

MESS MANAGEMENT SYSTEM

DATABASE SYSTEMS

ADITI GUPTA CS22B2048

SHRINIDHI CS22B1100

SRIHITHA PULAPA CS22B2009

LAKSHITHA CS22B1006

DHANUSH CS22B2017

May 17, 2024

1 Introduction

The purpose of this document is articulate the detailed methodology, implementation, and technology stack for the development of the Mess Management System within the mess Akshaya of our institute, promoting efficient resource utilization and enhancing the overall dining experience for students.

2 Methodology

2.1 Created Database

Using phpMyAdmin, we setup the database 'messydb', added necessary tables. (`admlogin`, `feedback`, `infra`, `menu`, `recipe`, `stulogin`, `supply`, `wastage`, and `workers`), and populated them with initial data. Constraints such as primary keys and foreign keys were implemented for data integrity. Lastly, we created a `config.php` file to establish a connection with the database using defined constants for the server, username, password, and database name

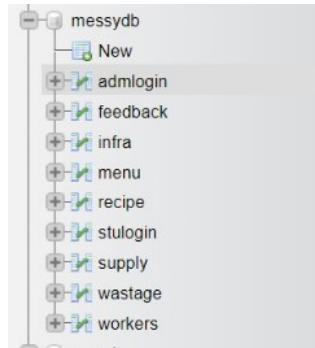


Fig. 1: Database Structure

- admlogin: Stores administrator login details.
- feedback: Collects feedback from students.
- infra: Manages infrastructure details.
- menu: Contains the daily or weekly mess menu.
- recipe: Lists recipes used in the mess.
- stulogin: Stores student login details.
- supply: Manages supply inventory.
- wastage: Tracks food wastage.
- workers: Contains details of workers in the mess.

2.1.1 admlogin

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	username	varchar(20)	utf8mb4_general_ci		No	None			
2	password	varchar(20)	utf8mb4_general_ci		No	None			

Fig. 2: Table for admin login

2.1.2 admlogin

The screenshot shows a MySQL query results page. At the top, a green bar indicates "Showing rows 0 - 1 (2 total. Query took 0.0021 seconds.)". Below this is the SQL query: "SELECT * FROM `admlogin`". Underneath the query, there are several buttons: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. A search bar at the bottom allows filtering by table name. The main content area displays the data from the 'admlogin' table:

username	password
sriram	sriramc2004
vikram	vikramsc0ok1990

Fig. 3: Populated admin login

2.1.3 feedback feedback

The screenshot shows the phpMyAdmin interface for the 'messydb' database. The left sidebar lists various databases and tables. The current table is 'feedback'. The top navigation bar includes tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. The 'Structure' tab is selected, showing the table's columns:

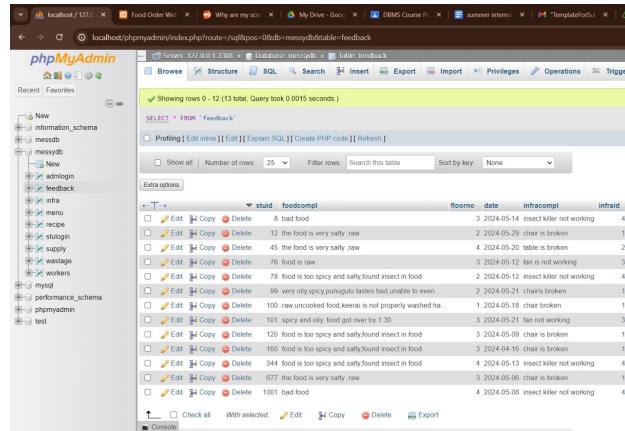
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	stud	int(11)			No	None			
2	feedcompl	varchar(300) utf8mb4_general_ci			No	None			
3	feemo	int(11)			No	None			
4	date	date			No	None			
5	infracompl	varchar(300) utf8mb4_general_ci			No	None			
6	infrad	int(11)			No	None			

Below the columns, there are buttons for Print, Propose table structure, Move columns, Normalize, Add, and Go. The 'Indexes' section shows two indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
	PRIMARY	BTREE	Yes	No	stud	13	A	No	
	rfeed	BTREE	No	No	infrad	13	A	No	

At the bottom, there is a button for Create an index on [] columns and a Go button.

Fig. 4: Feedback table

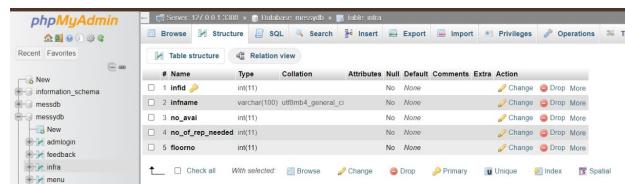


The screenshot shows the phpMyAdmin interface for the 'messydb' database. The left sidebar lists databases like information_schema, messydb, and mysql. The 'feedback' table is selected in the center. The table has columns: id, stuid, foodcompl, floorno, date, infracompl, and infraid. The data grid contains 12 rows of feedback entries, such as '8 bad food' and '12 the food is very salty raw'. Below the table are standard MySQL management options: Edit, Copy, Delete, etc.

Fig. 5: Populated feedback table

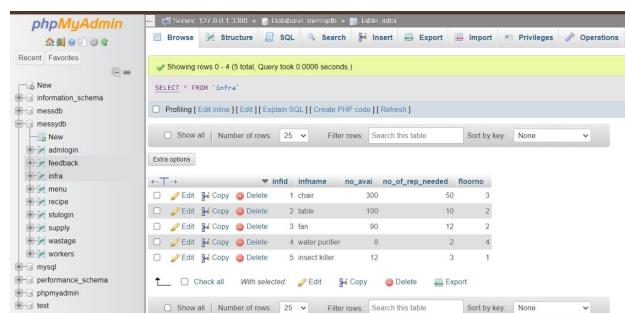
2.1.4 infra

infra



The screenshot shows the phpMyAdmin interface for the 'messydb' database. The left sidebar lists databases like information_schema, messydb, and mysql. The 'infra' table is selected in the center. The table structure is shown in the 'Structure' tab, with columns: id (int(11)), infname (varchar(100) utf8mb4_general_ci), no_avail (int(11)), no_of_rep_needed (int(11)), and floorno (int(11)). The 'Operations' tab shows icons for Edit, Copy, Delete, Change, Drop, and More.

Fig. 6: Infra table



The screenshot shows the phpMyAdmin interface for the 'messydb' database. The left sidebar lists databases like information_schema, messydb, and mysql. The 'infra' table is selected in the center. The table has columns: id (int(11)), infname (varchar(100) utf8mb4_general_ci), no_avail (int(11)), no_of_rep_needed (int(11)), and floorno (int(11)). The data grid contains 5 rows of infrastructure items, such as '1 char' and '2 table'. Below the table are standard MySQL management options: Edit, Copy, Delete, etc.

Fig. 7: Populated Infra Table

2.1.5 menu

menu

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	dinner	longtext	utf8mb4_bin		No	None		Change Drop More	Change Drop More
2	lunch	longtext	utf8mb4_bin		No	None		Change Drop More	Change Drop More
3	bfast	longtext	utf8mb4_bin		No	None		Change Drop More	Change Drop More
4	recip_id	longtext	utf8mb4_bin		No	None		Change Drop More	Change Drop More
5	day	varchar(40)	utf8mb4_general_ci		No	None		Change Drop More	Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	day	2	A	No	

Create an index on columns

Fig. 8: Menu Table

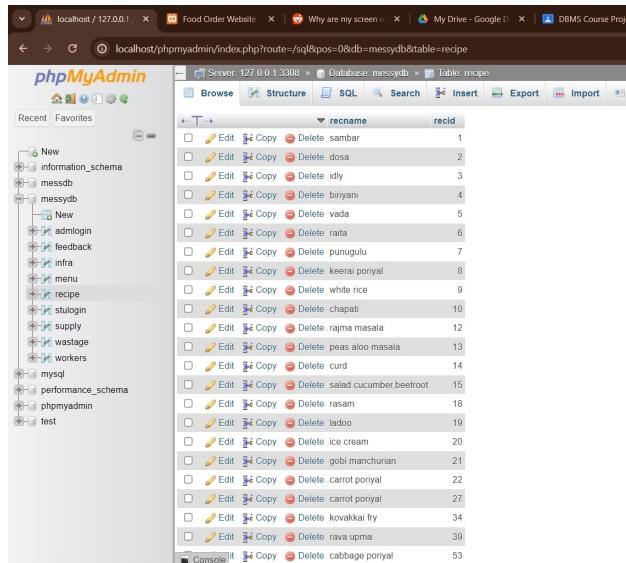
2.1.6 recipe

recipe

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	renename	varchar(100)	utf8mb4_general_ci		No	None		Change Drop More	Change Drop More
2	recipid	int(11)			No	None		Change Drop More	Change Drop More

Indexes

Fig. 9: Recipe Table



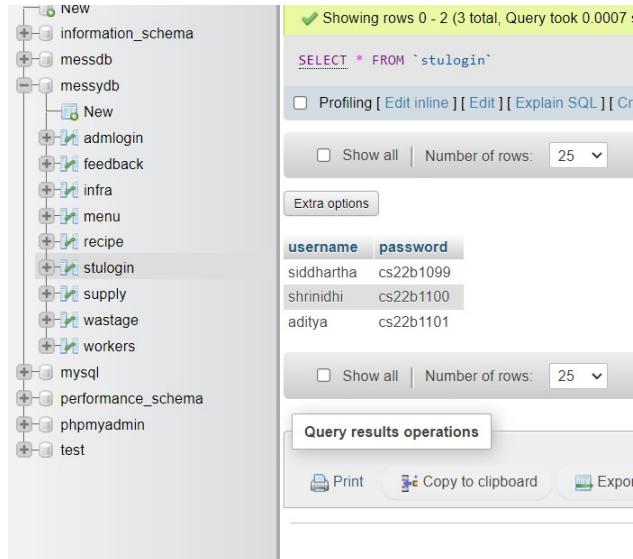
The screenshot shows the phpMyAdmin interface for the 'messydb' database. The left sidebar lists various databases and tables. The 'recipe' table is selected, displaying 53 rows of data. The columns are 'recname' and 'recid'. The data includes various Indian dishes like sambar, dosa, idly, biryani, vada, raita, punugulu, keerai poryal, white rice, chapati, rajma masala, peas aloo masala, curd, salad cucumber beetroot, rasam, ladoo, ice cream, gobi manchurian, carrot poryal, carrot poryal, kovakka fry, rava upma, and cabbage poryal.

	recname	recid
	sambhar	1
	dosa	2
	idly	3
	biryani	4
	vada	5
	raita	6
	punugulu	7
	keerai poryal	8
	white rice	9
	chapati	10
	rajma masala	12
	peas aloo masala	13
	curd	14
	salad cucumber beetroot	15
	rasam	18
	ladoo	19
	ice cream	20
	gobi manchurian	21
	carrot poryal	22
	carrot poryal	27
	kovakka fry	34
	rava upma	39
	cabbage poryal	53

Fig. 10: Populated Recipe Table

2.1.7 stulogin

stulogin



The screenshot shows the phpMyAdmin interface for the 'messydb' database. The left sidebar lists various databases and tables. The 'stulogin' table is selected, displaying 3 rows of data. The columns are 'username' and 'password'. The data includes siddhartha with password cs22b1099, shrinidhi with password cs22b1100, and aditya with password cs22b1101.

username	password
siddhartha	cs22b1099
shrinidhi	cs22b1100
aditya	cs22b1101

Fig. 11: Student Login Table

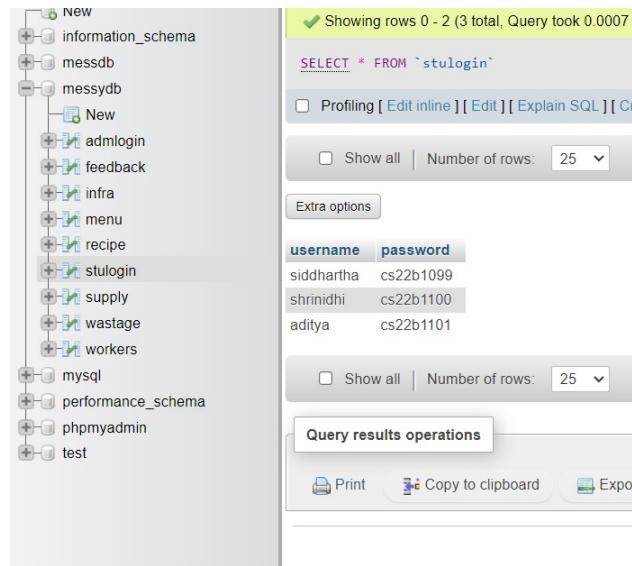


Fig. 12: Populated Student Login

2.1.8 supply

supply

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	supname	varchar(100)	utf8mb4_general_ci		No	None		Change	Drop More
2	cost	int(11)			No	None		Change	Drop More
3	dateoff	date			No	None		Change	Drop More
4	quantity	int(11)			No	None		Change	Drop More
5	Itemid	int(11)			No	None		Change	Drop More
6	Itemname	varchar(100)	utf8mb4_general_ci		No	None		Change	Drop More
7	supervisorid	int(11)			No	None		Change	Drop More

Fig. 13: Supply Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	supname	varchar(100)	utf8mb4_general_ci		No	None		Change	Drop More
2	cost	int(11)			No	None		Change	Drop More
3	dateoff	date			No	None		Change	Drop More
4	quantity	int(11)			No	None		Change	Drop More
5	Itemid	int(11)			No	None		Change	Drop More
6	Itemname	varchar(100)	utf8mb4_general_ci		No	None		Change	Drop More
7	supervisorid	int(11)			No	None		Change	Drop More

Fig. 14: Populated Supply Table

2.1.9 wastage

wastage

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
1	floorno	int(11)			No	None				
2	amountwaste	int(11)			No	None				
3	recipwasted	int(11)			No	None				
4	wastedate	date			No	None				

Fig. 15: Wastage Table

		floorno	amountwaste	recipwasted	wastedate
<input type="checkbox"/>		1	10	4	2024-05-28
<input type="checkbox"/>		3	68	5	2024-03-14
<input type="checkbox"/>		4	6	7	2024-04-30
<input type="checkbox"/>		1	34	7	2024-05-20
<input type="checkbox"/>		3	25	8	2024-05-03
<input type="checkbox"/>		4	20	10	2024-01-10
<input type="checkbox"/>		3	45	10	2024-05-05
<input type="checkbox"/>		4	12	14	2024-05-12
<input type="checkbox"/>		4	67	20	2024-05-21

Fig. 16: Populated Wastage Table

2.1.10 workers

workers

		lname	fname	empid	salary	desig
<input type="checkbox"/>		kumar	suresh	1005	12000	dosa master
<input type="checkbox"/>		reddy	suchith	1007	18000	head cook
<input type="checkbox"/>		raj	vel	1039	10000	cook
<input type="checkbox"/>		mouni	chandramouli	1200	40000	manager

Fig. 17: Workers

2.2 Created Partials/Repeating Parts

Description: For the repetitive parts like Menu and Footer, we modularized these sections into separate PHP files. This allowed us to include them in multiple pages easily, ensuring consistency and reducing redundancy.

2.2.1 menu.php

This file contains the HTML for the navigation menu of the website. The menu is enclosed in a `div` with a class of `menu text-center`, and it includes links to various sections of the website such as Home, Student Login, Admin, View Mess Menu, Update Mess Menu, About, and Contact Us. The CSS stylesheet is linked for styling purposes.

2.2.2 footer.php

This file contains the HTML for the footer of the website. The footer is enclosed in a `div` with a class of `footer` and includes two paragraphs. The first paragraph displays the copyright information, and the second one credits the development team with a link to their GitHub repository.

2.3 Add Admin to Database

Description: This implementation includes adding new admins and displaying all existing admins from the database. The `config/constants.php` file is included to establish a database connection. The HTML structure provides a login interface for admins, styled using `homestyle.css`, and includes a form for username and password input. Upon form submission, the provided credentials are checked against the `admlogin` table in the database. If the credentials match, the user is redirected to the dashboard.

SQL implementation:

```
if(isset($_POST['submit'])) {
    $username = $_POST['username'];
    $password = md5($_POST['password']);
    $sql = "SELECT * FROM admlogin WHERE username='$username' AND password='$password'";
    $res = mysqli_query($connection, $sql);
    $count = mysqli_num_rows($res);
```

2.4 Admin Login

The `admin-login.php` script facilitates the login process for administrators of the Mess Management system. It includes a form for admins to enter their username and password, which are then processed via a POST request. The provided password is hashed using MD5 before being compared against stored credentials in the `admlogin` table of the database. If a match is found, the admin is redirected to the dashboard. Otherwise, an error handling mechanism redirects the admin back to the login page.

SQL Implementation :

```
?php
if(isset($_POST['submit'])) {
    $username = $_POST['username'];
    $password = md5($_POST['password']);

    $sql = "SELECT * FROM admlogin WHERE username='$username' AND password='$password'";
    $res = mysqli_query($connection,$sql);
```

```

$count = mysqli_num_rows($res);

if($count == 1) {

    header('location:' . SITEURL . 'dashboard.php/');
}
else
header('location:' . SITEURL . 'dashboard.php/');
}
?;

```

2.5 Student Login

The student-login.php page is the gateway for students to access the Mess Management system. It presents a user-friendly login interface where students can input their credentials for authentication. Upon submission, the entered username and password are validated against the database.

Once logged in, students are redirected to their respective dashboards, where they can access features tailored to their needs. In the event of invalid credentials, students remain on the login page, and appropriate feedback mechanisms can be implemented to notify them of any errors.

The login process is secure, with passwords hashed using the MD5 algorithm before comparison. This ensures data security and integrity throughout the authentication process. Overall, the student login page provides a seamless and secure entry point for students into the Mess Management system.

SQL Implementation

```

$sql = "SELECT * FROM stulogin WHERE username='$_username' AND pass-
word='$_password'";
myalert('$_sql');
$res = mysqli_query($connection, $sql);

```

Along with the provided features, admin can also change password using email authentication. Also to increase the security, password protected login is required.

2.6 Give Feedback

The ‘givefeedback.php’ script allows students to submit feedback regarding food and infrastructure issues. It includes a form where students can input their student ID, food complaint, floor number, infrastructure complaint, infrastructure ID, and date. Upon form submission, the provided feedback is processed and inserted into the ‘feedback’ table in the database. If the feedback is successfully submitted, a success message is displayed and the user is redirected to the student dashboard. In case of failure, an error message is shown and the user remains on the feedback page. The form is styled for ease of use and includes options to submit or reset the form.

SQL Implementation

```

if(isset($_POST['submit'])) {
// Get the Data from form
$stuid = $_POST['stuid'];

```

```

$foodcompl = $_POST['foodcompl'];
$floorno = $_POST['floorno'];
$infracompl = $_POST['infracompl'];
$infraid = $_POST['infraid'];
$date = $_POST['date'];
// SQL Query to Save the data into feedback table
$sql = "INSERT INTO feedback SET stuid='$_POST[stuid]', foodcompl='$_POST[foodcompl]', floorno='$_POST[floorno]', infra-compl='$_POST[infracompl]', infraid='$_POST[infraid]', date='$_POST[date']";
// Executing Query and Saving Data into Database
$res = mysqli_query($connection, $sql) or die(mysqli_error($connection));

```

2.7 View Feedback

The viewfeedback.php script enables administrators to view feedback submitted by users regarding the Mess Management system. It presents a table displaying various types of feedback, including food complaints, floor numbers, infrastructure complaints, infrastructure IDs, and dates. The script retrieves feedback data from the database and populates the table accordingly. If feedback exists, it is displayed row by row. In case no feedback is found, an appropriate message is shown. The script ensures a well-organized presentation of feedback data for effective monitoring and management.

SQL Implementation

```

$sql = "SELECT * FROM feedback";
$res = mysqli_query($connection, $sql);

```

2.8 View Menu

The viewmenu.php script lets admins see the menu details for each day. It displays a table with information like breakfast, lunch, dinner, and recipe IDs, taken from the database. Admins can easily update or delete menu entries using the provided buttons. Clicking 'Update' directs them to updatemenu.php to make changes, while 'Delete' removes the entry via deletemenu.php. Feedback messages keep admins informed about any changes made to the menu, ensuring smooth management.

SQL Implementation

```

$sql = "SELECT * FROM menu";
$res = mysqli_query($connection, $sql);

```

2.9 Update Menu

The 'updatemenu.php' script allows admins to update the daily menu details for the Mess Management system. When a specific day is selected, the script retrieves the existing menu information from the database for that day, including breakfast, lunch, dinner, and recipe IDs. This data is displayed in a form where it can be edited. Upon submission, the updated values are sent to the server, which processes the data and updates the corresponding menu entry in the database. Successful updates result in a confirmation message, while any errors during the process are communicated back to the user. The script includes feedback and session handling to ensure a smooth user experience.

SQL Implementation

```

$sql = "SELECT * FROM menu WHERE day='\$day'";
$sql = "UPDATE menu SET day='\$new_day', bfast='\$breakfast', lunch='\$lunch', dinner='\$dinner', recip_id='\$recipe_id' WHERE day='\$day'";

// Execute the Query
$res = mysqli_query($connection, $sql);

```

2.10 Delete Menu

The deletemenu.php script allows administrators to remove menu entries from the Mess Management system. It verifies the provided day value, deletes the corresponding menu entry from the database, and provides feedback on the success or failure of the deletion operation. Proper redirection and session handling ensure a smooth user experience.

SQL Implementation

```
$sql = "DELETE FROM menu WHERE day='\$day'";
```

2.11 View Daily Menu

The viewdailymenu.php script provides administrators with a convenient way to view the daily menu details for the Mess Management system. It presents a user-friendly interface displaying a dropdown menu to select a specific day of the week. Upon selecting a day and submitting the form, the script retrieves and displays the corresponding menu details from the database. After selecting a day and submitting the form by clicking the "Submit" button, the script processes the request. It executes an SQL query to fetch the daily menu details from the database based on the selected day.

SQL Implementation

```

if(isset($_POST['submit'])) { $day = $_POST['day'];
// Query to Get Daily Menu from Database based on selected day
$sql = "SELECT * FROM menu WHERE day='\$day'";
$res = mysqli_query($connection, $sql);

```

2.12 Add Supply

The 'addsupply.php' script allows administrators to add new supplies to the database. It includes a form where users can input supply details such as name, cost, date of arrival, quantity, item ID, item name, and supervisor ID. Upon submission, the form data is processed and inserted into the 'supply' table in the database. If the data is successfully inserted, a success message is displayed and the user is redirected to the supply management page. In case of an error, a failure message is shown and the user remains on the add supply page. The form and messages are styled for a clear and user-friendly interface.

SQL Implementation

```
$sql = "INSERT INTO supply SET supname='\$supname', cost=$cost, dateofarr='\$dateofarr', quantity=
\$quantity, itemid=$itemid, itemname='\$itemname', supervisorid=$supervisorid ";
```

2.13 Delete Supply

The 'deletesupply.php' script handles the deletion of supply records from the database. When an 'itemid' parameter is provided via GET, it executes a SQL DELETE query to remove the corresponding supply

entry from the ‘supply’ table. The script checks if the query executes successfully, and sets a session message indicating whether the supply was successfully deleted or if the deletion failed. It then redirects the user to the ‘viewsupply.php’ page to display the updated supply list along with the relevant message. If no ‘itemid’ is provided, it redirects directly to the ‘viewsupply.php’ page.

SQL Implementation

```
$sql = "DELETE FROM supply WHERE itemid=$itemid";
```

2.14 Update Supply

The update-supply.php script facilitates the modification of supply details within the Mess Management system. Administrators can access this functionality to update information such as supply name, cost, date of arrival, quantity, item name, and supervisor ID.

Upon accessing the update supply page, the script retrieves the supply details based on the provided item ID. If the item ID is valid and corresponds to an existing supply entry, the script populates the form fields with the current supply details. Otherwise, an error message is displayed, indicating that the supply was not found.

Administrators can then make modifications to the supply details directly within the form. Upon submission, the updated values are sent to the server for processing. The script executes an SQL query to update the supply details in the database. If the update is successful, a success message is displayed, informing the user that the supply has been updated. Conversely, if the update fails, an error message is displayed, indicating that the update was unsuccessful.

SQL Implementation

```
if(isset($_POST['submit'])) {
    // Get all the values from form to update
    $itemid = $_POST['itemid'];
    $supname = $_POST['supname'];
    $cost = $_POST['cost'];
    $dateofarr = $_POST['dateofarr'];
    $quantity = $_POST['quantity'];
    $itemname = $_POST['itemname'];
    $supervisorid = $_POST['supervisorid'];

    // SQL Query to Update the supply
    $sql = "UPDATE supply SET supname='$supname', cost=$cost, dateofarr='$dateofarr', quantity=$quantity, itemname='$itemname', supervisorid=$supervisorid WHERE itemid=$itemid";
```

2.15 View Employee Details

The view-employee.php script provides administrators with the capability to view details of employees registered within the Mess Management system. It presents a table showcasing information such as employee last name, first name, employee ID, salary, and designation. The script retrieves this data from the database

and populates the table accordingly.

Administrators can take actions on employee records directly from this view, as the script includes options to update or delete each employee entry. The 'Update' button redirects administrators to the updateemp.php page for modifying employee details, while the 'Delete' button initiates the deletion process through the deleteemployee.php script.

SQL Implementation

```
$sql = "SELECT * FROM workers";
$res = mysqli_query($connection, $sql);
```

2.16 Add Employee

The addemployee.php script allows administrators to onboard new workers and maintain a comprehensive database of employee information.

Upon accessing the page, administrators are presented with a form containing fields for the worker's last name, first name, employee ID, salary, and designation. These fields enable administrators to input essential details regarding the new employee.

Administrators can input the worker's last name, first name, unique employee ID, salary, and designation into the corresponding fields. These details help in accurately identifying and categorizing employees within the system.

After entering the necessary information, administrators can submit the form by clicking the "Add employee" button. Upon submission, the data entered into the form is processed by the script.

The script executes an SQL query to insert the employee details into the database. If the insertion is successful, a success message is displayed, confirming that the employee has been added successfully. Subsequently, administrators are redirected to the viewemployee.php page to review the updated list of employees.

SQL Implementation

```
$sql = "INSERT INTO workers SET lname='\$lname', fname='\$fname', empid='\$empid', salary='\$salary',
desig='\$desig'
";
```

2.17 Update Employee

The update-employee.php script facilitates the modification of employee details within the Mess Management system. Administrators can access this functionality to update specific employee records. Upon loading the page, the script retrieves the existing employee details based on the provided employee ID (empid) from the database.

A form is presented with fields for the worker's last name, first name, employee ID, salary, and designation. The form is pre-filled with the current employee information, allowing for easy editing. The employee ID field is set to read-only to prevent inadvertent changes to this crucial identifier.

Administrators can make necessary adjustments to the employee's details directly within the form. Upon submission, the updated values are processed and sent to the server. The script executes an SQL query to update the corresponding employee entry in the database.

Feedback messages are provided to inform administrators about the outcome of the update operation. A success message indicates that the employee details were successfully updated, while an error message

notifies administrators if the update process encountered any issues..

SQL Implementation

```
// Get the details
$row = mysqli_fetch_assoc($res);

$lname = $row['lname'];
$fname = $row['fname'];
$empid = $row['empid'];
$salary = $row['salary'];
$desig = $row['desig'];

// SQL Query to Update the employee
$sql = "UPDATE workers SET
lname='$lname', fname='$fname', empid='$empid', salary='$salary', desig='$desig' WHERE empid='$original_empid'" ;
```

2.18 Delete Employee

The ‘delete-employee.php’ script allows administrators to delete an employee’s record from the database. When the script receives an ‘empid’ parameter via GET, it executes a SQL DELETE query to remove the corresponding record from the ‘workers’ table. The script checks if the query executes successfully, and based on the outcome, it sets a session message indicating success or failure. Finally, it redirects the user to the ‘viewemployee.php’ page to display the updated employee list along with the relevant message. If no ‘empid’ is provided, it redirects directly to the ‘viewemployee.php’ page.

SQL Implementation \$sql = "DELETE FROM workers WHERE empid='\$empid'" ;

2.19 Add Wastage Details

The addwaste.php script allows administrators to accurately track and manage wastage, ensuring efficient resource utilization.

Upon accessing the page, administrators are presented with a form containing fields for the floor number, amount of waste, recipe ID of the wasted item, and the date of wastage. These fields enable administrators to specify relevant details regarding the wastage event.

Administrators can input the floor number where the wastage occurred, the quantity of the wasted item, the corresponding recipe ID (if applicable), and the date on which the wastage occurred.

After entering the necessary information, administrators can submit the form by clicking the ”Add wastage of item” button. Upon submission, the data entered into the form is processed by the script. The script executes an SQL query to insert the wastage details into the database.

SQL Implementation if(isset(\$_POST['submit'])) {

// Button Clicked

//1. Get the Data from form

\$floorno = \$_POST['floorno'];

\$amountwaste= \$_POST['amountwaste'];

\$recipwasteid = \$_POST['recipwasteid'];

```
$datewaste = $_POST['wastedate'];

//2. SQL Query to Save the data into database
$sql = "INSERT INTO wastage SET
floorno=$floorno, amountwaste='$amountwaste', recipwasteid='$recipwasteid', wastedate='$datewaste'
";

//3. Executing Query and Saving Data into Database
$res = mysqli_query($connection, $sql) or die(mysqli_error($connection));
```

2.20 View Wastage Details

The viewwaste.php script allows administrators to monitor and manage wastage details within the Mess Management system. Upon accessing this feature, administrators are presented with a table displaying information regarding wastage, including the floor number, amount of wastage, recipe ID of the wasted item, and the date of wastage.

The script fetches data from the 'wastage' table in the database and dynamically populates the table rows accordingly. Administrators can easily identify and track instances of wastage through this organized presentation of data.

If there is data available in the database, the script displays it row by row. However, if no wastage details are found, an appropriate message is shown to inform administrators accordingly.

SQL Implementation \$sql = "SELECT * FROM wastage";

3 Implementation

System Features are implemented as follows:

3.1 Home Page

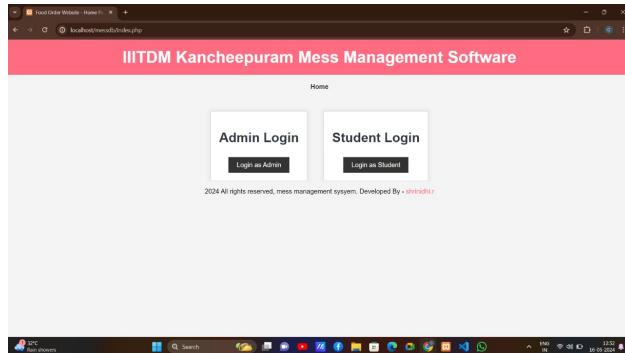


Fig. 18: Home Page

3.2 Login

1. Admin Login

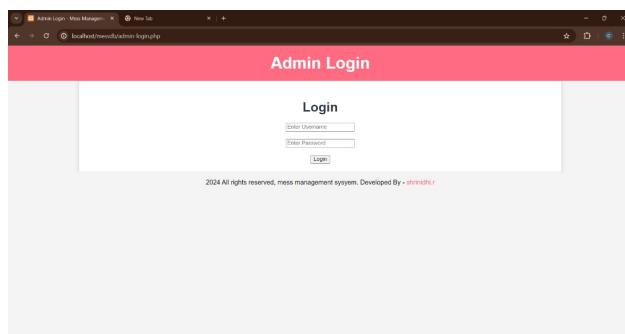


Fig. 19: Admin Login

2. Student Login

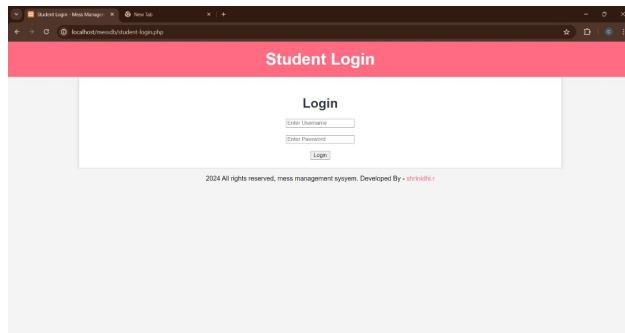


Fig. 20: Student Login

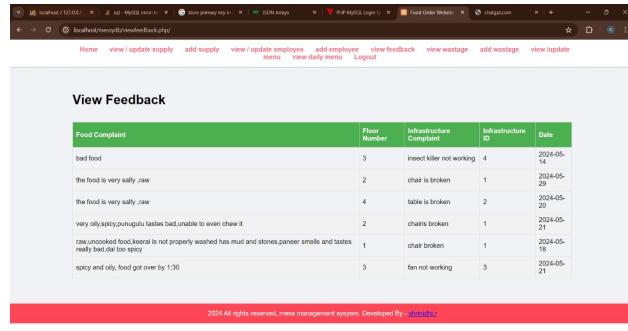
3.3 Logout

3.4 Give Feedback

A screenshot of a web browser window titled "Food Order Website - Home". The top navigation bar includes links for "Home", "add feedback", "view menu", "view daily menu", and "Logout". Below the navigation is a red header bar with the title "Give Feedback". The main content area contains several input fields: "Student ID" (with placeholder "Enter Your Student ID"), "Food Complaint" (with placeholder "Enter Food Complaint"), "Floor Number" (with placeholder "Enter Floor Number"), "Infrastructure Complaint" (with placeholder "Enter Infrastructure Complaint"), "Infrastructure ID" (with placeholder "Enter Infrastructure ID"), and a "Date" field showing "06-09-2024". At the bottom of the page is a red footer bar with the copyright notice "2024 All rights reserved, mess management system. Developed By - shrishruti".

Fig. 21: Feedback

3.5 View Feedback



The screenshot shows a web browser window with a navigation bar at the top containing links for Home, view / update supply, add supply, view / update employee, add employee, view feedback, view wastage, add wastage, and view update. Below the navigation bar is a title 'View Feedback'. Underneath the title is a table with the following data:

Food Complaint	Floor Number	Infrastructure Complaint	Infrastructure ID	Date
bad food	3	insect killer not working	4	2024-05-14
the food is very salty , raw	2	chair is broken	1	2024-05-29
the food is very salty , raw	4	table is broken	2	2024-05-20
very oily spicy,pungent taste bad,unable to even chew it	2	chairs broken	1	2024-05-21
raw,uncooked food,keenai is not properly washed has mud and stones,paneer smells and tastes really bad,etc too spicy	1	chair broken	1	2024-05-18
spicy and oily, food got over by 1:30	3	fan not working	3	2024-05-21

At the bottom of the page, there is a red footer bar with the text '2024 All rights reserved, mess management system. Developed By - [dinesh](#)'.

Fig. 22: View Feedback

3.6 View Menu



The screenshot shows a web browser window with a navigation bar at the top containing links for Home, view / update supply, add supply, view / update employee, add employee, view feedback, view wastage, add wastage, and view update. Below the navigation bar is a title 'View Menu'. Underneath the title is a table with the following data:

Day	Breakfast	Lunch	Dinner	Recipe IDs	Actions
monday	pongal, rava idli, neera	white rice, chapati, also peas masala, curd	white rice, chapati, rajma masala, curd	7, 9, 10, 12, 13, 14, 39	Update Delete
tuesday	idly, neva upma	white rice, curd, kovakkai fry, rasam, sambar, chapati	white rice, dosa, curd, also peas masala, rasam	1, 2, 3, 9, 10, 13, 14, 16, 34, 39	Update Delete
wednesday	dosa, vada	sambhar, rasam, white rice, chapati, curd, carrot poriyal	biryani, relish, ice cream, laddoo, gobi manchurian	1, 2, 4, 6, 5, 9, 10, 14, 18, 19, 20, 21, 22	Update Delete

At the bottom of the page, there is a red footer bar with the text '2024 All rights reserved, mess management system. Developed By - [dinesh](#)'.

Fig. 23: View Menu

3.7 Update Menu

The screenshot shows a web browser window with the title 'Update Menu'. The URL is 'localhost/messsys/updatemenu.php'. The page contains a form with four input fields labeled 'Day', 'Breakfast', 'Lunch', and 'Dinner'. Below these fields is a button labeled 'Update Menu'. At the bottom of the page, there is a red footer bar with the text '2024 All rights reserved, mess management system. Developed By: abhishek'.

Fig. 24: Update Menu

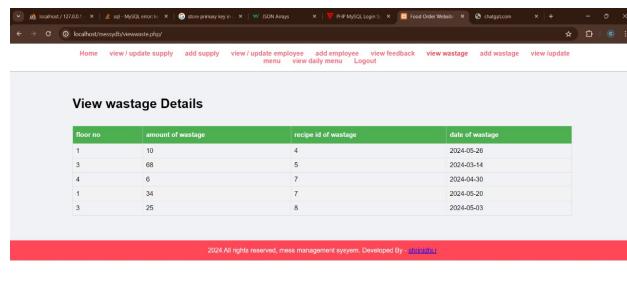
3.8 View Daily Menu

The screenshot shows a web browser window with the title 'View Daily Menu'. The URL is 'localhost/messsys/dailymenu.php'. The page displays a table with three columns: 'Day', 'Breakfast', and 'Dinner'. A dropdown menu for 'Select Day' is set to 'Monday'. The table shows data for Wednesday: Breakfast items include dosa, vada, sambar, rasam, white rice, chapati, curd, and carrot poriyal; Dinner items include biryani, raita, ice cream, laddo, and gobhi manchurian.

Select Day:	Monday	(Submit)	
Day	Breakfast	Lunch	Dinner
wednesday	dosa, vada sambar, rasam, white rice, chapati, curd, carrot poriyal		biryani, raita, ice cream, laddo, gobhi manchurian

Fig. 25: View Daily Menu

3.9 View Wastage Details

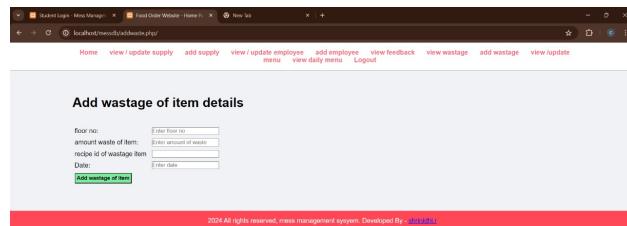


The screenshot shows a table titled "View wastage Details" with the following data:

floor no	amount of wastage	recipe id of wastage	date of wastage
1	10	4	2024-05-26
3	68	5	2024-03-14
4	6	7	2024-04-30
1	34	7	2024-05-20
3	25	8	2024-05-03

Fig. 26: View Wastage Details

3.10 Add Wastage Details



The screenshot shows a form titled "Add wastage of item details" with the following fields:

- floor no:
- amount waste of item:
- recipe id of wastage item:
- Date:

Add wastage of item

Fig. 27: Add Wastage Details

3.11 Add Employee Details

Add employee details

worker lastName:	<input type="text"/>
worker firstName:	<input type="text"/>
employee id:	<input type="text"/>
Salary:	<input type="text"/>
Designation:	<input type="text"/>
Add employee	

2024 All rights reserved, mess management system. Developed By: [abcde1234](#)

Fig. 28: Add Employee Details

3.12 View Employee Details

View employee Details

worker LastName	worker firstName	employee id	Salary	Designation	Actions
raj	naveen	77	12000	parotta master	Update Delete
chandramouli	anikam	1002	30000	manager	Update Delete
raj	vel	1003	6000	cleaner	Update Delete
shetty	vikram	1004	10000	cook	Update Delete
kumar	suresh	1005	12000	dosa master	Update Delete
reddy	sudhith	1007	18000	head cook	Update Delete
raj	subbu	1008	8000	cleaner	Update Delete

2024 All rights reserved, mess management system. Developed By: [abcde1234](#)

Fig. 29: View Employee Details

3.13 Update Employee Details

Update employee

worker lastName:

worker firstName:

empId:

salary:

designation:

Update employee

2024 All rights reserved, mess management system. Developed By - abraadu

Fig. 30: Update employee

3.14 Delete Employee from Database

3.15 View Supply Details

View Supply Details

Supply Name	Cost	Date of Arrival	Quantity	Item ID	Item Name	Supervisor ID	Actions
big basket	110	2024-05-08	5	2	salt	1005	Update Delete
kaleeswari oil hd	2100	2024-05-28	10	3	cooking oil	1007	Update Delete
rama jeyam mfls hd	2800	2024-05-08	100	5	ponni raw rice	1005	Update Delete
lc	38	2024-05-21	4	59	atta	1005	Update Delete

2024 All rights reserved, mess management system. Developed By - abraadu

Fig. 31: View Supply Details

3.16 Add Supply Details

Add Supply

Supply Name:

Cost:

Date of Arrival:

Quantity:

Item ID:

Item Name:

Supervisor ID:

Add Supply

Fig. 32: Add Supply Details

3.17 Update Supply Details

Update Supply

Supply Name:

Cost:

Date of Arrival:

Quantity:

Item Name:

Supervisor ID:

Update Supply

Fig. 33: Update Supply Details

Technology Stack

1.) **PHP:** It is utilized to handle server-side logic, such as processing form submissions, querying the database, and generating dynamic content.

2.) **Mysql:** It is employed in this project to store various data related to the Mess Management system, including menu details, employee information, and feedback submissions. PHP scripts interact with MySQL to perform operations like inserting, updating, deleting, and retrieving data from the database.

3.) **HTML:** HTML is used to create the layout and structure of the web pages, including forms for submitting feedback, displaying menu details, and presenting employee information. HTML forms facilitate user interaction by allowing them to input data and submit it to the server for processing.

4.)**CSS:** In this project, CSS is utilized to define styles for elements such as tables, buttons, and text, improving the overall user interface and experience.

5.)**Visual Studio Code:** It has been used for writing, editing, and debugging PHP, HTML and CSS project files.