name

Your Last Name

Your Email

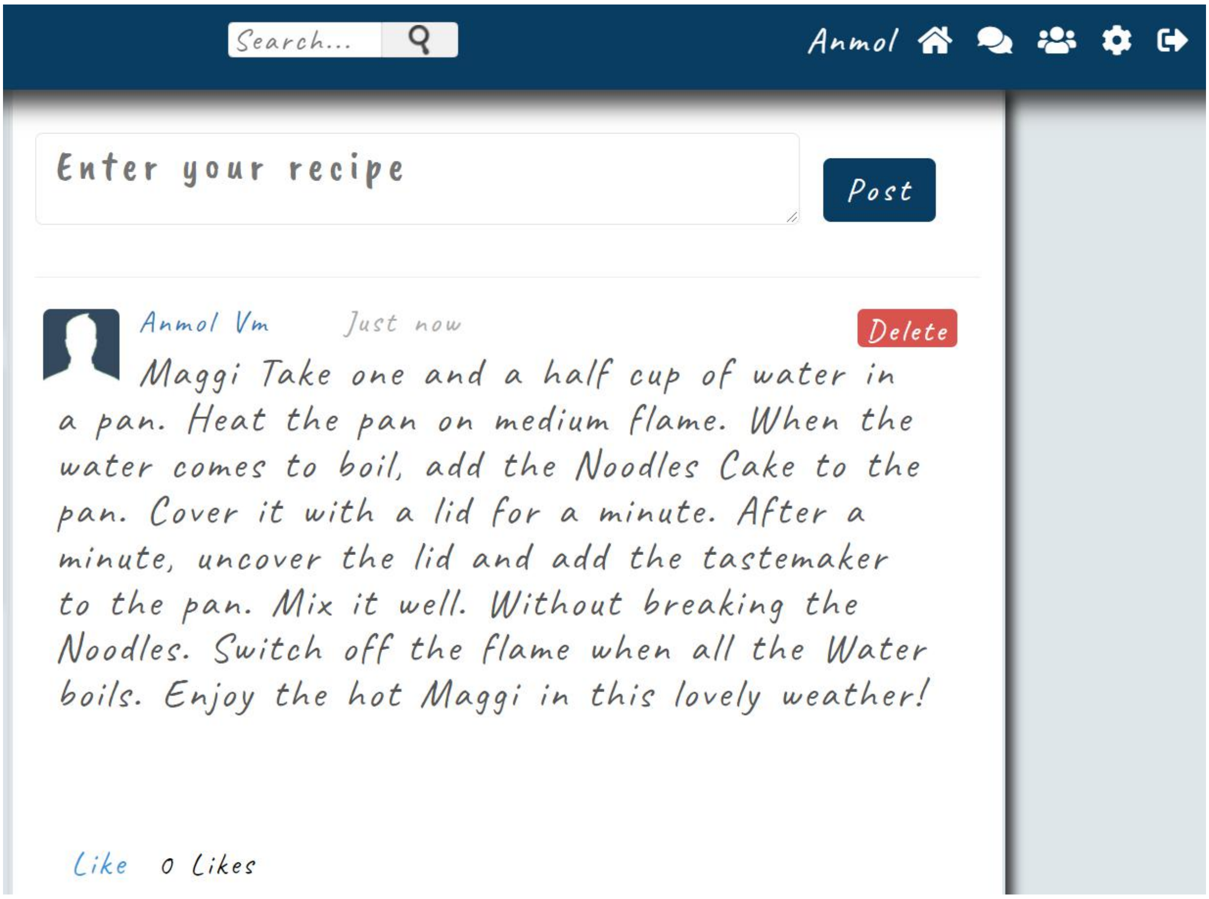
Confirm Email

Password

Confirm Password

Already have an account? Sign in here!

Postpage

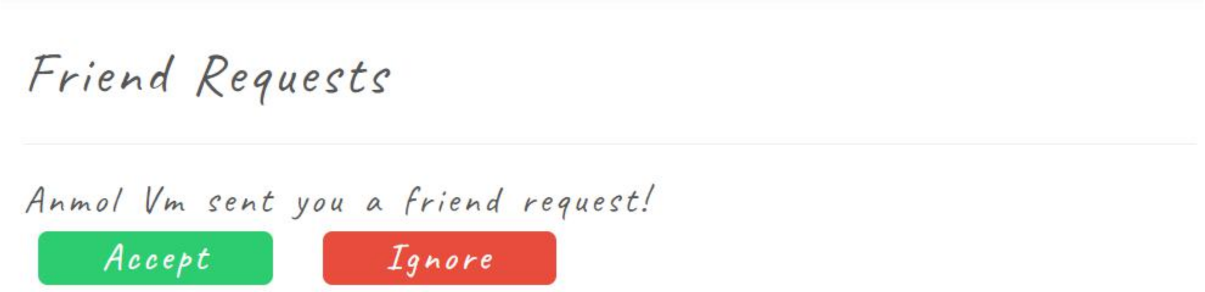


Anmol Vm Just now

Maggi Take one and a half cup of water in a pan. Heat the pan on medium flame. When the water comes to boil, add the Noodles Cake to the pan. Cover it with a lid for a minute. After a minute, uncover the lid and add the tastemaker to the pan. Mix it well. Without breaking the Noodles. Switch off the flame when all the Water boils. Enjoy the hot Maggi in this lovely weather!

0 Likes

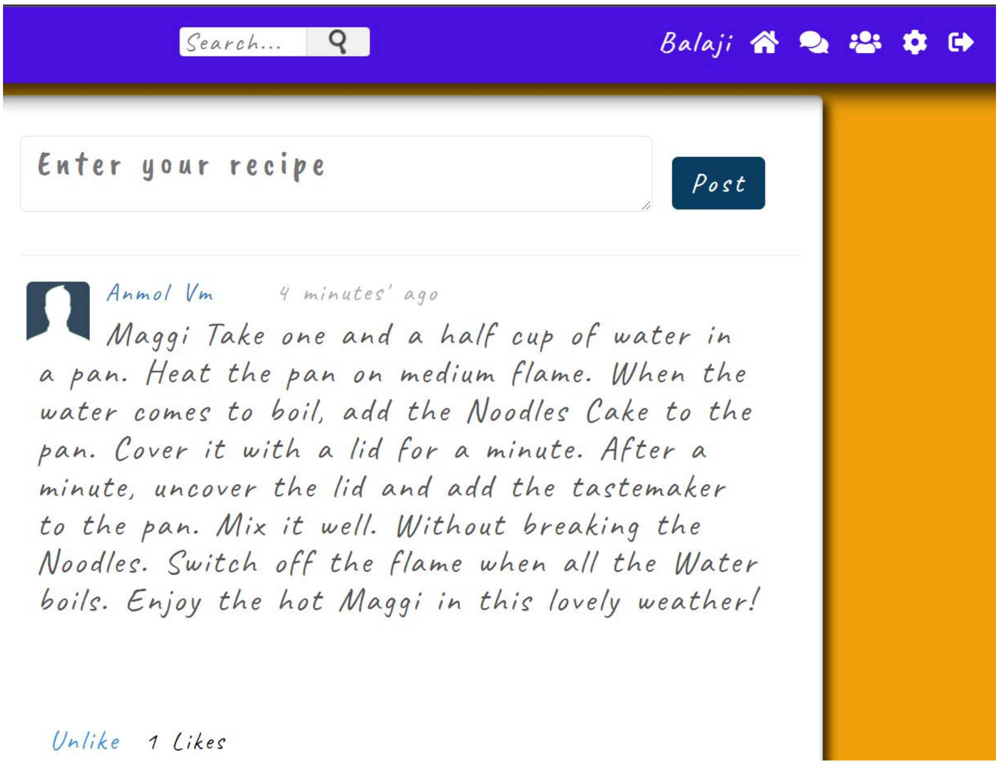
Send friend request



Friend Requests

Anmol Vm sent you a friend request!

Updation of posts based on friends

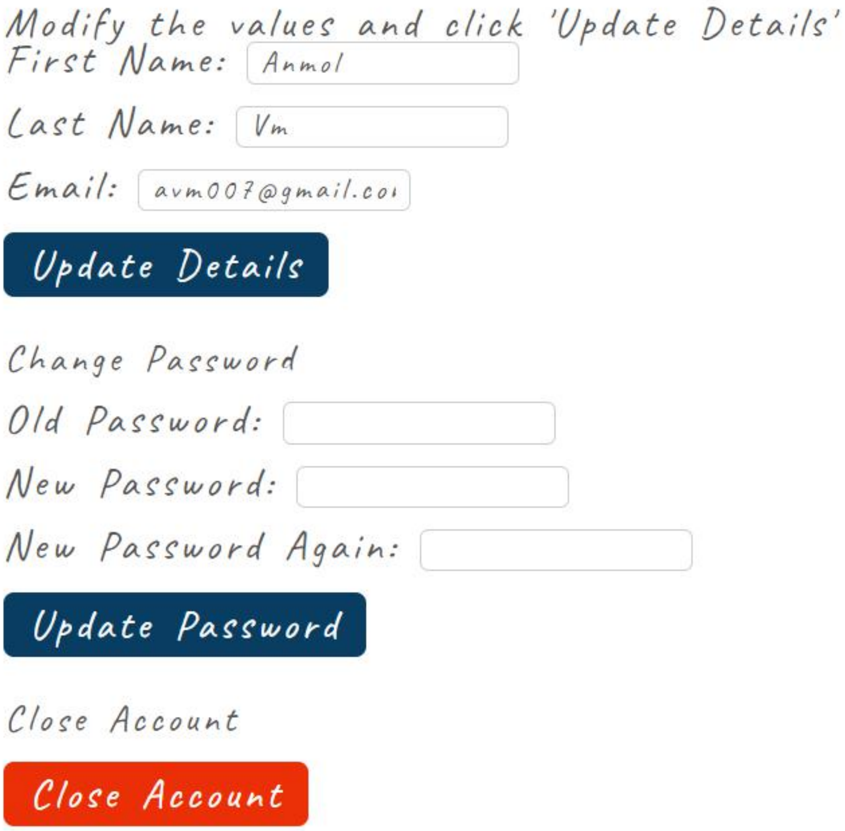


Anmol Vm 4 minutes' ago

Maggi Take one and a half cup of water in a pan. Heat the pan on medium flame. When the water comes to boil, add the Noodles Cake to the pan. Cover it with a lid for a minute. After a minute, uncover the lid and add the tastemaker to the pan. Mix it well. Without breaking the Noodles. Switch off the flame when all the Water boils. Enjoy the hot Maggi in this lovely weather!

Unlike 1 Likes

settings



Modify the values and click 'Update Details'

First Name: Anmol

Last Name: Vm

Email: avm007@gmail.co

Change Password

Old Password:

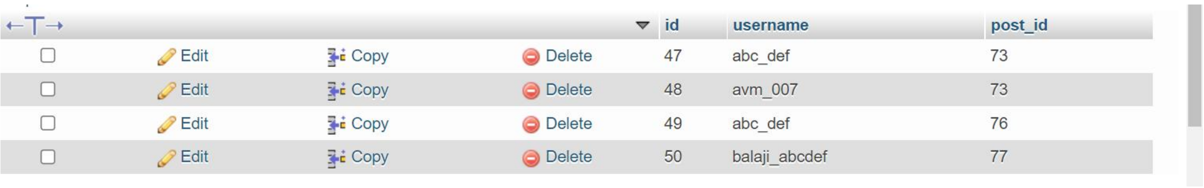
New Password:

New Password Again:

Close Account

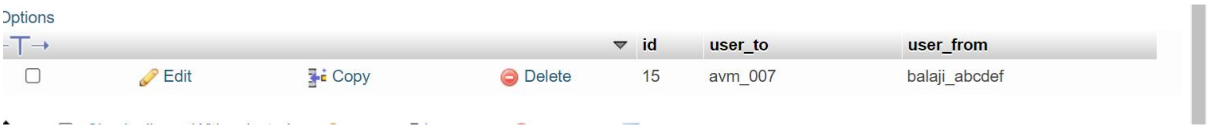
User_likes

db



←T→			▼	id	username	post_id
□	Edit	Copy	Delete	47	abc_def	73
□	Edit	Copy	Delete	48	avm_007	73
□	Edit	Copy	Delete	49	abc_def	76
□	Edit	Copy	Delete	50	balaji_abcdef	77

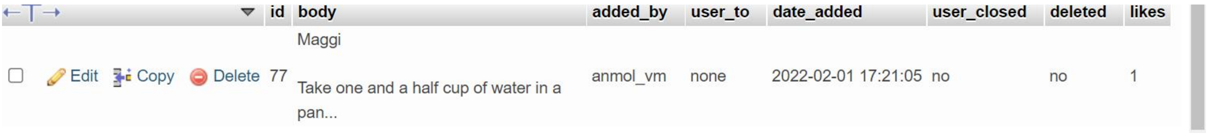
Friend_request_db



Options

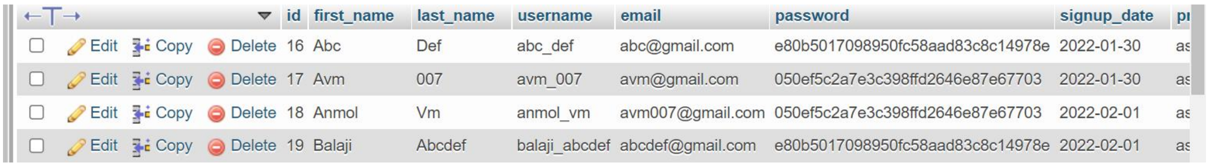
←T→			▼	id	user_to	user_from
□	Edit	Copy	Delete	15	avm_007	balaji_abcdef

Post_db



←T→			▼	id	body	added_by	user_to	date_added	user_closed	deleted	likes
□	Edit	Copy	Delete	77	Maggi Take one and a half cup of water in a pan...	anmol_vm	none	2022-02-01 17:21:05	no	no	1

User_db



←T→			▼	id	first_name	last_name	username	email	password	signup_date	pı
□	Edit	Copy	Delete	16	Abc	Def	abc_def	abc@gmail.com	e80b5017098950fc58aad83c8c14978e	2022-01-30	as
□	Edit	Copy	Delete	17	Avm	007	avm_007	avm@gmail.com	050ef5c2a7e3c398ffd2646e87e67703	2022-01-30	as
□	Edit	Copy	Delete	18	Anmol	Vm	anmol_vm	avm007@gmail.com	050ef5c2a7e3c398ffd2646e87e67703	2022-02-01	as
□	Edit	Copy	Delete	19	Balaji	Abcdef	balaji_abcdef	abcdef@gmail.com	e80b5017098950fc58aad83c8c14978e	2022-02-01	as

CONCLUSION

This project enables the users to learn about different recipes of different cuisines.

The users can also explore various food coming from different regions in a country.

This gathers a variety of food recipes from all over the world to appreciate the culinary culture spread across various regions.

This project enables the users to share their own recipes to the world.

REFERENCES

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill

3. https://www.w3schools.com/TAGS/default.ASP

4. https://www.w3schools.com/cssref/

5. https://developer.mozilla.org/en-US/docs/Web/JavaScript

6. https://api.jquery.com/category/ajax/

7. https://dev.mysql.com/doc/