An Advanced Database Project

FOOD BANK MANAGEMENT SYSTEM

# **Abstract:**

In the present generation, people are wasting food, and food wastage is a crucial issue that is being increased day by day worldwide. Food preparation costs a lot of money and effort and if we waste the food all the money and effort will get wasted. This is a serious issue that needs to be prevented. We know that in our special events like parties, get-togethers, etc. most of the leftover food is being wasted. Not only in the events, but in every food centre, and restaurant a lot of food is being wasted, instead of throwing the food away, we can donate that to the less fortunate people who cannot afford it. To prevent food wastage, we create an application to collect the food, which is excessive in those events, and hotels and make it available to donate to the needy. By doing this we can reduce both food wastage and take care of less fortunate people and orphans.

# **Description**:

In this website, we collect and distribute the food in a request-and-response manner which means the person who wants to donate food requests the admin. Admin accepts the requests based on the availability conditions and distributes the food as a response to the request which has been received from the food recipients. In our project, we have three modules: Admin, Contributors, and Recipients. Admin will log in with default credentials. Admin will add the food contributors and can view the food contributors. Contributors log in with their credentials and view the food items that and already added by the admin. They can provide the quantity of food to donate and the expiry date and donate the food. Recipients will register and then log in with their credentials. The admin can view posted food from contributors, admin can collect the posted food from contributors, The admin can create meals and add the food items received from the contributors to the meals. Recipients can view the meals and request the quantity of melas they require. If they do not require the meal, they can cancel the request. The admin can accept or reject the food requests, and the status of the request will be changed accordingly.

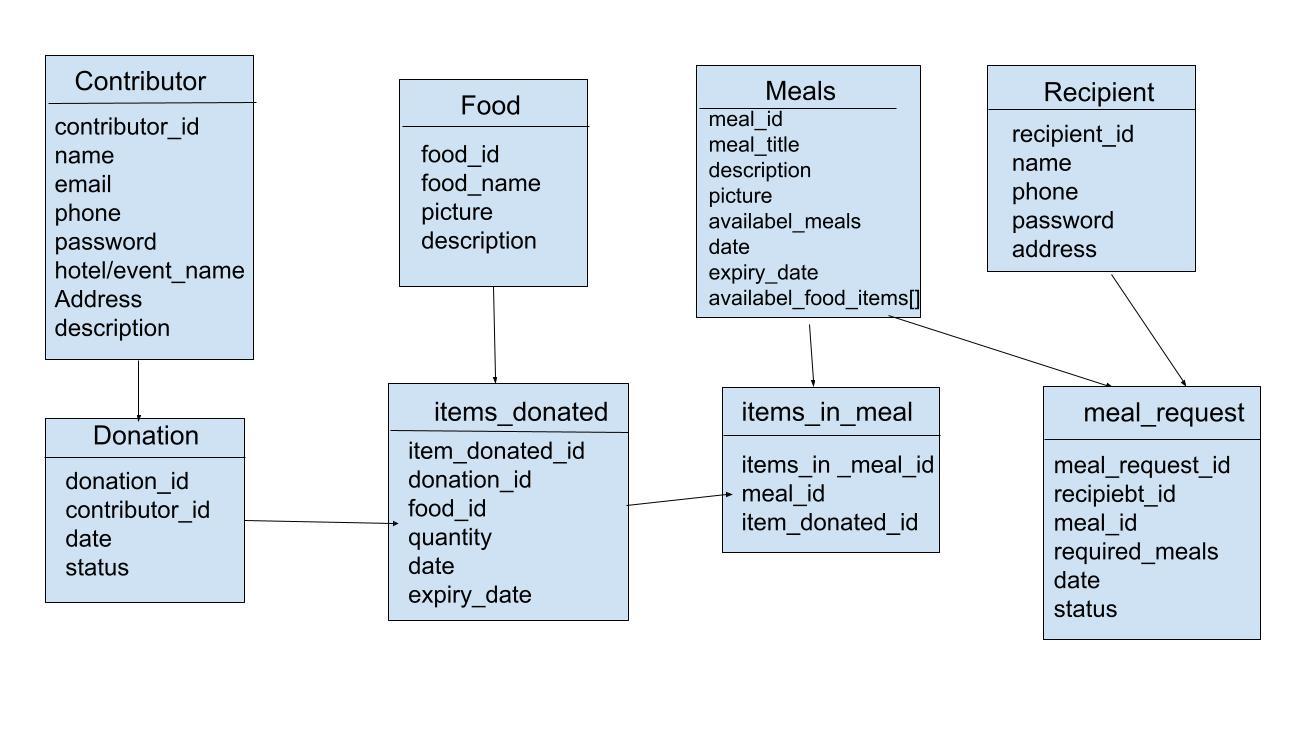
The restaurant should specify the count of available food. For example, 10 milk, 6 sandwiches, and 8 meats are available. The admin can combine them and make a meal of 5 with 5 milk, 5 meat, and 5 sandwiches, and the remaining food will be stored and combined with the next available food and made as a meal. The food is always categorized as meals.

Through our website, we can reduce food wastage, and help the needy. Food is a basic and essential need for every human being. It is our core value to help each other as humans and we fulfil this by providing food.

**Database Description:**

In this project, we have nine collections. The admin can add the food items details in the Food collection while giving the food name, picture, and description. The contributor can donate the food. Admin will add the details of the contributor and give them login credentials. These details are stored in the Contributor collection. Admin can activate and deactivate a contributor. After getting the authorization, the contributor can donate the food. These donation details are stored in the Donations collection. It contains date, status, and contributor\_id which is a foreign key, and it comes from contributor collection, here the status is “Donated to Admin”. When the contributor wants to donate the food they can choose the food item names are already added by the admin so these donated food details are stored in the item\_donated collection which contains the donation\_id, food\_id These two ids are foreign keys that come from the food collection and donation collection, and also collection contains food quantity, date and expiry date of the food. After these, the admin combines the food items received from the contributors and makes meals, so these are stored in the Meals collection. It contains the meal title, description, picture, available meals, date, expiry date, and availabel\_food\_items in an array. The admin can combine the donated food and create a meal pack by choosing the donated food items. These details are stored in the items\_in\_meal collection containing meal\_id and food\_id these are foreign keys that come from the Meal collection and Food collection. The recipient can register while giving the name, phone number, password, and address. These are all stored in the recipient collection. After the recipient login and send a request for meals to the admin, these request details are stored in the meal\_request collection containing recipient\_id and meal\_id these are foreign keys that come from the recipient collection and meal collection, required meals, date, and status. If the received request is accepted by the admin then the status is updated to “Request Accepted by Admin” or if the admin can reject the request then the status is updated to “Request rejected by Admin” These status changes are done in meal\_request collection.

# **Document Data Model:**



# **Database Collections:**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

# **Sample Data:**

**Admin:**

**Graphical user interface, text, application

Description automatically generated**

**Contributor:**

****

**Food:**

**Graphical user interface, text, application

Description automatically generated**

**Donation:**

**Graphical user interface, text, application

Description automatically generated**

**Item\_donated:**

**Graphical user interface, text, application

Description automatically generated**

**Meals:**

**Graphical user interface, text, application

Description automatically generated**

**Items\_in\_meal:**

**Graphical user interface, text, application, email

Description automatically generated**

**Recipient:**

**Graphical user interface, text, application

Description automatically generated**

**Meal\_request:**

**Graphical user interface, text, application, email

Description automatically generated**

# **Functional Requirements:**

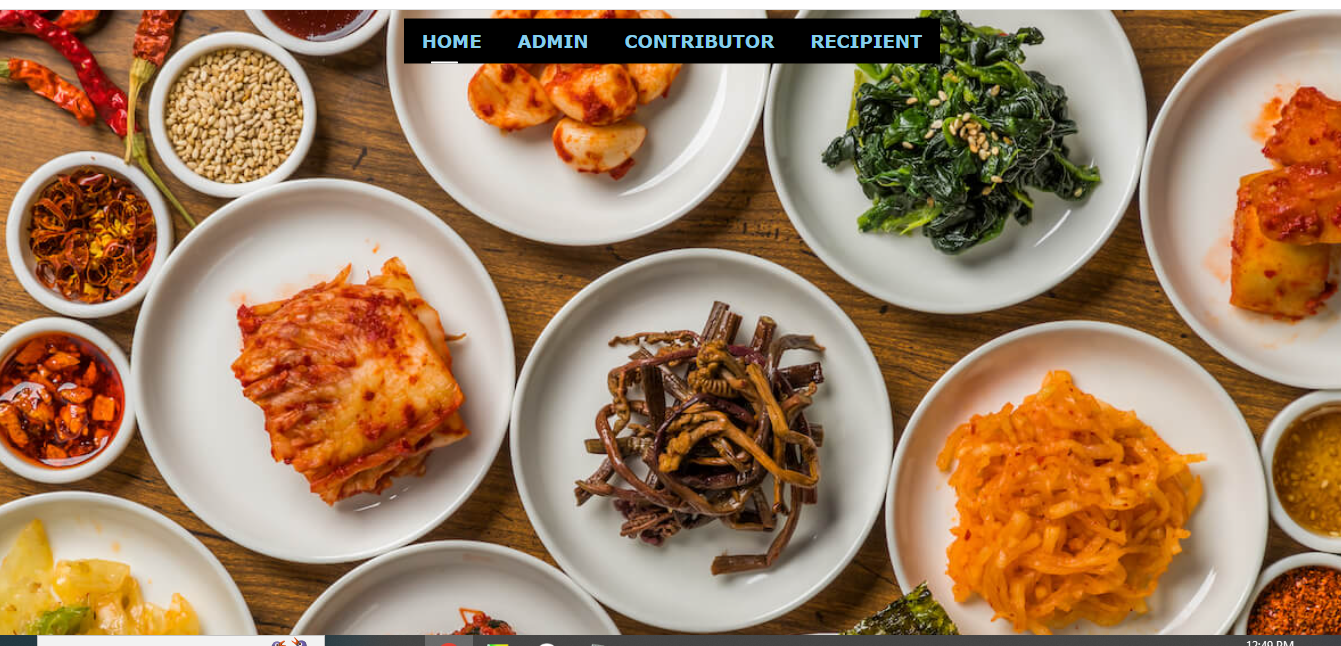
* Admin will access the website with login credentials.
* Admin can add and view contributors.
* Admin can add and view food on this website.
* Create meals from the food items received from contributors.
* Accept or reject the food request from the recipients.
* Contributors can log in to the website with login credentials and donate food to admin.
* The recipient also logs in with credentials.
* Before login recipients will get registered on the website.

# **User Interfaces and Forms:**

* **Add Contributer.html:** Admin will add contributors on the admin home pag**e.**
* **Add Food.html:** the admin will add the available food on the website.
* **Add food.html:** this code will add the food to the website.
* **Add meal.html:** Admin will add the meals with this code.
* **Add head.html:** the admin will add the head section.
* **Admin login.html**: the login page will be displayed here.
* **Admin home.html**: this code will add the homepage.
* **Contributor.html**: this is the contributor code.
* **Contributor head.html**: this code will write the contributor head section.
* **Contributor home.html:** After login the home page will display this code.
* **Crete meal.html:** this code will create the meal section.
* **Recipient.html:** this code will create the recipient.
* **Recipient home.html:** this code displays the home page of recipients.
* **Recipient login.html:** this page will display the login page.
* **Recipient registration.html:** this code will display registrations for the recipient
* **View donations.html:** this code will display donations.

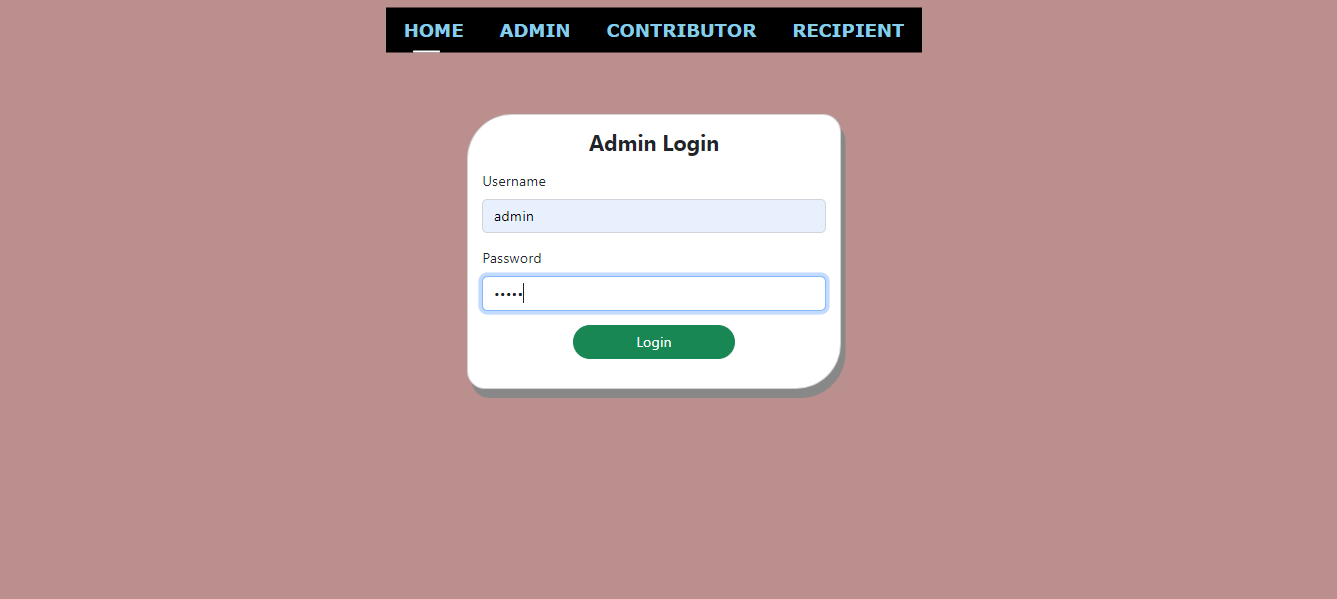
**Screen Shots:**

**Home Page:**

****

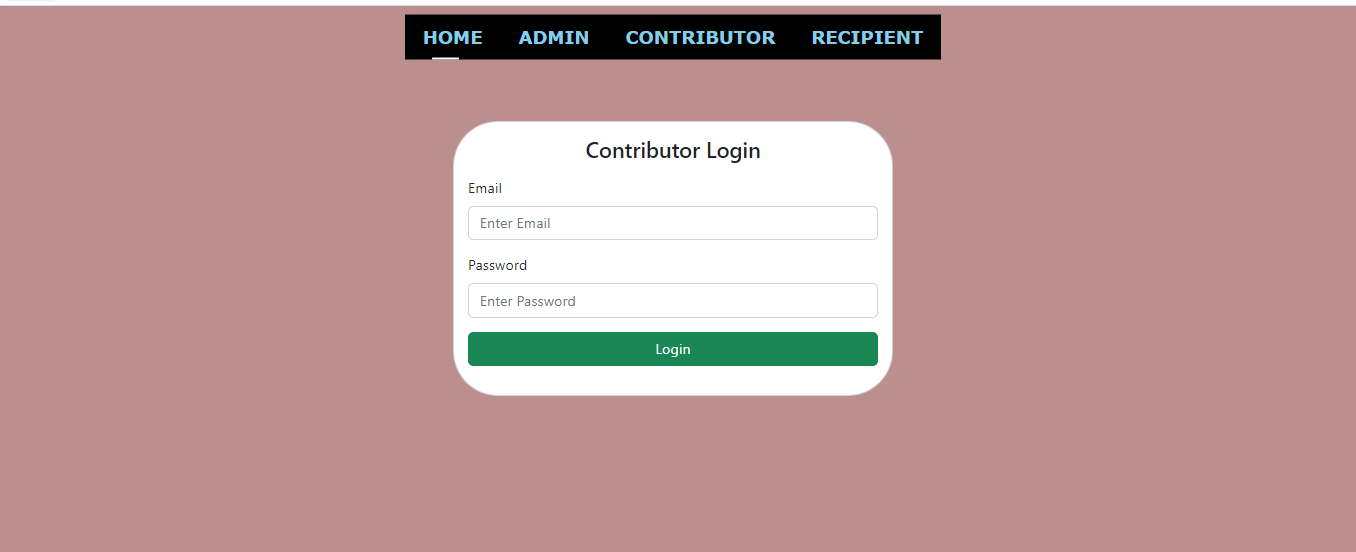
This is the homepage of the website, in which we have admin, contributor, and recipient.

**Admin Login:**

****

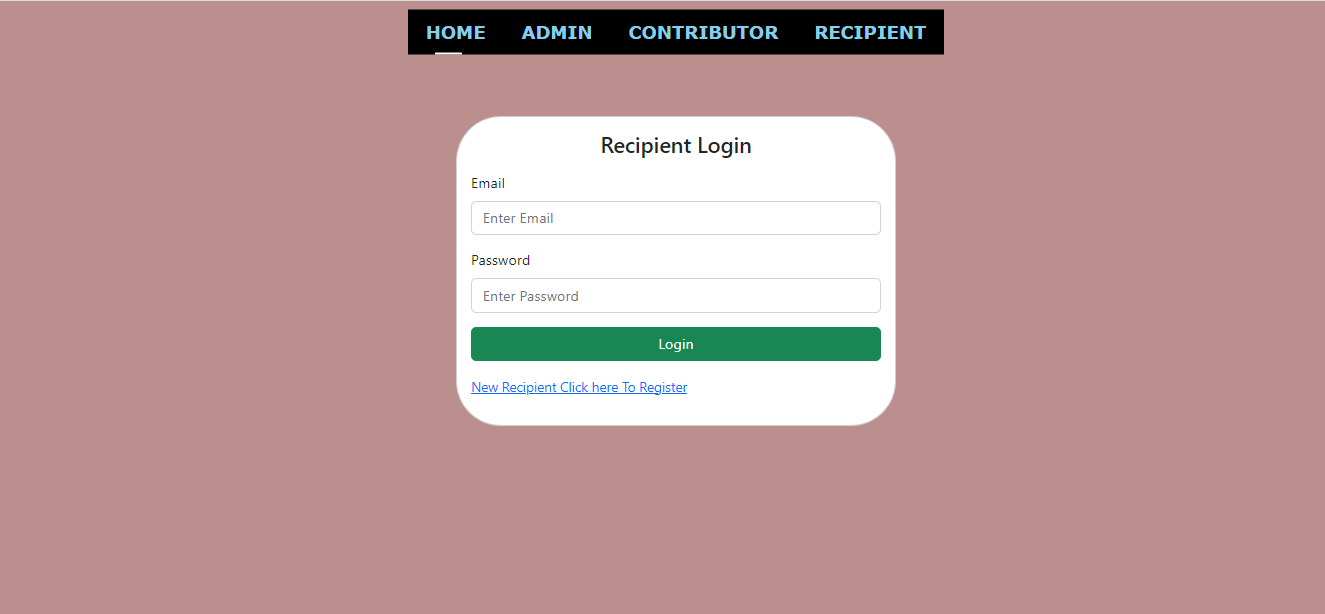
This is an admin login page, admin can log in to the page with default credentials, by clicking the login button the admin will enter the admin homepage.

**Contributor Login:**

****

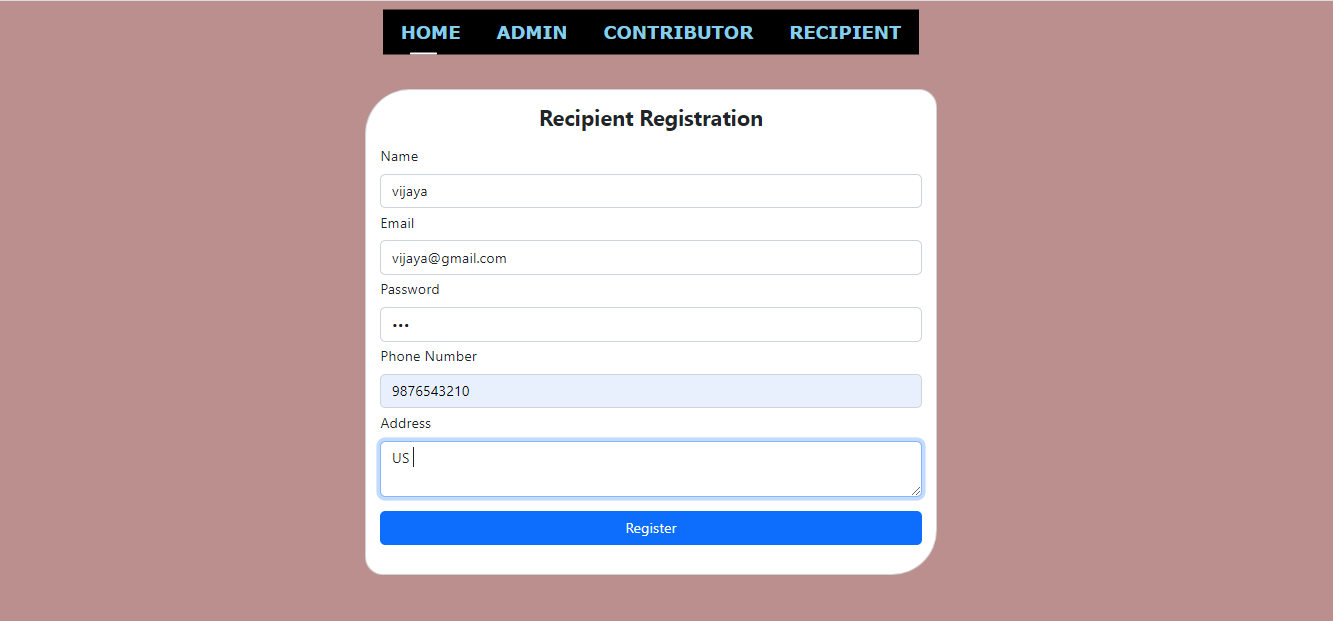
The contributor will log in to the website with login credentials, like email addresses and passwords.

**Recipient Login:**

****

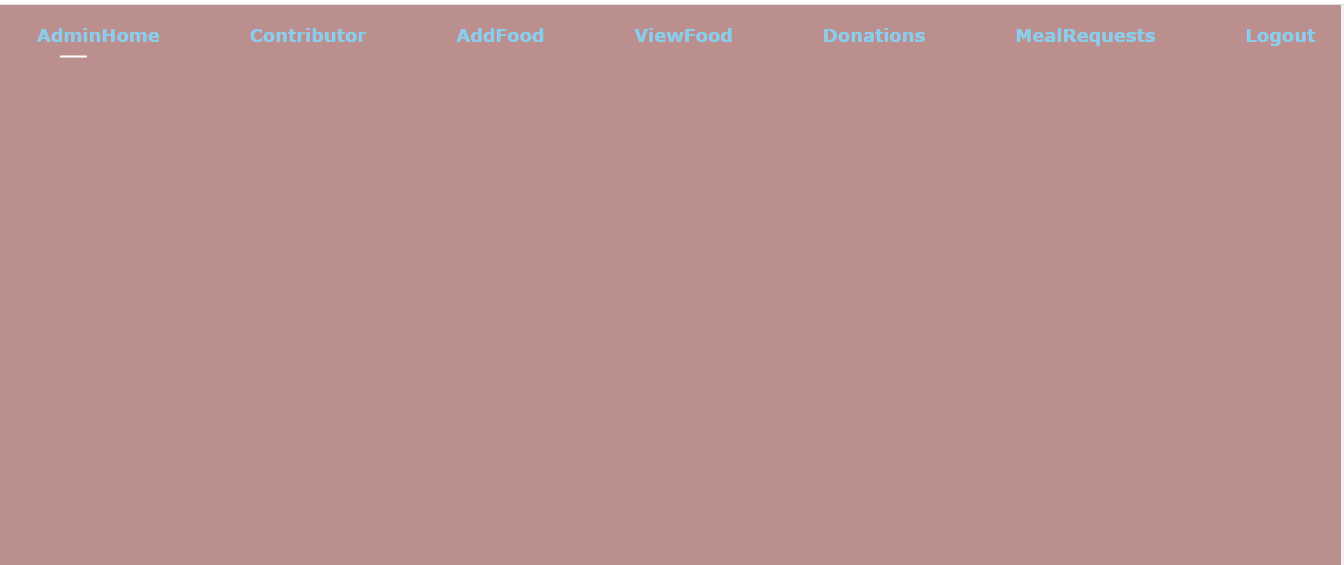
The recipients will log in to the website with login credentials, like email addresses and passwords.

**Recipient Registration:**

****

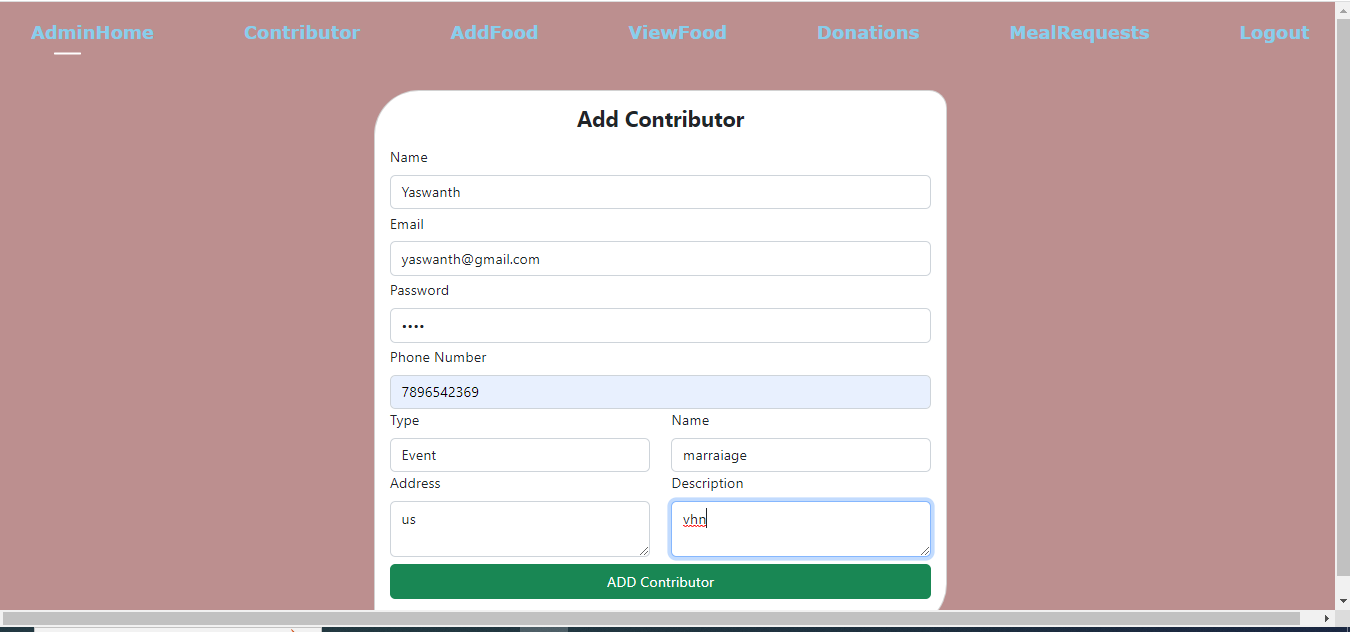
Before login to the website recipient must get registered on this page, by clicking the register button.

**Admin Home:**

****

After login the admin will get this page this page admin can perform these operations, adding contributors, adding food, viewing food, donations, and meal requests.

**Add Contributor:**

****

Admin will add the contributor by clicking the add button with their name address and email address, phone, password, etc...

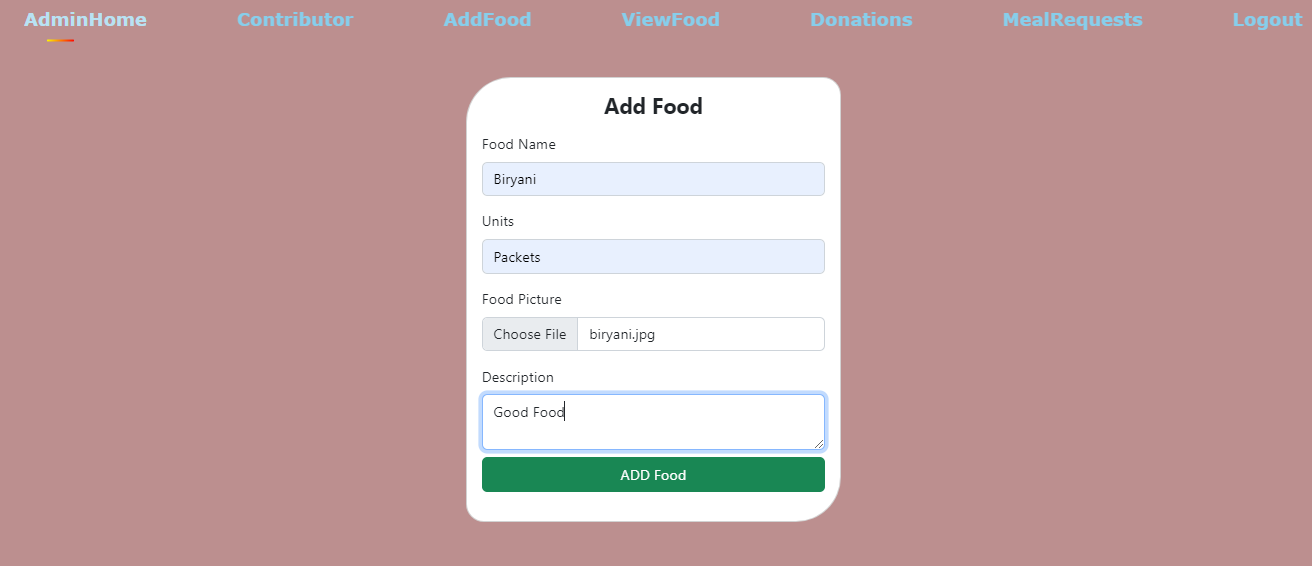
**View Contributors:**

**Graphical user interface, application, table

Description automatically generated**

The admin can view the contributor and can provide authorization to a contributor by clicking on Activate button. Once the contributor is active they login and donate the food.

**Add Food:**

****

On this page, the admin will add the food.

**View Added Foods:**

**Graphical user interface, application

Description automatically generated**

Admin can view the added food by clicking the view added food button.

**Contributor Home:**

****

After login, the contributor will reach this page and perform these operations like food and donations.

**Donate Food:**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

The contributor can view the various types of food items that are added by the admin. The contributor can select the quantity and expiry date and donate the food by clicking the Donate Now button.

**Donated Food:**

**Graphical user interface, application

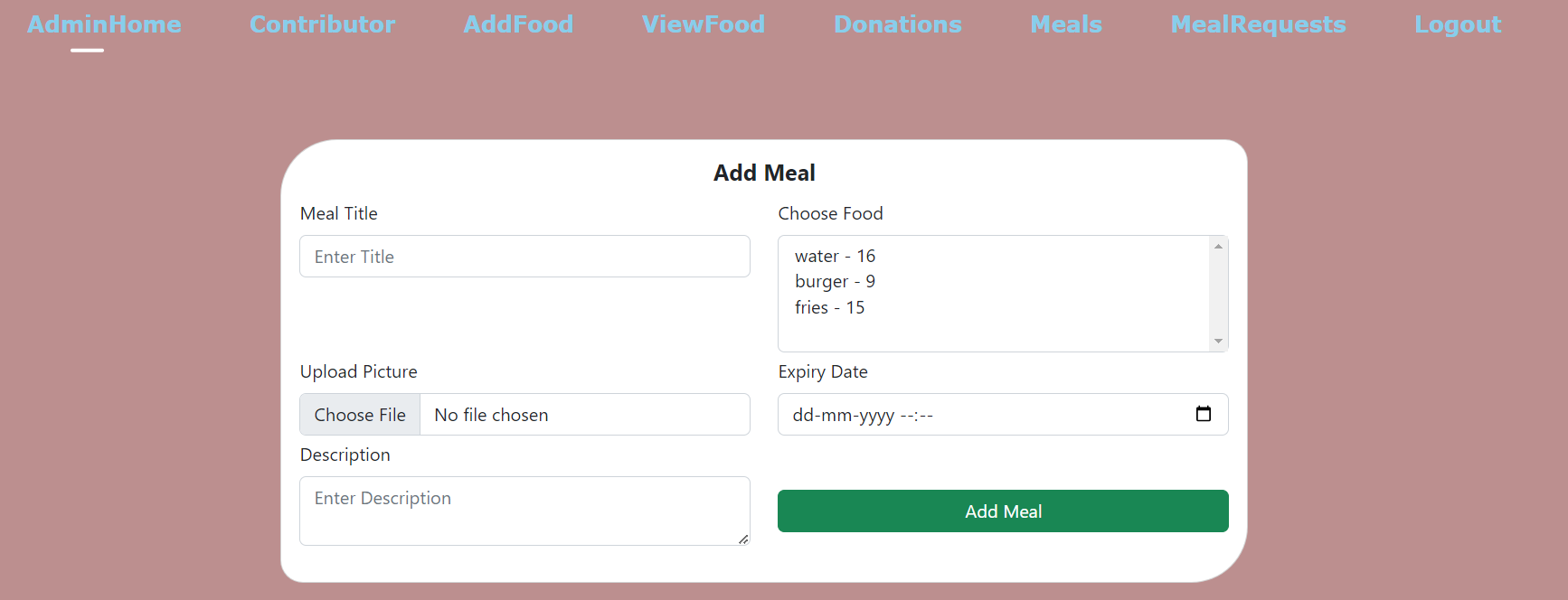
Description automatically generated**

On this page, the donated food will be viewed by contributors.

**Create and Add food to Meal:**

**A screenshot of a computer

Description automatically generated with low confidence**

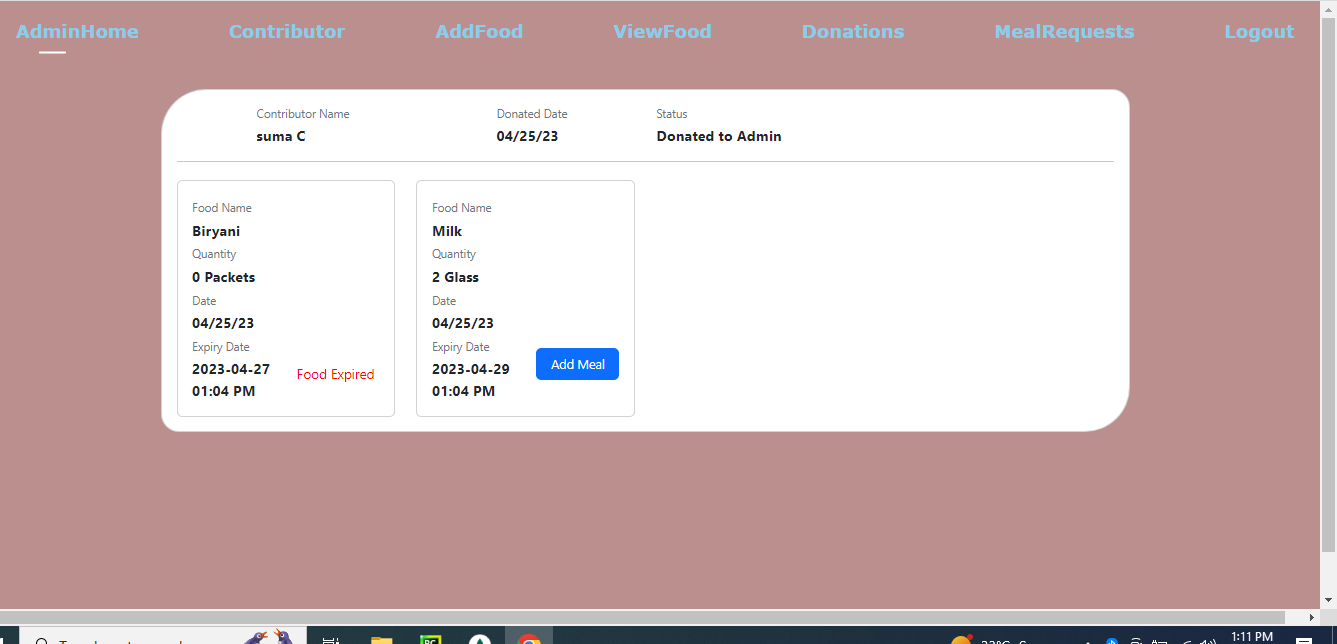
****

**Graphical user interface, text, application

Description automatically generated**

By clicking the Create Meal button a new form will be opened where the admin can select the list of items to create a meal. Number of plates will be automatically created based on the least count of the items selected. By clicking Add Meal button the admin can add the item to the meal.

**Food Expired:**

****

If the food is expired, it will show a red text telling Food Expired.

**Meal Request:**

**Graphical user interface, text, application

Description automatically generated**

The recipient can select the quantity of the Available meals and send a request to admin.

**Recipient View Meal Request:**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

The recipients can view the requests sent on this page, and if they don't need they will cancel the requests.

**Admin View Meal Request:**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

After recipients send the request, the admin can view the requests and accept or reject the request.

**Graphical user interface, text, application, chat or text message

Description automatically generated**

This page displays the status of each request.