NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Nitte (DU) established under Section 3 of UGC Act 1956 | Accredited with 'A+' Grade by NAAC

Department of Computer Science & Engineering

Report on Mini Project

Food Donation Management System

Course Code : CS2102-1
Course Name : **Database Management System**

Semester: IV SEM Section: A

Submitted To:

Ms. Jayapadmini Kanchan
Assistant Professor [Gd-II]
Department of Computer Science and Engineering

Submitted By:

Ansh Vashisht (NNM22CS029)
A Shreya Nair (NNM22CS001)

Date of submission:
18/04/2024

Signature of Course Instructor

ABSTRACT

The comprehensive food donation management system(FoodDan) revolutionizes the process of connecting donors, admins, and volunteers. Donors register effortlessly, and for food donation provides essential details like food type, location, food name, and quantity, while volunteers efficiently distribute food orders. Admins has an intuitive interface to monitor activities like food tracking , history of food authorized by him and feedback from donors. By using the technology, we addresses challenges of conventional systems, promoting community participation and fostering a culture of voluntary food donation and brotherhood. The primary objective is to help each other by facilitating prompt access to food during emergencies, ultimately aiming to create a more resilient and supportive society. Through this initiative, the system strives to make a meaningful impact on food delivery, encouraging individuals and communities to engage in a noble cause that benefits those in need.

ACKNOWLEDGMENT

It is with great satisfaction and enthusiasm that we are submitting the Mini Project Report on "**Food Donation System**". We have completed it as a part of the IV semester **Database Management System (CS2102-1)** of Bachelor of Engineering in **Computer Science & Engineering** of Nitte University, Deralakatte.

I take this opportunity to express my gratitude and regards to my project guide, **Ms. Jayapadmini Kanchan**, Assistant Professor Gd-II, Department of Computer Science Engineering for her guidance, monitoring, and encouragement throughout this Mini project.

I would like to express my deepest gratitude to **Dr. Jyothi Shetty**, Head of the Department of Computer Science and Engineering, for her invaluable guidance, unwavering support, and encouragement throughout my academic journey.

My sincere thanks to our beloved principal, **Dr. Niranjan N Chiplunkar**, and vice principal **Dr. I.R.Mithanthaya** for permitting us to carry out this Mini project at our college and providing us with all needed facilities.

I thank my parents and friends for the immense support, honest opinion, and suggestions throughout the mini project.

Table of Contents

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

FoodDan is a dedicated website designed to simplify and enhance the food donation process. Through user-friendly registration, donor can effortlessly sign up, contributing to a growing database accessible to anyone in need. The platform allows users to request food efficiently, connecting them with people based location, and availability. With a secure and confidential environment, we ensure the privacy and integrity of user data. The admin control panel empowers administrators to manage donor listings, food listings ,users feedback, and oversees the platform's seamless functioning. FoodDan fosters community engagement, creating a space where verified recipients can get food easily , ultimately making a positive impact on society. Join FoodDan today and play a pivotal role in the network of compassionate contributors dedicated to removing hunger and helping people in need.

1.1 Purpose

The primary aim of the Food Donation Management system is to establish a digital platform for managing and recording food donations, while also facilitating interactions between donors and recipients. Our goal is to showcase the implementation of fundamental database operations such as create, read, update, and delete (CRUD) using MySQL. The system commences with the establishment of a donation platform, enabling donors to submit details of the food items they wish to contribute. Concurrently, recipients can access the platform to view available donations and make requests based on their needs. Administrators play a pivotal role in overseeing donation activities, managing inventory, and coordinating volunteer efforts to ensure seamless operations. By providing comprehensive descriptions of available food items, the system empowers recipients to make informed decisions when selecting items that align with their preferences and requirements. Ultimately, the Food Donation Management System strives to optimize the process of food distribution, fostering community engagement and addressing food scarcity issues.

1.2 Scope

To tackle the challenges associated with manual record-keeping and coordination in food donation processes, the Food Donation Management System has been developed. We offer a digital solution for efficiently managing food donation activities and interactions between donors, administrators, and volunteers. It provides a centralized system for tracking food donations, allowing administrators to manage inventory, donor records, and volunteer activities seamlessly. With customizable features, administrators can adapt the system to their specific needs, ensuring flexibility and ease of use. By digitizing the food donation process, this system aims to enhance coordination, transparency, and efficiency, ultimately contributing to the reduction of food insecurity in local communities.

1.3 Overview

The project starts by creating a Food Donation Organization and adding details of food donated by donors and volunteers associated with it. The role of an admin is to manage daily activities in a food donation like looking after the food donated, their quality and how all need food .The donor can register to the website by providing required details. This system will provide the detailed description of the food to the person who is needy so that they can view the different food which are listed and can gut the desired the food easily.

CHAPTER 2

REQUIREMENTS SPECIFICATION

2.1 Hardware Specification

- Processor : Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz
- RAM : 8GB
- Hard Disk : 1TB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

2.2 Software Specification

- Database: MySQL 5.5
- Markup Language: HTML5
- Scripting Language: PHP 7.0.1
- IDE: Visual Studio Code
- Server: Apache
- Browser: Google Chrome, Microsoft Edge , Firefox

CHAPTER 3

SYSTEM DESIGN

3.1 ER Diagram

For the project there are 5 strong entities Head, Delivery, Food, User and Feedback. Head has 4 attributes aid, name, email and location where aid is primary key. Review has reviewid, time, information, email and reviewed is primary key. For Delivery there are did ,city, name, email, pickup where did is primary key. Food has fid, name, address, quantity, category where fid is primary key. User has 4 attributed which are uid, email,name, gender where uid is primary key. All the ids in the entities are auto-incrementing and primary keys.

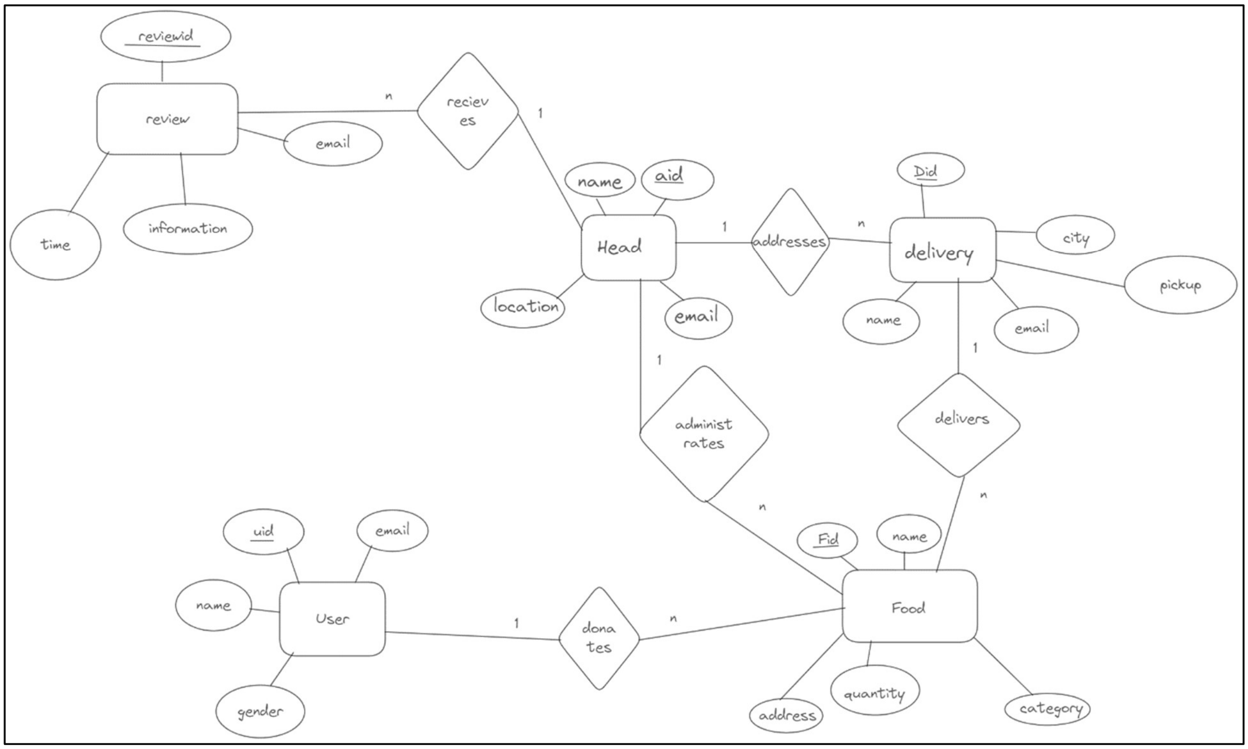


Fig 3.1: ER Diagram for food donation

3.2 Mapping From ER diagram to Schema

To convert ER to schema there are 7 steps which are as follows:

1. **Mapping of Regular Entities**: This step involves mapping all the regular strong entities types to tabular format by identifying their primary keys.
2. **Mapping of 1:1 Relation:** In this step foreign keys are assigned using foreign key approach. The primary key of the participating relations is added as primary key to second entity types by looking at the participating constraints.
3. **Mapping of 1:N Relation:** Foreign key approach is used to add one sided primary key to the n sided entity at foreign key.
4. **Mapping of M:N Relation:** Here we use the cross-reference approach where the relationship is converted to a new relation within attributes on primary keys of both participating relations.
5. **Mapping of Weak Entity:** When mapping weak entity types along with other attributes the partial key and primary key of parent entity together will form their primary key of the new relation.
6. **Mapping of N-ary Relation:** For mapping N array relationship we create a new relation with a relationship name in its attribute and primary keys of all participating entity types.
7. **Mapping of Multivalued Relation:** For multivalued attributes a separate relation has to be created along with primary key of parent relation.

To get schema for database these steps need to be followed:

1. **Mapping of Regular Entities:** Initially strong entities are identified(the entities which have primary key in them). In database these are the strong entities with the attributes

   Head (aid , name , email , location)

   Delivery (did , city , pickup , name , email , aid)

   Review (reviewid , time , information , email , location)

   User (uid , name , email , gender , location)

   Food (fid , name , category , quantity , address)
2. **Mapping of 1:1 Relation**: None of the entities are participating in the 1:1 relation type.

In it each record in 1 table corresponds uniquely to a record in another table.

3. **Mapping of 1:N Relation:** In database all the entities are participating in 1:n. In a one-to-many relationship, the "n" side entity includes a foreign key referencing the primary key of the "one" side entity.

   The entities and attributes which are in 1:n are :

   Head (aid , name , email , location)

   Delivery (did , city , pickup , name , email , aid)

   Review (reviewid , time , information , email , location)

   User (uid , name , email , gender , location)

   Food (fid , name , category , quantity , address)

4. **Mapping of M:N Relation:** None of the entities are participating in m:n relation. In a many-to-many relationship, a separate associative entity is created to link the participating entities.

5. **Mapping of Weak Entities:** Next step is to identify the weak entities(the entities which don’t have primary key in them). In database these are no entities with such attributes.

6. **Mapping of N-ary Relation**: None of the entities are participating in this relation. In it the relation is linked to and linked from same entity.

7. **Mapping of Multivalued relation:** A multivalued attribute allows an entity to have multiple values for a single attribute. This is typically represented as a separate table with a foreign key referencing the primary key of the original entity.

3.3 Assumptions

1. Assumption of Database Usage: The assumption that the MySQL database is used for storing data related to the project.

2. Assumption of Table Structure: The assumption that the database consists of several tables representing different entities such as admin, delivery persons, food donations, login, and user feedback.

3. Assumption of Primary Keys: The assumption that each table has a primary key column (e.g., aid, did, fid, uid, feedbackid) to uniquely identify each record within that table.

4. Assumption of Indexes: The assumption that indexes are created on certain columns (e.g., email, id) to improve query performance.

5. Assumption of Auto-increment: The assumption that certain primary key columns (e.g., Aid, Did, Fid, id, feedbackid) are set to auto-increment to automatically generate unique values for new records.

3.4 Schema Diagram

A Schema is a pictorial representation of the relationship between the tables in the database that is created. The term ”schema” refers to the representation of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a schema is a set of formulas (sentences) called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language .The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modelled in the database.

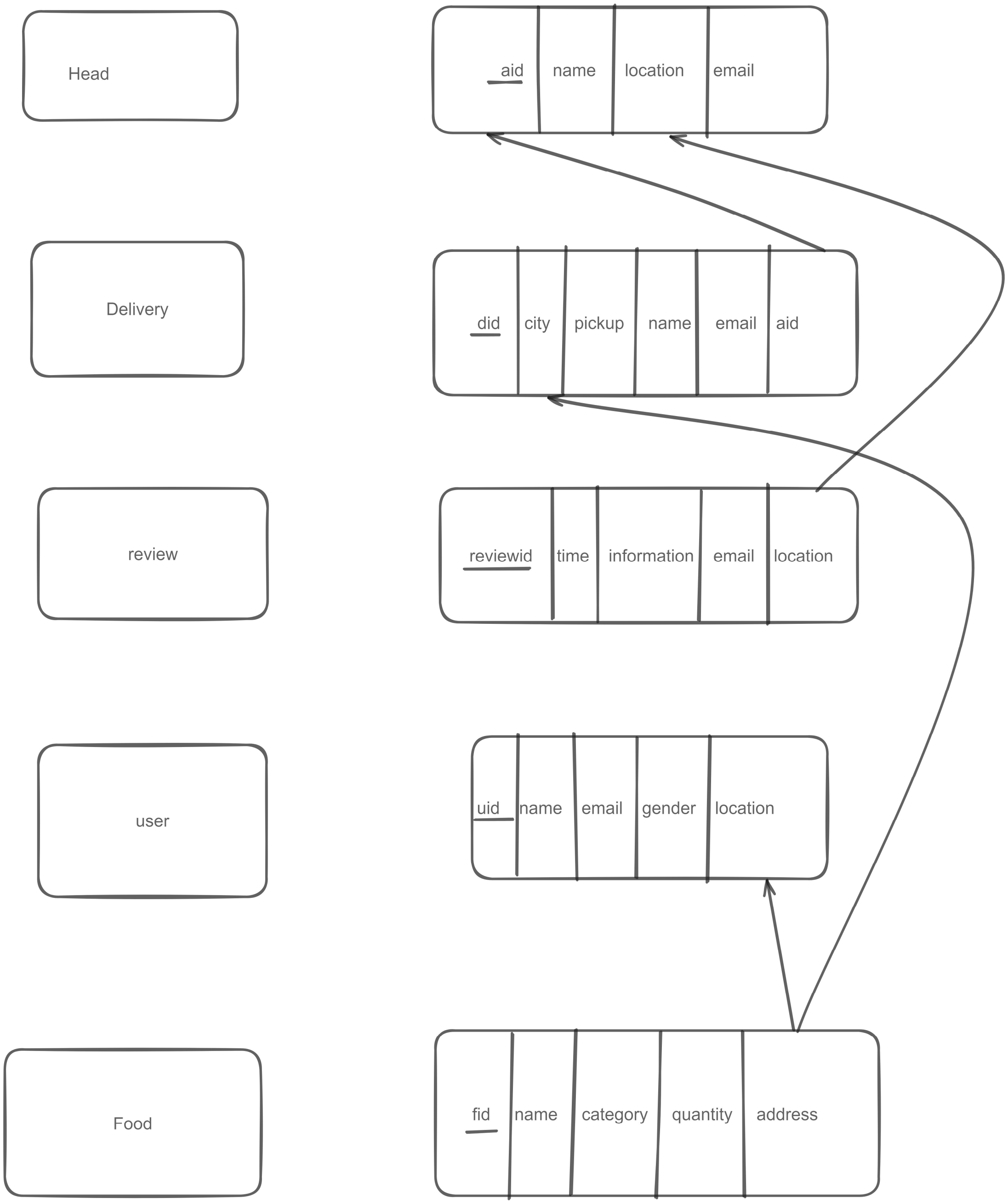

Head
aid | name | location | email
Delivery
did | city | pickup | name | email | aid
review
reviewid | time | information | email | location
user
uid | name | email | gender | location
Food
fid | name | category | quantity | address

Fig 3.2: Schema Diagram of food donation

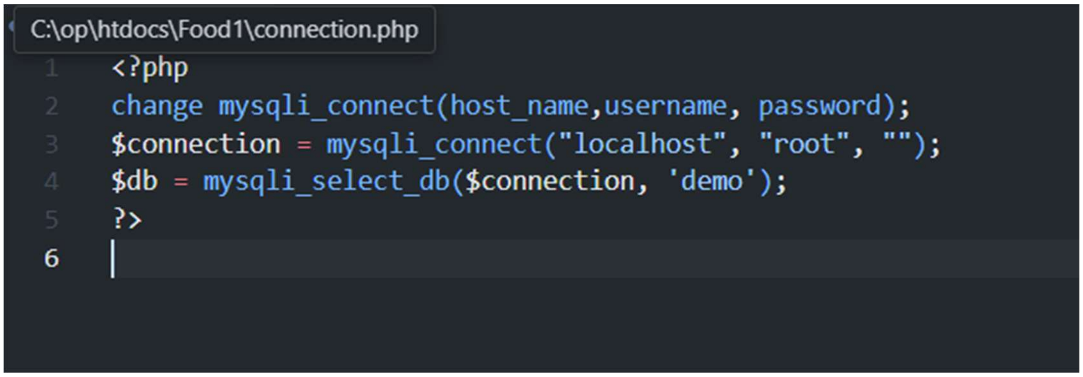
CHAPTER 4

IMPLEMENTATION

4.1 Pseudocodes used

Pseudocode to connect SQL and PHP:

In order to store or access the data inside a MySQL database, the first step is to connect to the MySQL database server. In PHP we can do this using the mysqli connect() function. All communication between PHP and the MySQL database server takes place through this connection .The hostname parameter in the above syntax specify the host name, whereas the username and password parameters specifies the credentials to access MySQL server, and the database parameter, if provided will specify the default MySQL database to be used when performing queries. The default username for MySQL database server is root and there is no password and hostname is localhost.



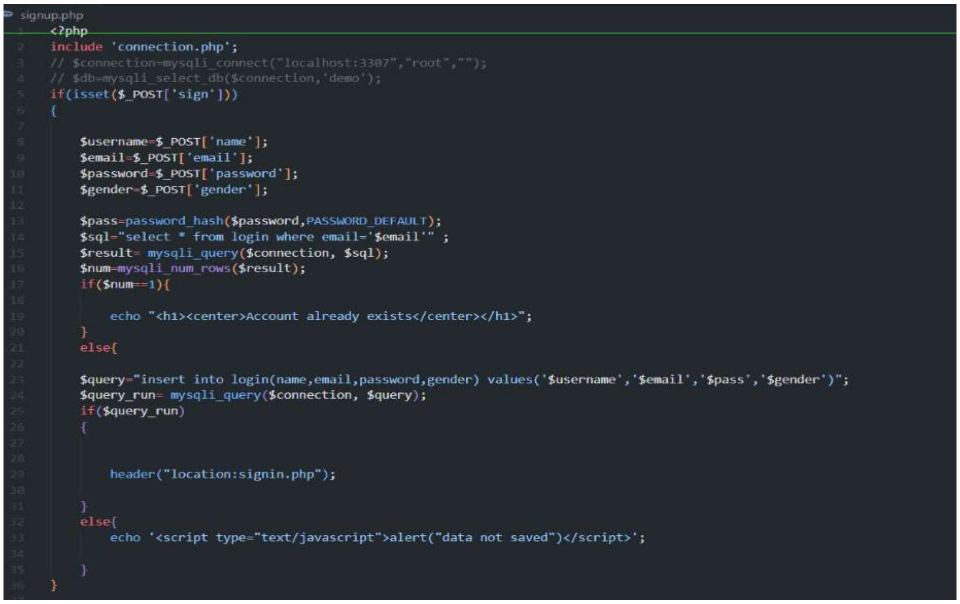
C:\op\htdocs\Food1\connection.php

```
<?php
change mysqli_connect(host_name,username, password);
$connection = mysqli_connect("localhost", "root", "");
$db = mysqli_select_db($connection, 'demo');
?>
```

Fig 4.1 Pseudocode to connect SQL

Pseudocode for INSERT :

Insert statement is a DML (Data modification language) statement which is used to insert data in the MySQL table. PHP$ POST is a PHP super global variable which is used to collect the form data from the user.

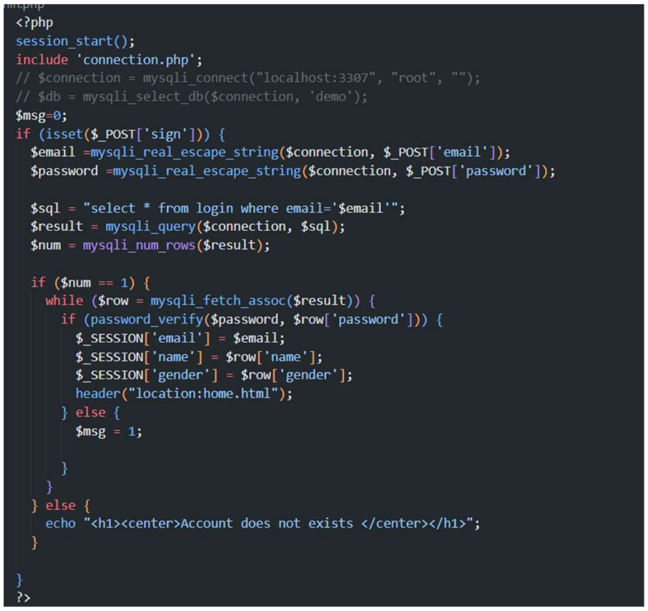


```php
signup.php
<?php
include 'connection.php';
// $connection=mysqli_connect("localhost:3307","root","");
// $db=mysqli_select_db($connection,'demo');
if(isset($_POST['sign']))
{

    $username=$_POST['name'];
    $email=$_POST['email'];
    $password=$_POST['password'];
    $gender=$_POST['gender'];

    $pass=password_hash($password,PASSWORD_DEFAULT);
    $sql="select * from login where email='$email'" ;
    $result= mysqli_query($connection, $sql);
    $num=mysqli_num_rows($result);
    if($num==1){

        echo "<h1><center>Account already exists</center></h1>";
    }
    else{

    $query="insert into login(name,email,password,gender) values('$username','$email','$pass','$gender')";
    $query_run= mysqli_query($connection, $query);
    if($query_run)
    {


        header("location:signin.php");

    }
    else{
        echo '<script type="text/javascript">alert("data not saved")</script>';

    }
}
```

.　Fig 4.2: Sign-up code

Pseudocode for SELECT:

The SELECT statement is used to print the existing records in a table. We can put condition by using WHERE clause.

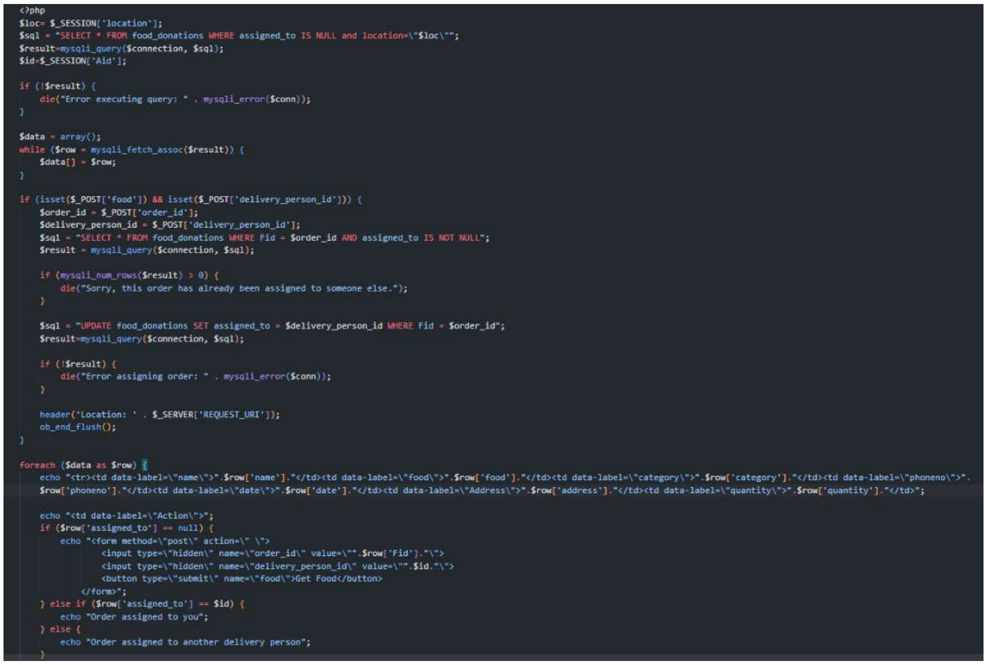


```php
<?php
session_start();
include 'connection.php';
// $connection = mysqli_connect("localhost:3307", "root", "");
// $db = mysqli_select_db($connection, 'demo');
$msg=0;
if (isset($_POST['sign'])) {
  $email =mysqli_real_escape_string($connection, $_POST['email']);
  $password =mysqli_real_escape_string($connection, $_POST['password']);

  $sql = "select * from login where email='$email'";
  $result = mysqli_query($connection, $sql);
  $num = mysqli_num_rows($result);

  if ($num == 1) {
    while ($row = mysqli_fetch_assoc($result)) {
      if (password_verify($password, $row['password'])) {
        $_SESSION['email'] = $email;
        $_SESSION['name'] = $row['name'];
        $_SESSION['gender'] = $row['gender'];
        header("location:home.html");
      } else {
        $msg = 1;

      }
    }
  } else {
    echo "<h1><center>Account does not exists </center></h1>";
  }

}
?>
```

Fig 4.3 Sign-in code

Pseudocode for UPDATE:

The UPDATE statement is used to modify the existing records in a table. The WHERE clause specifies which record(s) that should be updated.



```
<?php
$loc= $_SESSION['location'];
$sql = "SELECT * FROM food_donations WHERE assigned_to IS NULL and location=\"$loc\"";
$result=mysqli_query($connection, $sql);
$id=$_SESSION['Aid'];

if (!$result) {
    die("Error executing query: " . mysqli_error($conn));
}

$data = array();
while ($row = mysqli_fetch_assoc($result)) {
    $data[] = $row;
}

if (isset($_POST['food']) && isset($_POST['delivery_person_id'])) {
    $order_id = $_POST['order_id'];
    $delivery_person_id = $_POST['delivery_person_id'];
    $sql = "SELECT * FROM food_donations WHERE Fid = $order_id AND assigned_to IS NOT NULL";
    $result = mysqli_query($connection, $sql);

    if (mysqli_num_rows($result) > 0) {
        die("Sorry, this order has already been assigned to someone else.");
    }

    $sql = "UPDATE food_donations SET assigned_to = $delivery_person_id WHERE Fid = $order_id";
    $result=mysqli_query($connection, $sql);

    if (!$result) {
        die("Error assigning order: " . mysqli_error($conn));
    }

    header('Location: ' . $_SERVER['REQUEST_URI']);
    ob_end_flush();
}

foreach ($data as $row) {
    echo "<tr><td data-label=\"name\">".$row['name']."</td><td data-label=\"food\">".$row['food']."</td><td data-label=\"category\">".$row['category']."</td><td data-label=\"phoneno\">".
    $row['phoneno']."</td><td data-label=\"date\">".$row['date']."</td><td data-label=\"Address\">".$row['address']."</td><td data-label=\"quantity\">".$row['quantity']."</td>";

    echo "<td data-label=\"Action\">";
    if ($row['assigned_to'] == null) {
        echo "<form method=\"post\" action=\" \">
                <input type=\"hidden\" name=\"order_id\" value=\"".$row['Fid']."\">
                <input type=\"hidden\" name=\"delivery_person_id\" value=\"".$id."\">
                <button type=\"submit\" name=\"food\">Get Food</button>
            </form>";
    } else if ($row['assigned_to'] == $id) {
        echo "Order assigned to you";
    } else {
        echo "Order assigned to another delivery person";
    }
```

Fig 4.4 To Update values in table

4.2 Tables used

In database there are total 5 tables are used and they are as following:

Head (aid , name , email , location)

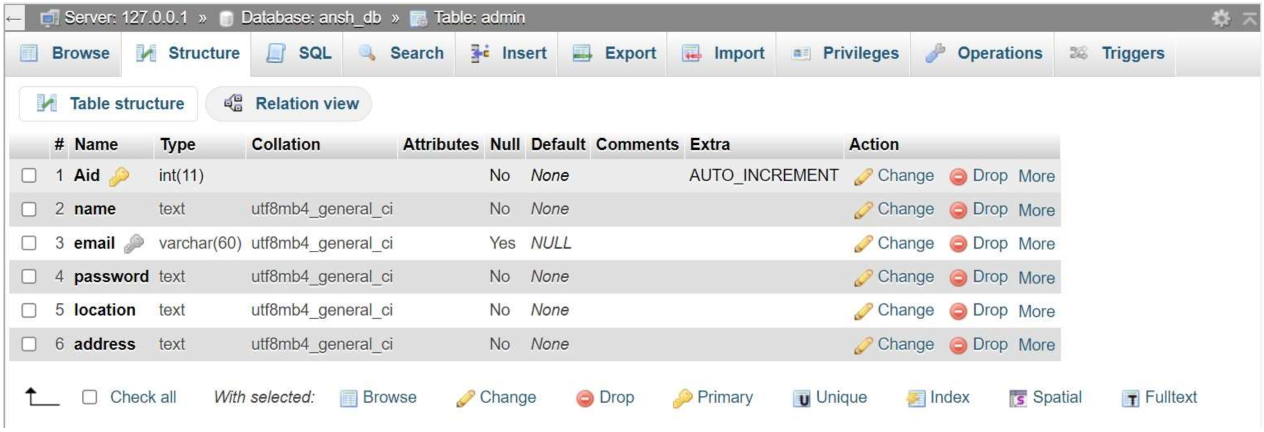
Delivery (did , city , pickup , name , email , aid)

Review (reviewid , time , information , email , location)

User (uid , name , email , gender , location)

Food (fid , name , category , quantity , address)

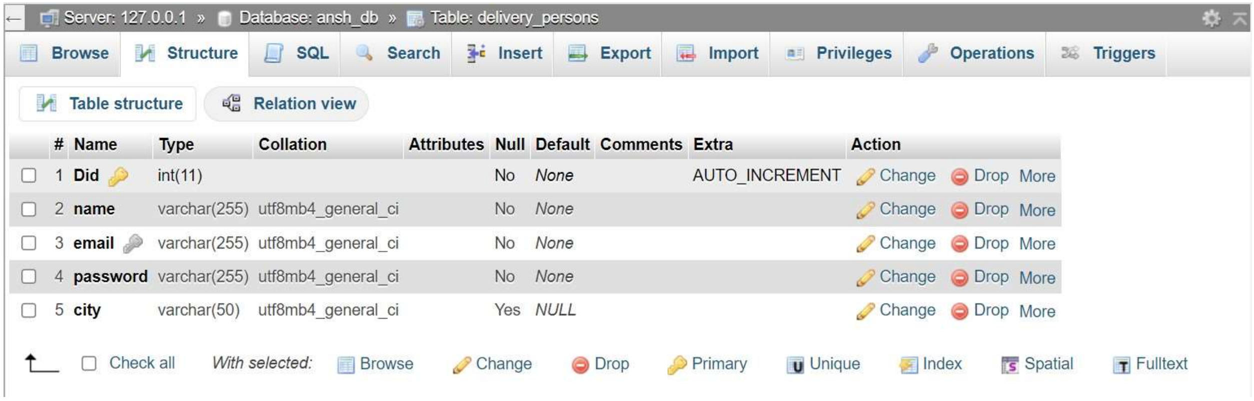
Admin table has aid , name , email , password , location and address as attributes and aid is primary key.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	Aid	int(11)			No	None		AUTO_INCREMENT	Change Drop More
□	2	name	text	utf8mb4_general_ci		No	None			Change Drop More
□	3	email	varchar(60)	utf8mb4_general_ci		Yes	NULL			Change Drop More
□	4	password	text	utf8mb4_general_ci		No	None			Change Drop More
□	5	location	text	utf8mb4_general_ci		No	None			Change Drop More
□	6	address	text	utf8mb4_general_ci		No	None			Change Drop More

Fig 4.5 Admin Table

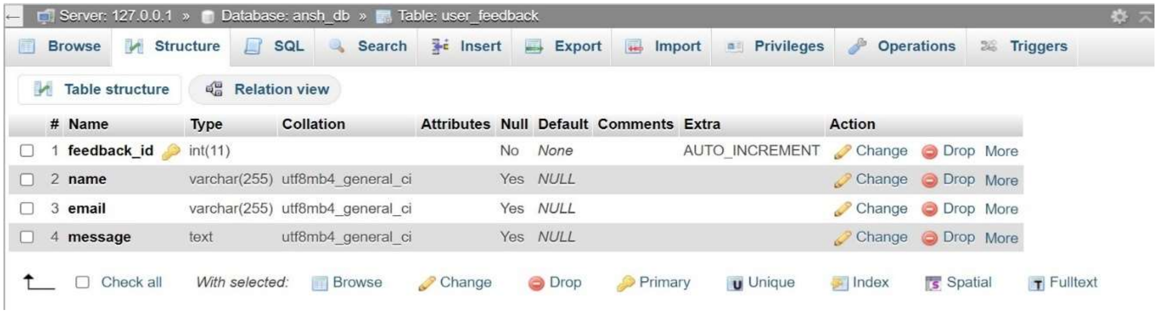
Delivery table has did , name , email , password and city as attributes and did is primary key.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	Did	int(11)			No	None		AUTO_INCREMENT	Change Drop More
□	2	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
□	3	email	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
□	4	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
□	5	city	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Fig 4.6 Delivery Table

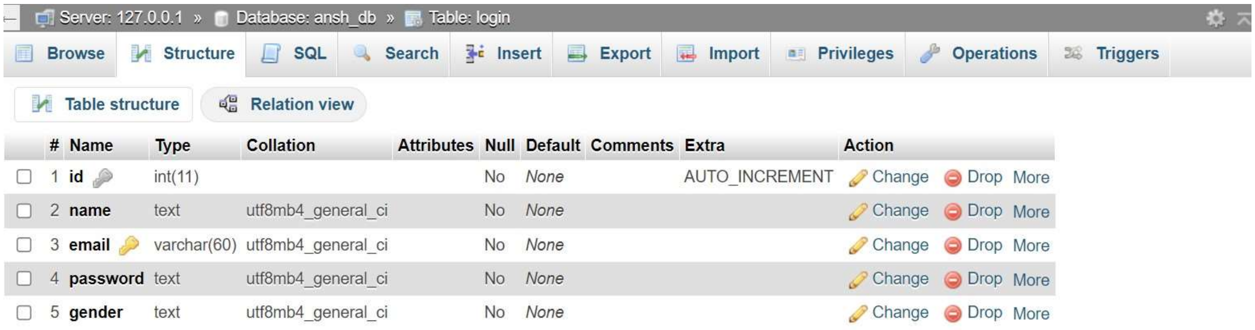
Review table has feedback_id , name , email ,and message as attributes and feedback_id is primary key.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	feedback_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
□	2	name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
□	3	email	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
□	4	message	text	utf8mb4_general_ci		Yes	NULL			Change Drop More

Fig 4.7 Review Table

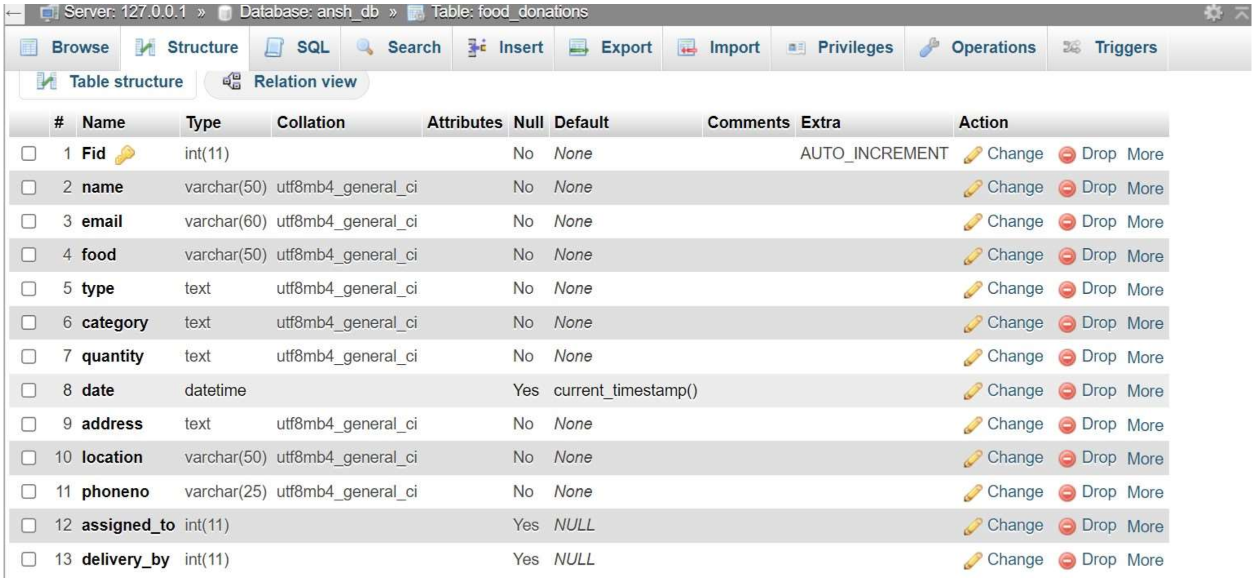
User table has id , name , email , password and gender as attributes and id is primary key.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**name**	text	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	**email**	varchar(60)	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	**password**	text	utf8mb4_general_ci		No	*None*			Change Drop More
□	5	**gender**	text	utf8mb4_general_ci		No	*None*			Change Drop More

Fig 4.8 User Table

Food table has fid , name , email , type, category , quality , date , address , location , phoneno , assigned_to and delivery_to as attributes and fid is primary key.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**Fid**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**name**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	**email**	varchar(60)	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	**food**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
□	5	**type**	text	utf8mb4_general_ci		No	*None*			Change Drop More
□	6	**category**	text	utf8mb4_general_ci		No	*None*			Change Drop More
□	7	**quantity**	text	utf8mb4_general_ci		No	*None*			Change Drop More
□	8	**date**	datetime			Yes	current_timestamp()			Change Drop More
□	9	**address**	text	utf8mb4_general_ci		No	*None*			Change Drop More
□	10	**location**	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
□	11	**phoneno**	varchar(25)	utf8mb4_general_ci		No	*None*			Change Drop More
□	12	**assigned_to**	int(11)			Yes	*NULL*			Change Drop More
□	13	**delivery_by**	int(11)			Yes	*NULL*			Change Drop More

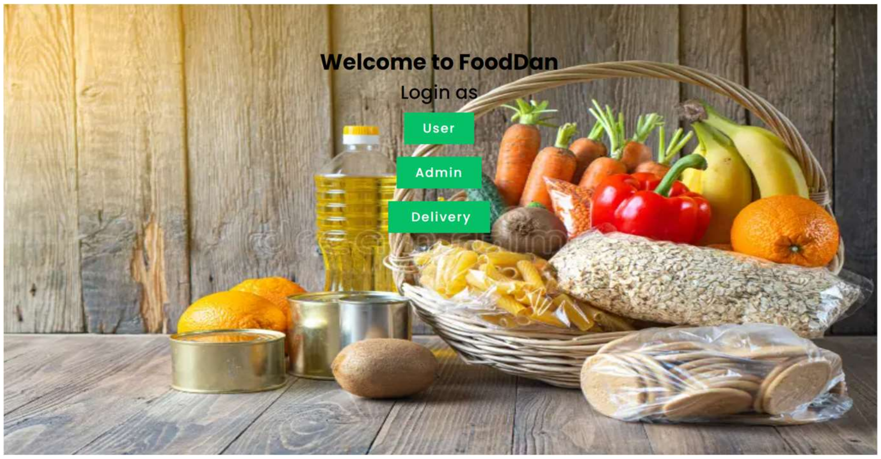
Fig 4.9 Food Table

CHAPTER 5

RESULTS AND DISCUSSION

Sign-Up Page:

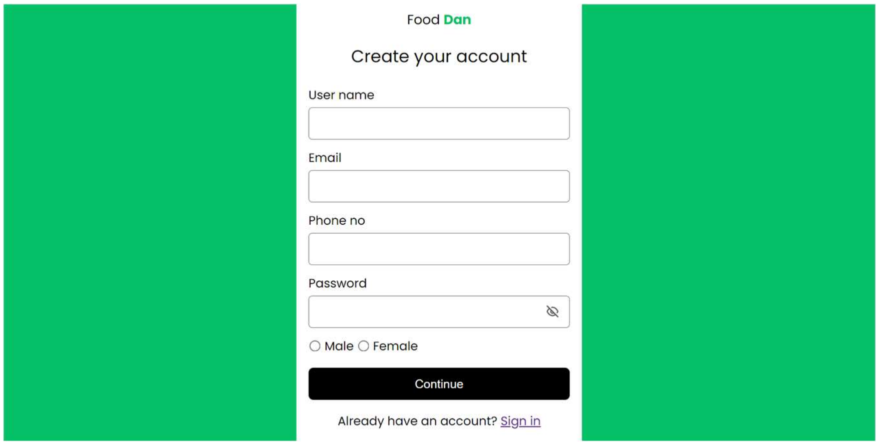
Through this page user , admin and volunteer can sign up to the food donation website.



Welcome to FoodDan

Login as

Fig 5.1: Home Page



Food Dan

Create your account

User name

Email

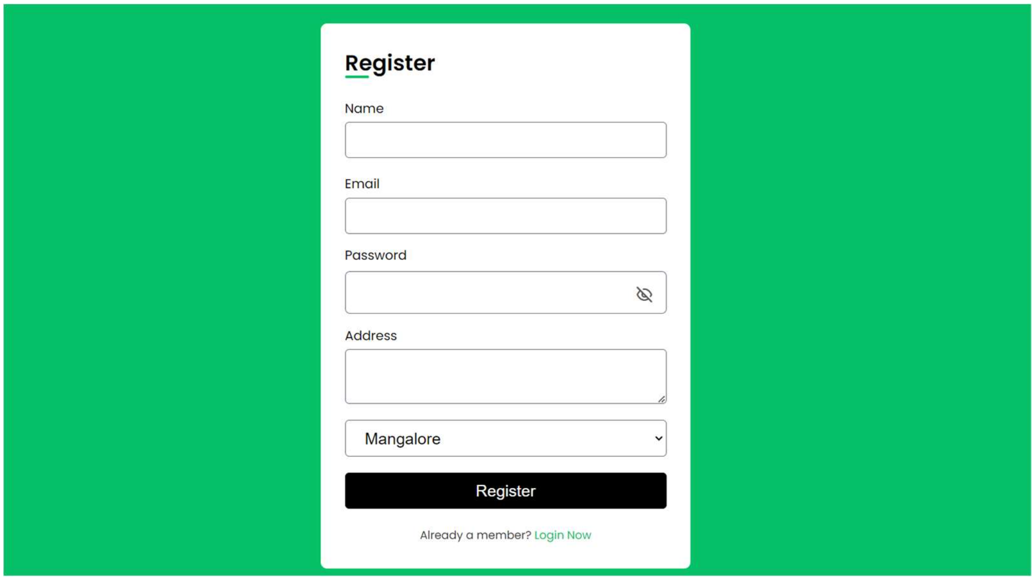
Phone no

Password

○ Male ○ Female

Already have an account? Sign in

Fig 5.2: Donor Sign-up Page



Register

Name

Email

Password

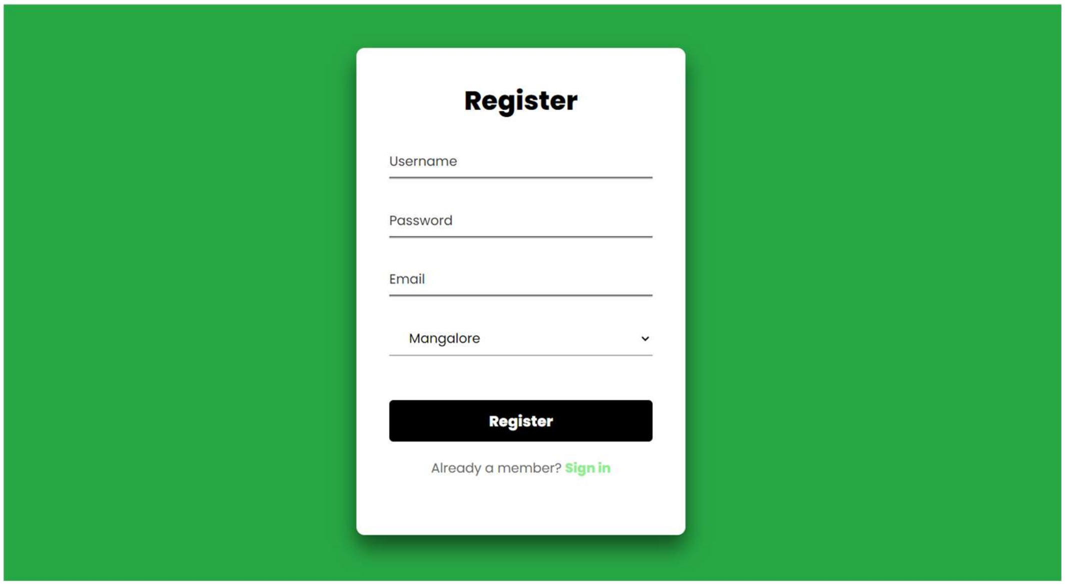
Address

Mangalore

Register

Already a member? Login Now

Fig 5.3: User Sign-Up Page



Register

Username

Password

Email

Mangalore

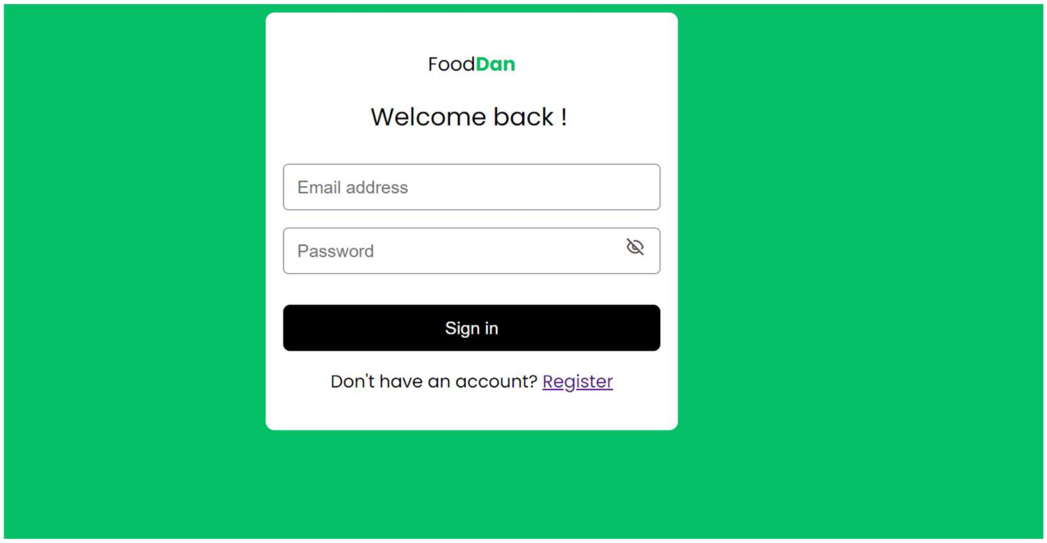
Register

Already a member? Sign in

Fig 5.4: Volunteer Sign-Up Page

Sign-In Page:

Through this page user, admin and volunteer can sign in to the food donation website if they already have an account in it.

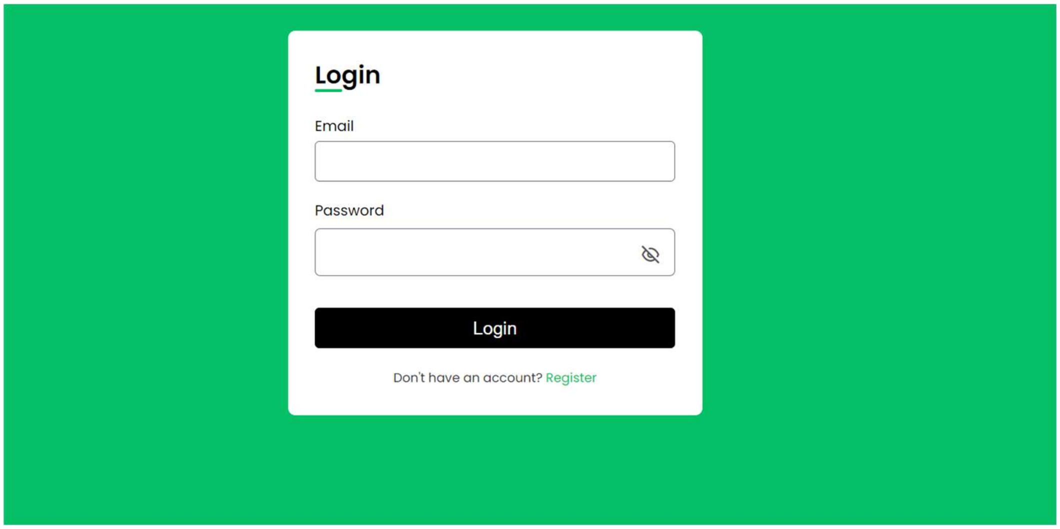


FoodDan

Welcome back !

Don't have an account? Register

Fig: 5.5 User Sign-In Page



Don't have an account? Register

Fig: 5.6 Admin Sign-In Page

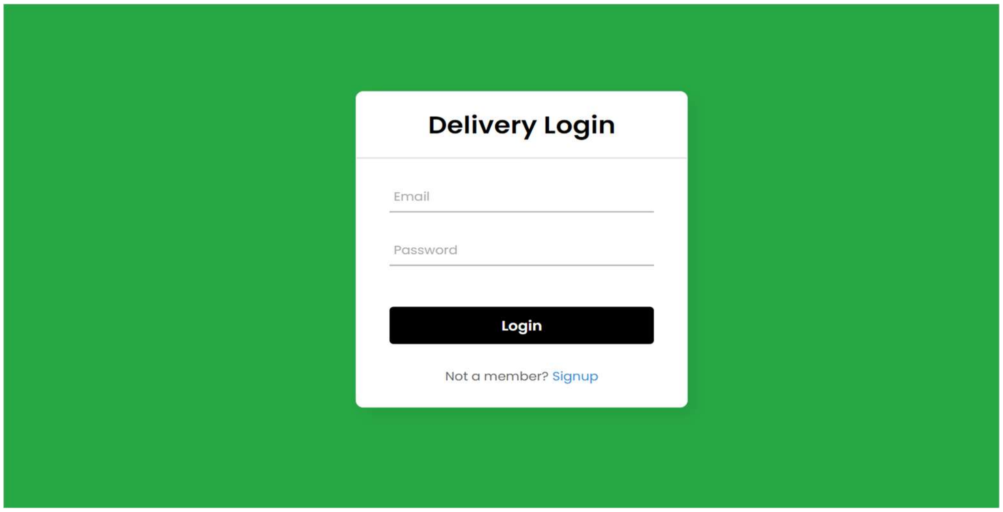


Fig 5.7: Volunteer Sign-In Page

User:

In it, donor can put the message that he is ready to donate food, he or she can give feedback to the admin and has catalogue of all the donations done by the donor.

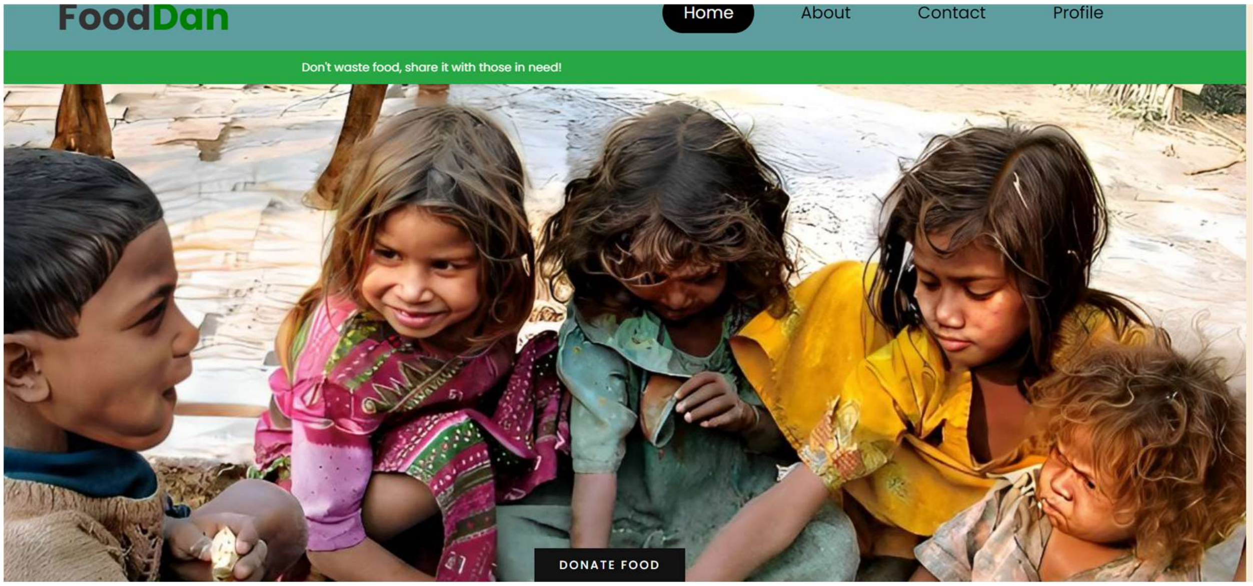
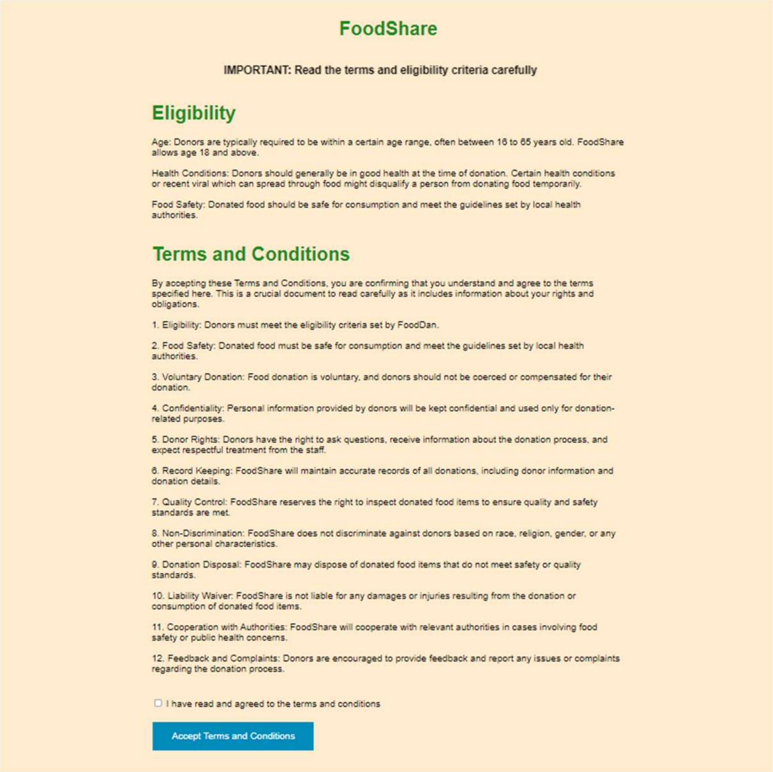


Fig 5.8: User HomePage



FoodShare

IMPORTANT: Read the terms and eligibility criteria carefully

Eligibility

Age: Donors are typically required to be within a certain age range, often between 16 to 65 years old. FoodShare allows age 18 and above.

Health Conditions: Donors should generally be in good health at the time of donation. Certain health conditions or recent viral which can spread through food might disqualify a person from donating food temporarily.

Food Safety: Donated food should be safe for consumption and meet the guidelines set by local health authorities.

Terms and Conditions

By accepting these Terms and Conditions, you are confirming that you understand and agree to the terms specified here. This is a crucial document to read carefully as it includes information about your rights and obligations.

1. Eligibility: Donors must meet the eligibility criteria set by FoodDan.

2. Food Safety: Donated food must be safe for consumption and meet the guidelines set by local health authorities.

3. Voluntary Donation: Food donation is voluntary, and donors should not be coerced or compensated for their donation.

4. Confidentiality: Personal information provided by donors will be kept confidential and used only for donation-related purposes.

5. Donor Rights: Donors have the right to ask questions, receive information about the donation process, and expect respectful treatment from the staff.

6. Record Keeping: FoodShare will maintain accurate records of all donations, including donor information and donation details.

7. Quality Control: FoodShare reserves the right to inspect donated food items to ensure quality and safety standards are met.

8. Non-Discrimination: FoodShare does not discriminate against donors based on race, religion, gender, or any other personal characteristics.

9. Donation Disposal: FoodShare may dispose of donated food items that do not meet safety or quality standards.

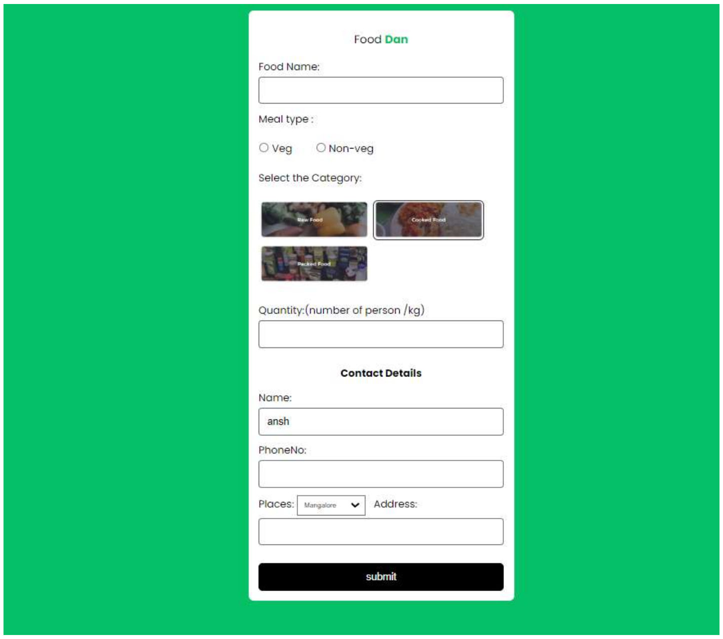
10. Liability Waiver: FoodShare is not liable for any damages or injuries resulting from the donation or consumption of donated food items.

11. Cooperation with Authorities: FoodShare will cooperate with relevant authorities in cases involving food safety or public health concerns.

12. Feedback and Complaints: Donors are encouraged to provide feedback and report any issues or complaints regarding the donation process.

□ I have read and agreed to the terms and conditions

Fig 5.9: Terms and Condition



Food Dan

Food Name:

Meal type :

○ Veg ○ Non-veg

Select the Category:

Quantity:(number of person /kg)

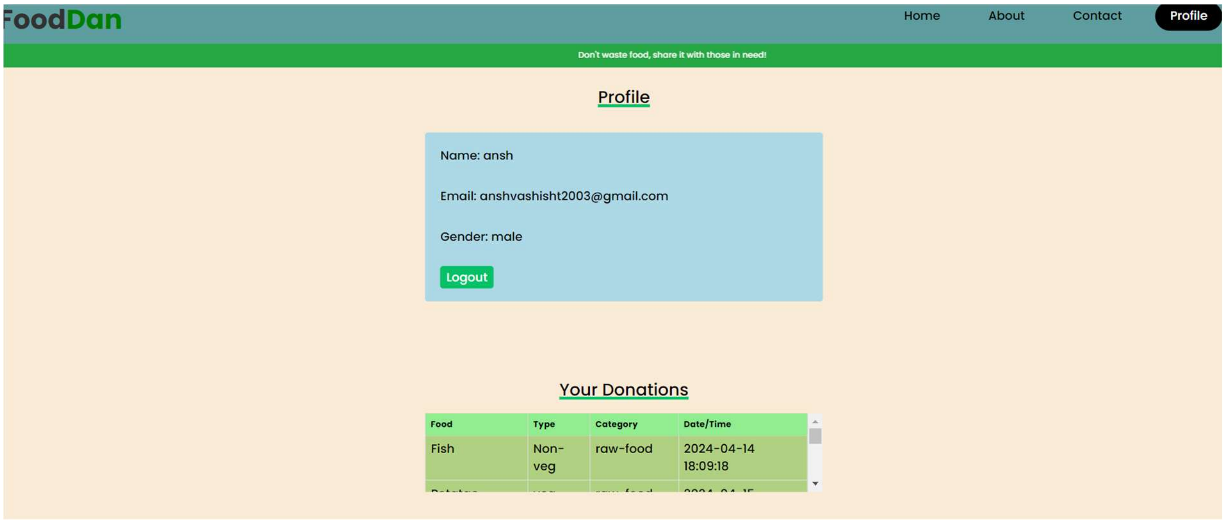
Contact Details

Name: ansh

PhoneNo:

Places: Mangalore Address:

Fig 5.10: Food Donate Form



Don't waste food, share it with those in need!

Profile

Name: ansh

Email: anshvashisht2003@gmail.com

Gender: male

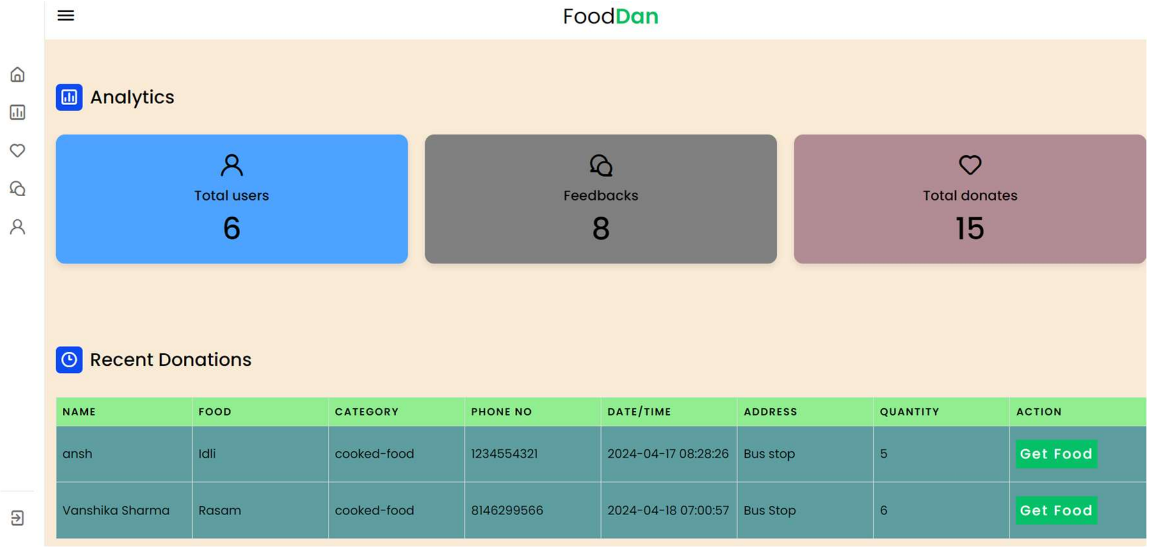
Your Donations

Food	Type	Category	Date/Time
Fish	Non-veg	raw-food	2024-04-14 18:09:18

Fig 5.11: Profile Page

Admin:

In it, admin can see all the activities taking palce through the website. He will come to know the amount if food donated by an individual , how many orders he authorized and what was the donors feedback to him.



Analytics

Total users
6

Feedbacks
8

Total donates
15

Recent Donations

NAME	FOOD	CATEGORY	PHONE NO	DATE/TIME	ADDRESS	QUANTITY	ACTION
ansh	Idli	cooked-food	1234554321	2024-04-17 08:28:26	Bus stop	5	Get Food
Vanshika Sharma	Rasam	cooked-food	8146299566	2024-04-18 07:00:57	Bus Stop	6	Get Food

Fig: 5.12 Admin homepage

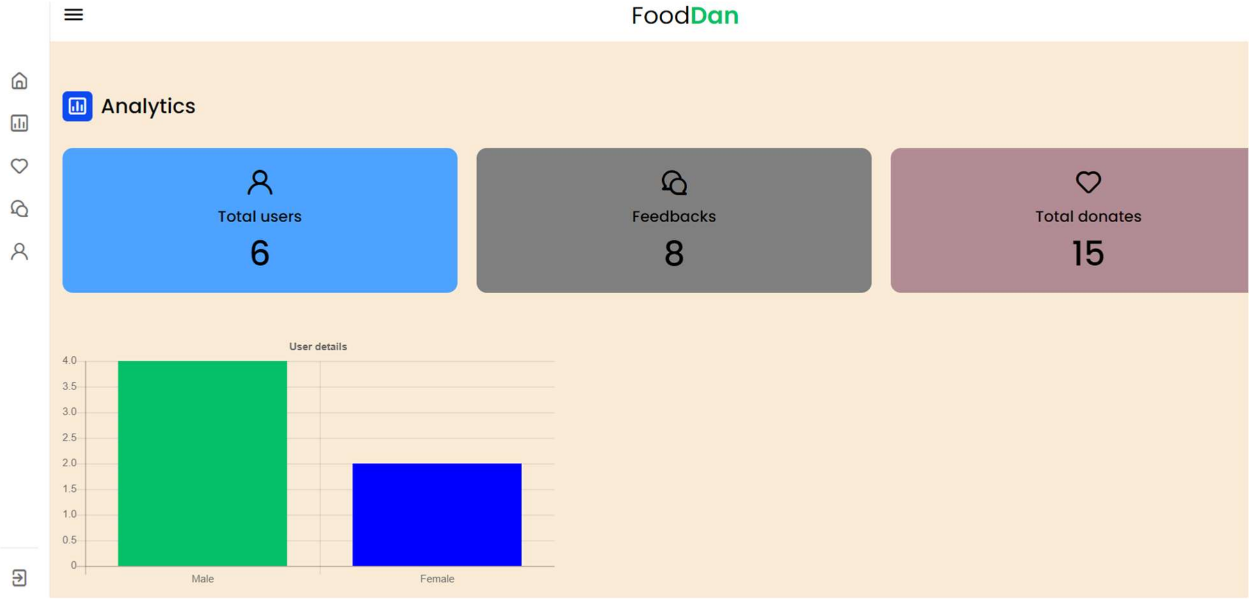
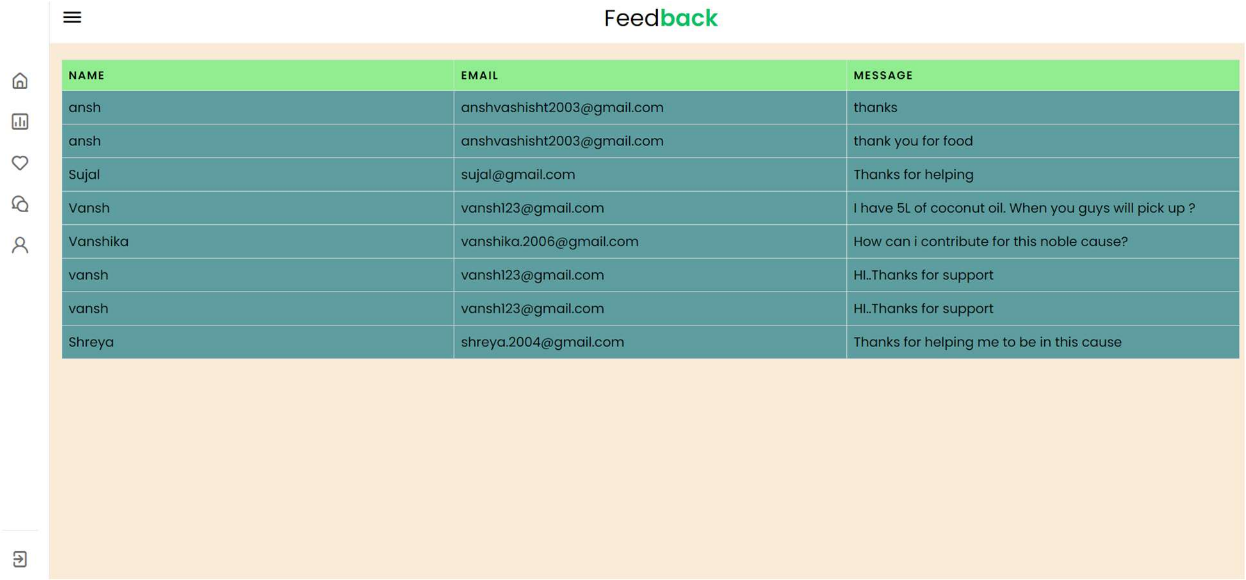
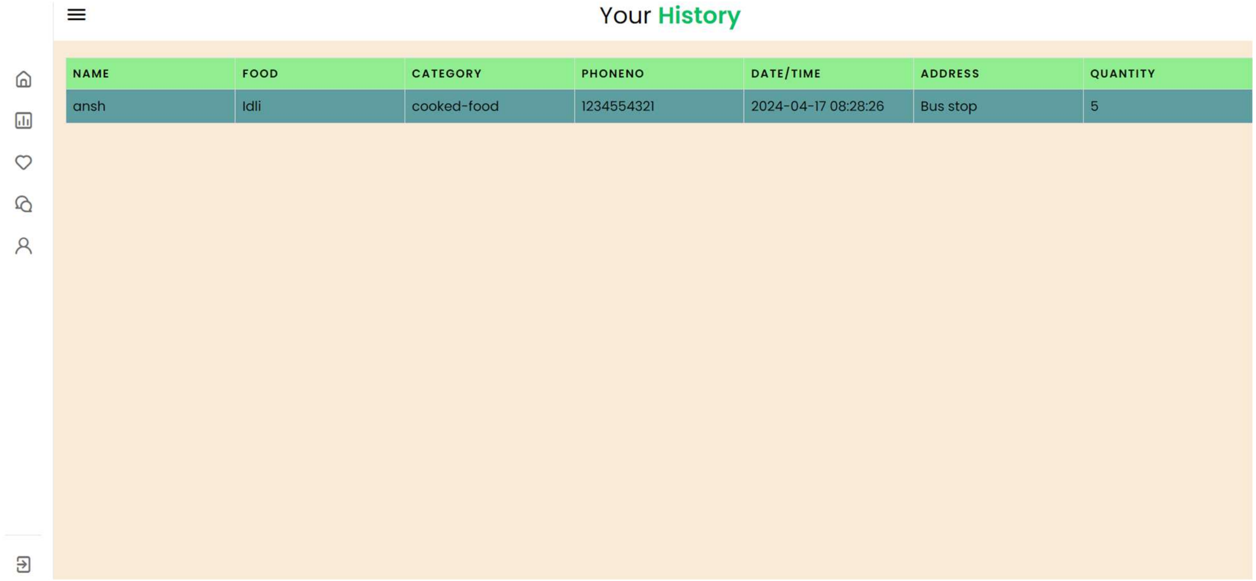


Fig 5.13: Admin Analytics Page



NAME	EMAIL	MESSAGE
ansh	anshvashisht2003@gmail.com	thanks
ansh	anshvashisht2003@gmail.com	thank you for food
Sujal	sujal@gmail.com	Thanks for helping
Vansh	vansh123@gmail.com	I have 5L of coconut oil. When you guys will pick up ?
Vanshika	vanshika.2006@gmail.com	How can i contribute for this noble cause?
vansh	vansh123@gmail.com	HI..Thanks for support
vansh	vansh123@gmail.com	HI..Thanks for support
Shreya	shreya.2004@gmail.com	Thanks for helping me to be in this cause

Fig 5.14: Feedback page



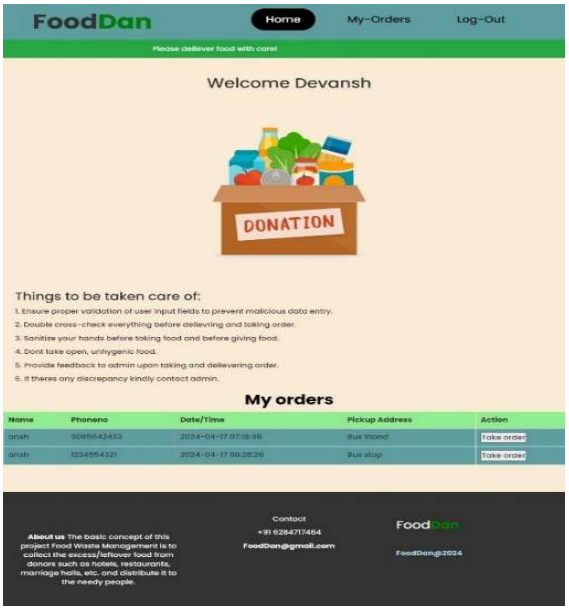
Your History

NAME	FOOD	CATEGORY	PHONENO	DATE/TIME	ADDRESS	QUANTITY
ansh	Idli	cooked-food	1234554321	2024-04-17 08:28:26	Bus stop	5

Fig 5.15: Food Authorized Page

Volunteer:

In it, volunteer can see all the food donation requests which needs transportation in the city from where he belongs. He can take them and help in its distribution in the city. He can also find all the orders which are done by him.



Please deliever food with care!

Welcome Devansh

Things to be taken care of:

1. Ensure proper validation of user input fields to prevent malicious data entry.
2. Double cross-check everything before delievring and taking order.
3. Sanitize your hands before taking food and before giving food.
4. Dont take open, unhygenic food.
5. Provide feedback to admin upon taking and delievering order.
6. If theres any discrepancy kindly contact admin.

My orders

Name	Phoneno	Date/Time	Pickup Address	Action
ansh	9085642453	2024-04-17 07:18:46	Bus Stand	Take order
ansh	1234554321	2024-04-17 08:28:26	Bus stop	Take order

About us The basic concept of this project Food Waste Management is to collect the excess/leftover food from donors such as hotels, restaurants, marriage halls, etc. and distribute it to the needy people.

Contact
+91 6284717464
FoodDan@gmail.com

FoodDan@2024

Fig 5.16: Volunteer Home Page

CHAPTER 6

CONCLUSION AND FUTURE WORK

The project represents a pivotal advancement in addressing food insecurity and promoting community welfare through digital innovation and compassionate engagement. We could do so only with help of PHP, MySQL, and HTML. The platform revolutionizes the process of food donation, fostering a dynamic ecosystem of giving and receiving. By providing a user-friendly interface, the system facilitates seamless communication and collaboration, ensuring that surplus food reaches those in need efficiently and effectively.

As the Food Donation Management System continues to evolve through continuous iteration and refinement, it remains steadfast in its mission to alleviate food insecurity, promote sustainable food distribution, and foster a culture of generosity and compassion within local communities. By leveraging technology to facilitate meaningful connections between donors and recipients, the platform exemplifies the intersection of innovation and humanity, making food donation a seamless, efficient, and deeply impactful experience for all involved.

REFERENCES

[1] Database systems Models, Languages, Design and Application Programming, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, Pearson.

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

[3] Silberschatz Korth and Sudharshan: Database System Concepts, 6th Edition, McGraw Hill, 2013.

[4] Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.