



Give & Get

*A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Business Administration in MIS*

Prepared by

Shahad Yasseen Albannay	221416390
Shaima Ahmad Alhajri	221419858
Marwoh Abdulrahman Alshuaibi	221419123
Sokainh Moneer Altarouti	221426510
Fatimah Ali Almukhaydir	221415417

Supervised by

Dr. Yonis Gulzar

December 2024

ACKNOWLEDGEMENT

We would like to thank our instructor, Prof. Dr. Yonis Gulzar, for his invaluable support and guidance throughout the duration of this project. His expertise, encouragement, and willingness to share his knowledge have played a key role in shaping our understanding and enhancing the quality of our project and work. We are deeply honored to extend our gratitude to Prof. Dr. Yonis Gulzar for his exceptional guidance and assistance throughout every phase of our project. Prof. Dr. Gulzar's deep knowledge and expertise, passion for his field, and commitment to academic excellence have been nothing short of inspiring. His insightful feedback, thoughtful mentorship, and dedication to fostering our development have not only enriched our work but have also left a lasting mark on our academic path. We are grateful for the efforts and time he has invested in mentoring us. We feel incredible lucky to have such an amazing tutor and mentor like him, whose dedication to our learning has made all the difference. Thank you for being the incredible that you are.

UNDERTAKING

This is to declare that the project entitled “Give & Get” is an original work done by undersigned, in partial fulfillment of the requirements for the degree of “Bachelors of Business Administration” at Management Information Systems Department, School of Business, King Faisal University.

Further, it is hereby declared that the entire project work including the analysis, design and system development has been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Shahad Yasseen Albaanay

Shaima Ahmad Alhajri

Marwoh Abdulrahman Alshuaibi

Sukaina Moneer Altarouti

Fatimah Ali Almukhaydir

Note: sign across your name

ABSTRACT

This research paper explores the utilization of a website-based service aimed at collecting and distributing surplus food that has been converted into fertilizer. The service targets individuals, businesses, and farmers, addressing the prevalent issue of food waste resulting from factors such as overproduction and excessive purchasing. By offering the option to either transform surplus food into manure or feed it to animals, these services aid in food recycling efforts. Users can conveniently select their preferred service upon visiting the website. This innovative approach is designed to mitigate food waste, providing a convenient solution for disposing of surplus food while reducing environmental impact, particularly in terms of greenhouse gas emissions. The findings of the study illustrate the effectiveness and potential benefits of this service for enhancing various sectors, including agriculture, renewable energy, and public health, benefiting individuals, businesses, and farmers alike. By utilizing modern technology, this research paper highlights how complex challenges associated with food waste management can be addressed while also extracting value from waste streams.

Table of Contents

LIST OF TABLES	viii
LIST OF FIGURES	x
1. Introduction.....	1
1.1 <i>Problem Statement</i>	1
1.2 <i>Solution</i>	1
1.3 <i>Scope of Work/ Project Scope</i>	2
1.4 <i>Feasibility Study</i>	2
<i>Technical Feasibility</i>	2
<i>Financial Feasibility</i>	3
<i>Operational Feasibility</i>	4
<i>Environmental Feasibility</i>	4
1.5 <i>Functional Requirements</i>	4
1.7 <i>Data Collection Techniques</i>	6
<i>Social Media Monitoring</i>	7
<i>Survey</i>	7
1.8 <i>System Requirements</i>	10
<i>Hardware Requirements</i>	10
<i>Software Requirements</i>	11
1.9 <i>Technologies</i>	11
<i>Identifying problems, opportunities, and objectives</i>	11
<i>Determining human requirements</i>	12
<i>Analyzing the system's need</i>	12
<i>Designing</i>	12

<i>Developing and documenting software</i>	13
<i>Testing and maintaining the system</i>	13
<i>Implementing the system</i>	13
<i>1.10 Project Timeline</i>	13
2 System Analysis	15
2.1 <i>Context diagram</i>	15
2.2 <i>Functional Decomposition Diagram</i>	16
2.3 <i>Data Flow Diagram</i>	17
3 System Design	20
3.1 <i>Input and output forms design</i>	20
3.2 <i>Prototype design</i>	24
3.3 <i>Entity Relationship Diagram</i>	29
3.4 <i>Normalization</i>	30
3.5 <i>Logical Database</i>	32
3.6 <i>Physical Database</i>	33
4 Development.....	39
4.1 <i>Tools/ technologies selected for implementation</i>	39
4.2 <i>Web system navigation design</i>	40
4.3 <i>User Interface Design</i>	40
4.4 <i>Database connectivity</i>	49
5 Testing and Conversion	52
5.1 <i>Test Plan</i>	52
5.2 <i>Unit testing</i>	54
5.3 <i>Integration Testing</i>	77
5.4 <i>Conversion Plan</i>	81

6 System Documentation.....	82
<i>6.1 User Manual</i>	82
<i>6.2 Technical Support Manuals</i>	85
<i>6.2.1 General Troubleshooting</i>	85
<i>6.2.2. User Account Management</i>	86
<i>6.2.3. Notifications and Alerts</i>	86
<i>6.2.4. System Administration</i>	87
<i>6.2.5. Advanced Support</i>	88
<i>6.2.6. Contact Technical Support</i>	89
<i>6.3 Online Help and FAQs</i>	89
Major Contributions.....	92
References.....	96

LIST OF TABLES

Table 1.1 Hardware Requirements	10
Table 1.2 Software Requirements.....	11
Table 5.1 User Registration Case.....	54
Table 5.2 User Registration Failure Case	55
Table 5.3 User Registration Case.....	55
Table 5.4 User Registration Failure Case	56
Table 5.5 User Registration Case.....	56
Table 5.6 User Registration Failure Case	57
Table 5.7 User Registration Case.....	57
Table 5.8 User Registration Failure Case	58
Table 5.9 Admin sign in case.....	59
Table 5.10 Employee sign in case.....	59
Table 5.11 Client sign in case	60
Table 5.12 farmer sign in case	60
Table 5.13 Admin sign in case.....	61
Table 5.14 restaurant sign in case	61
Table 5.15 Posting Meals case.....	62
Table 5.16 Posting Meals failure	62
Table 5.17 Handling Requests case	63
Table 5.18 Handling Requests failure.....	63
Table 5.19 Meal Reservation case	64
Table 5.20 Meal Reservation failure.....	64
Table 5.21 Posting Leftover Food case.....	65
Table 5.22 Posting Leftover Food failure	65
Table 5.23 Purchasing Manure failure.....	67
Table 5.24 Posting Meals case.....	67
Table 5.25 Farmer Purchasing Manure failure	68
Table 5.26 Farmer Purchasing Leftover case	69
Table 5.27 Admin Managing Leftover Shop case	70
Table 5.28 Admin Managing Leftover Shop failure.....	71

Table 5.29 Admin Managing Manure Shop case.....	71
Table 5.30 Admin Managing Manure Shop failure	72
Table 5.31 Admin Managing Client Manure Orders case	73
Table 5.32 Admin Managing Client Manure Orders failure.....	73
Table 5.33 Admin Managing Farmer Manure Orders case	74
Table 5.34 Admin Managing Farmer Manure Orders failure.....	74
Table 5.35 Admin Managing Farmer Leftover Orders case	75
Table 5.36 Admin Managing Farmer Leftover Orders failure	75
Table 5.37 Admin Managing Daily Reports case	76
Table 5.38 Integration Testing	77
Table 5.39 Integration Testing	78
Table 5.40 Integration Testing	79
Table 5.41 Integration Testing.....	80

LIST OF FIGURES

Figure 1.1 responses of targeting population upon food wastage.....	8
Figure 1.2 responses of targeting population upon food wastage.....	9
Figure 1.3 responses of targeting population upon food wastage.....	9
Figure 1.4 responses of targeting population upon food wastage.....	10
Figure 1.5 responses of targeting population upon food wastage.....	10
Figure 1.6 Project Timeline using Gantt Chart.....	14
Figure 2.1 Context diagram of Give & Get website	15
Figure 2.2 Functional Decomposition diagram of Give & Get website	16
Figure 2.3 Data Flow diagram of Give & Get website	17
Figure 3.1 User's registration screen form	20
Figure 3.2 User's login screen form	21
Figure 3.3 Users' registration/login output.....	21
Figure 3.4 Adding to cart	22
Figure 3.5 Adding to cart details selection by customer.....	22
Figure 3.6 The output of adding to cart	23
Figure 3.7 payment selection details.....	23
Figure 3.8 output of a successful payment.....	24
Figure 3.9 Login/Signup page.....	24
Figure 3.10 Home page.....	25
Figure 3.11 About us page.....	25
Figure 3.12 Cart page.....	26
Figure 3.13 Products page.....	26
Figure 3.14 contact us page	27
Figure 3.15 Profile page.....	27
Figure 3.16 Previous order page	28
Figure 3.17 Entity Relationship Diagram	29
Figure 3.18 1NF Diagram	30
Figure 3.19 Logical Database	32
Figure 3.20 Give&Get Database.....	33
Figure 3.21 admin table	33

Figure 3.22 branch table	33
Figure 3.23 client table	34
Figure 3.24 client orders table	34
Figure 3.25 employee table.....	34
Figure 3.26 farmer table.....	35
Figure 3.27 food item table	35
Figure 3.28 manure table	35
Figure 3.29 non-profit organization table	36
Figure 3.30 restaurant table.....	36
Figure 3.31 donations table.....	36
Figure 3.32 shared food items table.....	37
Figure 3.33 leftover items table	37
Figure 3.34 leftover orders table	37
Figure 3.35 orders table	38
Figure 4.1 Web System Navigation Design.....	40
Figure 4.2 Portal Page.....	41
Figure 4.3 Home Page.....	41
Figure 4.4 Home Page.....	42
Figure 4.5 About Page	42
Figure 4.6 About Page	43
Figure 4.7 Manure Shop Page.....	43
Figure 4.8 Leftover Shop Page	44
Figure 4.9 Branch Page.....	44
Figure 4.10 Branch Page.....	45
Figure 4.11 Contact US Page.....	45
Figure 4.12 Help and FAQ Page.....	46
Figure 4.13 Settings Page	46
Figure 4.14 Restaurant Services Page.....	47
Figure 4.15 Nonprofit Organization Services Page	47
Figure 4.16 Nonprofit Organization Services Page	48
Figure 4.17 Client Services Page	48

Figure 4.18 Give and Get tables 50

1. Introduction

1.1 Problem Statement

Surplus food has several challenges that arise from households and restaurants, and in particular, these challenges impact the environment negatively. Food wastage arises from inefficient use of food such as overproduction, suboptimal harvesting methods, restrict cosmetic standards, for instance, when a supermarket rejects a food item based solely on its appearance, and buyer's behavior in households such as overbuying, poor planning of meals, and unawareness of expiry dates. One of the noticeable challenges that arise from food wastage is its harmful influence on ecosystems and contribution to climate change. Initially, the environmental effects of surplus food are depletion of natural resources and releasing greenhouse gases during its lifecycle [1]. Furthermore, the significant demand for energy and resources worsens pollution levels [1]. It is noticeable that wasted food does not only impact the environment during the disposal but also during the production stages.

Moreover, food wastage not only has a damaging consequence on the environment but also on food insecurity among vulnerable populations. The absence of an efficient system for redistributing wasted food from households and restaurants leads to missed opportunities of reducing food wastage levels and addressing hunger.

1.2 Solution

To address the issue of food waste and facilitate effective redistribution, we propose the development of a user-friendly online platform. This platform will serve as a central hub for collecting surplus food from both clients and businesses, which will then be redistributed to clients, farmers, or sold via our website. Following the collection process, wasted food will undergo conversion into fertilizer or repurposing as animal feed. Additionally, we will provide the option to donate untouched edible food from restaurants to individuals facing food insecurity. Such a website will save the environment and help recycling surplus food.

1.3 Scope of Work/ Project Scope

Our system's scope will begin with the city of Al-Ahsa, with the goal of eventually expanding to encompass all Saudi Arabia's areas. Our system will serve several parties, including restaurants and clients with surplus food, entities responsible for converting food into fertilizer or animal feed, clients, or organizations in need of manure or animal feed for their plants or animals, and nonprofit organizations that collect untouched food. We donate unused food from restaurants to these charitable groups that help individuals in need. Additionally, the technology will make it easier for users to schedule pick-ups and deliveries.

The project will not address topics that are not related to preserving surplus food and making use of it.

1.4 Feasibility Study

Technical Feasibility

System requirements include:

- Collection and transportation logistics.
- Converting food to manure or animal feed.
- Created a website to oversee food distribution and appointment scheduling.
Evaluating the technology entails assessing the accessibility of essential technologies. We require a robust server for our website to ensure seamless user experience with cloud services, prioritizing minimal latency. The server is rigorously secured to safeguard consumer data. Furthermore, we employ open-source Java, CSS, and HTML coding for front-end development. Data will be stored within a database management system, utilizing either MySQL or Oracle Live SQL.
- Scalability and compatibility of chosen solutions. Design a user-friendly interface that links individuals and restaurants, farmers on one platform, and sells fertilizer through our website, with an emphasis on making users feel at ease and providing easy access to icons.

Financial Feasibility

Cost Analysis:

Initial Setup Costs

1. Equipment:
 - Collection bins for surplus food storage.
 - Vehicles for food collection and delivery.
 - Equipment for converting food into fertilizer or feed.
2. Website Development:
 - Created an easy-to-use website for food distribution and scheduling.
 - Costs for hosting and domain registration.
3. Marketing
 - Raise service awareness through advertising and promotions.
 - Printing advertising items like flyers, brochures, and banners.

Ongoing operational expenses include:

1. Transportation:
 - Fuel costs for vehicles used in food collection and delivery.
 - Maintenance and repair expenses for transportation fleet.
2. Maintenance:
 - Regularly maintain equipment used in the food conversion process.
3. Staffing:
 - Salaries for operational staff responsible for food collection, processing, and distribution.

Fertilizer Sales:

1. Strategy Sales:
 - Conduct market research to identify current fertilizer pricing in Saudi Arabia.
 - Price the converted fertilizer competitively based on production costs and market demand.
2. Sales Volume:
 - Determine the potential demand for natural fertilizer in the target market.

- Forecast sales volume based on market demand and price strategy.

3. Revenue Forecast:

- To calculate revenue from fertilizer sales, multiply the predicted volume by the unit price.

Operational Feasibility

- Implementation Plan:
 - Phased rollout in Al-Ahsa.
 - Training for personnel and partners.
- Scalability:
 - Ability to extend operations to other locations.
 - Risk analysis involves identifying possible hazards and devising mitigation solutions.

Environmental Feasibility

- Environmental Impact Assessment:
 - Reduced greenhouse gas emissions through food recycling.
 - Improve soil quality through fertilizer usage.
 - Reduced dependency on synthetic fertilizers.
- Regulatory Compliance:
 - Adhere to local environmental rules.

1.5 Functional Requirements

Functional requirements are the set of capabilities and features that the system must have to collect surplus food and manage its transformation into either animal food or fertilizer effectively and efficiently.

- **Registration:** The system should enable users to register for purchases, notify about food availability, or order food donations through the website.
- **User interface:** The system must enable users to input and review their profile information, select fertilizer distribution times, and provide details on surplus food, including quantity and type of fertilizer.

- **Database:** The system should offer a database for storing and analyzing data concerning food that has been transformed, utilized, or donated, encompassing details such as its condition, source, and collection time.
- **Reporting:** The system should produce reports on the surplus food gathered, its utilization as animal feed, the quantity of fertilizer produced, and sales figures, as well as the food that has been donation to facilitate the preservation and analysis of records.
- **Scheduling:** The system should allow authorized users to schedule the receipt and transportation of surplus food to farms, fertilizer production facilities, or to fertilizer purchasers.
- **Notification:** The system should offer a notification feature to alert registered users of the availability of surplus food and manure for sale, or food availability for donation.
- **Compliance:** The system must guarantee adherence to food safety regulations and environmental guidelines at every stage of the collection, transformation, and utilization processes.

1.6 List of Staff's Functions

The staff's responsibility and rules functions are the backbone of the success of the GIVE & GET system. Each role has its own duties and expectations which together form a capable group to seek the system's goals.

1. Clients:

- Register and login to the system.
- Order manure.
- Share surplus food.

2. Restaurants:

- Register and login to the system.
- Notify about food availability.
- Donate food
- Confirms donations.

3. Non-profit Organizations:

- Register and login to the system.
- Order food donations.

4. Manure:

- Shared surplus food that transform into manure/fertilizer.
- Offered for sale on the website.

5. Farmers:

- Register and login to the system.
- Order manure and surplus food.

6. Administrator:

- Manages users.
- Reviews and modifies food and manure listings.
- Provides technical support.
- Resolves disputes.

1.7 Data Collection Techniques

The foundation of every market research is data collection. This phase starts with the establishment of the study goals and concludes upon the acquisition of the data for analysis.

We chose two methods for collecting data: the first is social media monitoring and the second is surveys.

Social Media Monitoring

A method of gathering data that is comparable to transaction tracking is social media monitoring. But this kind of technology tracks a customer's social media history rather than their transaction history[2]. This is used by several platforms and businesses to monitor users' interactions with various online content in order to get insight into the goods and services that users may find interesting as well as the things that matter to them. We will be able to identify our target market and comprehend their interests from this position.

The result of Social Media Monitoring:

We noticed during our monitoring of social media sites that there are many people who want to benefit from surplus food and not throw it away without benefiting from it, especially in our Saudi society.

Survey

One of the most popular ways to gather data is through surveys. Both quantitative and qualitative data may be collected using questionnaires, which can be administered both digitally and physically [5]. Surveys are an accessible choice because these questions are generally cheap to develop and respond to. Subsequent research, decision-making matrices, or product enhancement might be based on the replies gathered [2].

The Questions of the Survey

- 1- Are you currently facing challenges related to surplus food or food wastage in your household or business?
- 2- Have you ever considered alternative methods for managing surplus food, such as converting it into fertilizer or animal feed?
- 3- Do you believe there is a need for a systematic approach to manage food wastage in your community?
- 4- Do you believe it's a good idea to combine profit generation with environmental protection through initiatives like converting surplus food into fertilizer or animal feed?

- 5- Would you use a website that addresses food wastage by collecting surplus food, repurposing it into fertilizer or animal feed, and redistributing it to individuals, farmers, or selling it?

The Results of the Survey

We collected 105 survey responses, with the breakdown of results for each question as follows:

- **Question one:** 80 responses were affirmative (yes), and 24 were negative (no).
- **Question two:** 60 responses were affirmative (yes), 24 responses were negative (no), and 20 responses were uncertain (maybe).
- **Question three:** 91 responses were affirmative (yes), 8 responses were negative (no), and 5 responses were uncertain (maybe).
- **Question four:** 99 responses were affirmative (yes), and 5 responses were negative (no).
- **Question five, the final question:** 82 responses were affirmative (yes), and 22 responses were negative (no).

The results of the Survey generally express the community's need for the services that our site will provide, the need to benefit from surplus food, and their interest in it.

Figures from 1.1 to 1.5 show the responses of Al-Ahsa population in terms of interest in having redistributing system of food wastage.



Figure 1.1 responses of targeting population upon food wastage

Have you ever considered alternative methods for managing surplus food, such as converting it into fertilizer or animal feed?

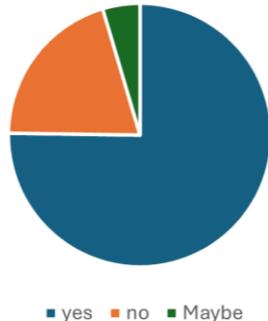


Figure 1.2 responses of targeting population upon food wastage

Do you believe there is a need for a systematic approach to manage food wastage in your community?

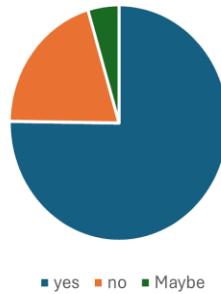


Figure 1.3 responses of targeting population upon food wastage



Figure 1.4 responses of targeting population upon food wastage



Figure 1.5 responses of targeting population upon food wastage

1.8 System Requirements

Hardware Requirements

Table 1.1 Hardware Requirements

Hardware	The requirements
Devices	Laptop / machine that convert food to fertilizer
Processor	Intel Core i7
Memory	-8 ram - 1 tera internal memory
Network infrastructure	Router
Internet access	5G - Fiber internet.

Software Requirements

Table 1.2 Software Requirements

Software	The requirements
Operating system	-Microsoft Windows, MacOS, Android and Apple - Using these four operating systems will facilitate for restaurants, normal people, farmers and agriculture stores to receive and provide surplus food and fertilizer.
Web server	<ul style="list-style-type: none">• Google cloud server for website• Oracle SQL for database
Programing languages	HTML, PHP and JavaScript
Design tools	CSS and Adobe XD to design website interface
GPS and map software	Google and Apple maps

1.9 Technologies

We will need a machine that converts food into fertilizer, as well as large bags for distributing it to farmers and stores specializing in agricultural tools. Additionally, a sizable inventory with sufficient storage capacity will be required to store all the fertilizer and food products, each labeled with information about the type of food, whether it is fertilizer or food, quantity, and expiration date. Moreover, the main website will require a robust server to ensure smooth operation and handle data storage, along with a database system to store all order information, chat logs, and personal information of farmers and stores. Furthermore, a dashboard will be necessary for analyzing user behavior and identifying areas for website improvement. Finally, we will establish a contract with a logistics company to handle product delivery and receipt.

Identifying problems, opportunities, and objectives

Using social media and questionnaires, we identified a pressing societal need for a platform like ours. The primary challenge we uncovered was the inefficiency and inconvenient nature of the current manual process of collecting food waste. This process is not only labor-intensive but also sensitive to errors, resulting in significant food loss. Our

website aims to address these challenges by simplifying and optimizing the process, thereby reducing food waste across Saudi Arabia, with a particular focus on Al-Ahsa. Given the abundance of farmers in this region, it presents an ideal hub for effectively repurposing food waste in a more sustainable manner.

Determining human requirements

To determine the human requirements, we used a questionnaire and interviews to make the website easier for all categories of society because it is a website from society to society. We used a Google form for a questionnaire and sat with farmers, restaurants managers and store of agriculture tools to find out what they exactly need and how we can make the website easier for them and user friendly.

Analyzing the system's need

We employed a comprehensive approach, utilizing three distinct diagrams: the context diagram, the data flow diagram (DFD), and the Entity-Relationship Diagram (ERD). These diagrams served to carefully and thoroughly outline the various processes, inputs, and outputs involved in our system. Moreover, they elucidated the complex interactions between different entities, describing the permissions and restrictions governing each entity's actions. This methodical analysis provided invaluable insights into the system's architecture, facilitating a clear understanding of its functionality and enhancing our ability to optimize its performance.

Designing

We utilized UML diagrams as our primary tool to model and visualize the system's architecture and functionality [4]. These diagrams were meticulously crafted using Lucidchart, FigJam and Draw.io, enabling us to create detailed representations of the logical design and the overall website framework. This approach facilitated effective communication among team members and stakeholders, ensuring alignment on the design elements and providing a solid foundation for the development process.

Developing and documenting software

- Server development tool (Oracle SQL, Google Clouds, PhpMyAdmin)
- Programming development tool (visual studio, HTML editor)
- Web Page format (CSS)
- Designing logs, graphs and icons (Canva, Adobe Photoshop, Adobe illustrator)
- Images (Pixel website)

Testing and maintaining the system

We rigorously conducted website bug testing utilizing Powermapper.com to ensure optimal functionality and seamless user experience. Continuous maintenance will be carried out through timely updates, addressing any newly identified errors or bugs, as well as implementing enhancements to cater to the evolving needs of both users and managers. This proactive approach to maintenance guarantees the sustained efficiency and effectiveness of the website, fostering long-term satisfaction and usability.

Implementing the system

Once the website garners positive recommendations from users, it will undergo thorough preparations for publication across multiple platforms, ensuring compatibility with various operating systems such as iOS for Apple iPhones, Android for Samsung and Xiaomi devices, as well as MacOS and Windows operating systems. This comprehensive approach to platform integration guarantees accessibility to a diverse range of users, maximizing the website's reach and impact across different devices and operating environments.

1.10 Project Timeline

Figure 1.10.1 shows the timeline of completing process of this project.



Figure 1.6 Project Timeline using Gantt Chart

2 System Analysis

2.1 Context diagram

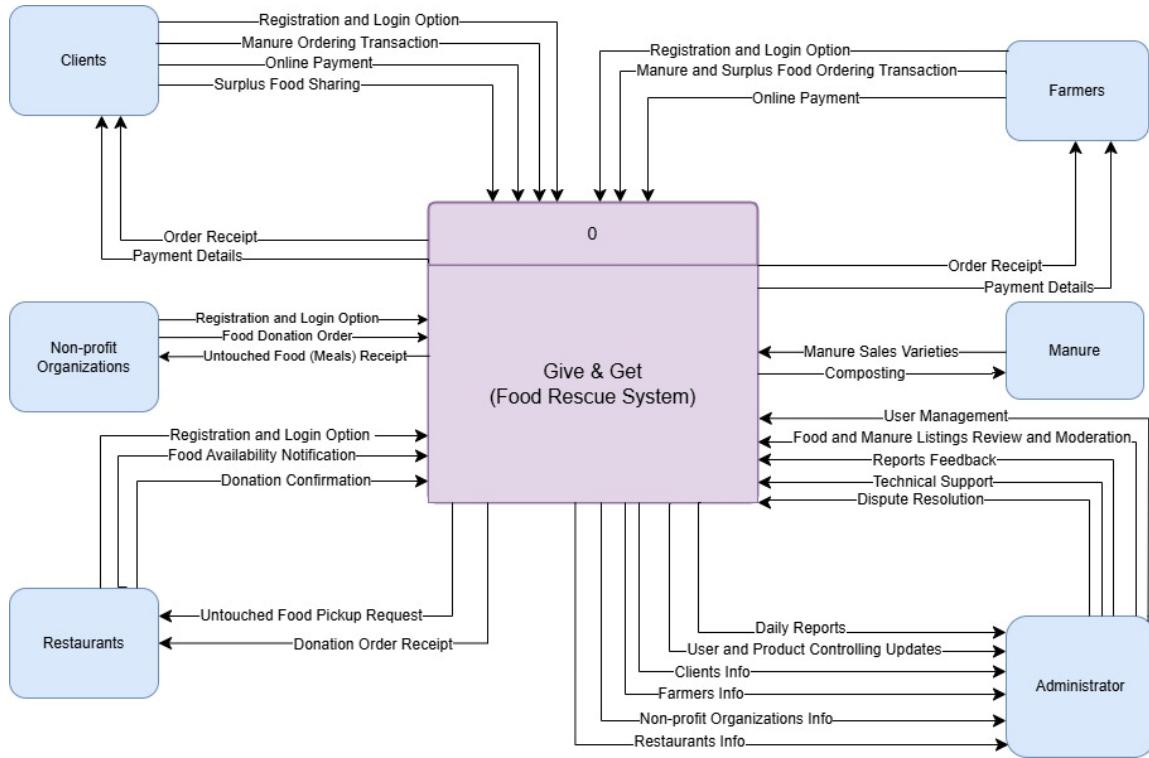


Figure 2.1 Context diagram of Give & Get website

Figure 2.1 shows the context diagram of Give & Get website. This context diagram clarifies what each user can do in the system:

Administrator: Manages users and has access to their information, supports technical operations, resolves disputes, reviews and moderates' food and manure listings, receives users' information once they register, receives daily reports, and receives user and controlling updates.

Clients: Register, log in, order manure, pay online, share surplus food, and receive order and payment details.

Manure: Surplus food that has been shared by the clients will be transformed into manure (compost); after composting leftover food, manure will be offered for sale on the website.

Farmers: Register, log in, order manure and surplus food, pay online, and receive order and payment details.

Restaurants: Register, log in, receive order for donating (untouched) meals, and send a notification for food availability, and confirms donation.

Non-profit organizations: Register, log in, order food donation and receive the donation they ordered.

2.2 Functional Decomposition Diagram

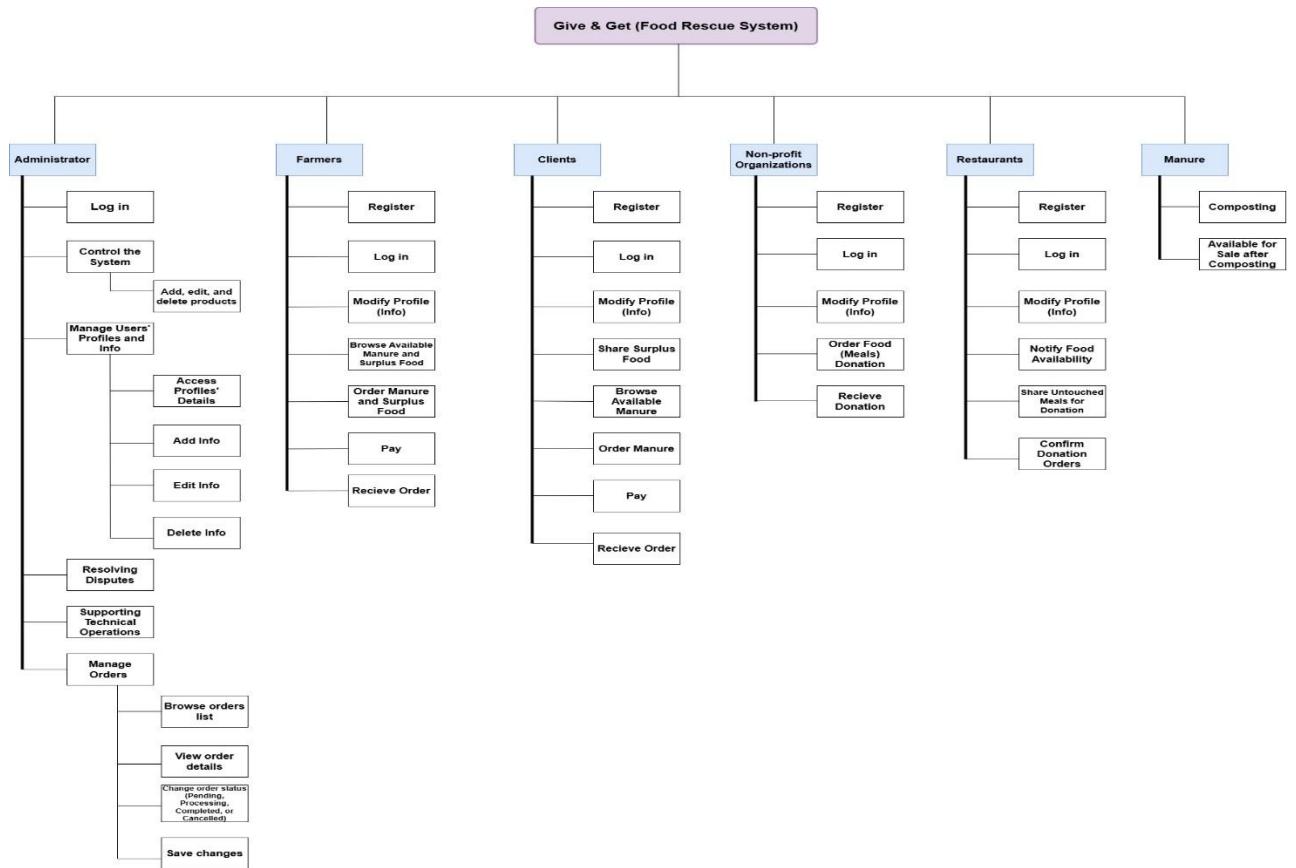


Figure 2.2 Functional Decomposition diagram of Give & Get website

Figure 2.2 shows the Functional Decomposition Diagram of Give & Get website, outlining all systems functions along with the required sub-functions needed to complete the main functions. The administrator controls the system, manages users' profiles and information which he or she can add, edit, and delete products or information. Also, the administrator can resolve disputes, and support technical operations. Both clients and farmers can register, log in, modify their profiles; however, farmers can order and browse both surplus food and manure, but clients can only order and browse manure, they both also are able to pay and receive orders. After receiving surplus food from clients, it will be transformed into manure or sold as is. Moreover, non-profit organizations can register, log

in, modify their profiles, order food donations, and receive them. Restaurants, like other users, can register, log in, modify their profile, and share untouched food (meals) for donation after notifying the system about its availability, and accept or reject donation requests.

2.3 Data Flow Diagram

Figure 2.3 shows Data Flow diagram of Give & Get website

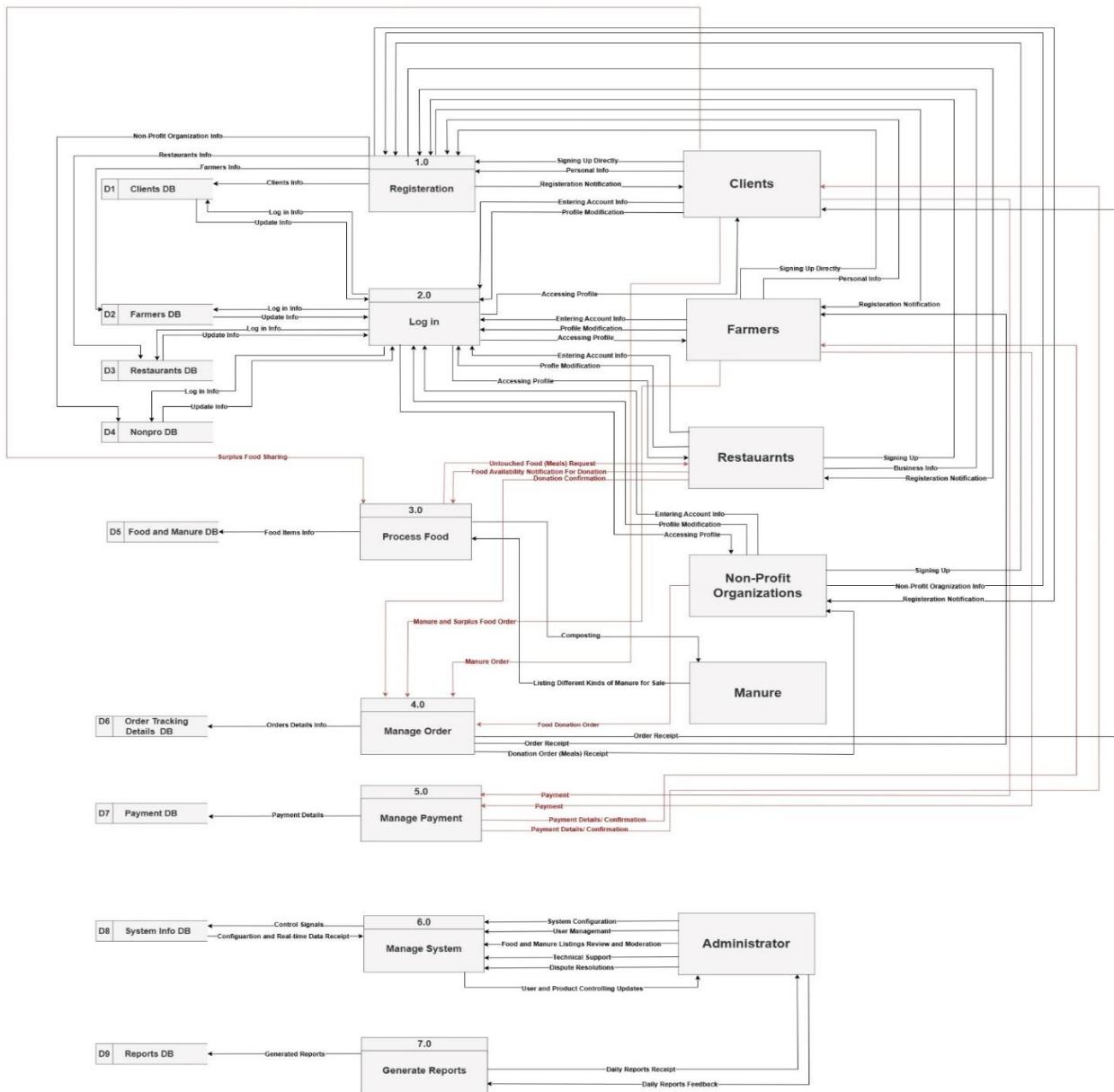


Figure 2.3 Data Flow diagram of Give & Get website

Level zero diagram displays the processes in detail and shows how they're connected with each entity:

1. Register:

- Users register by filling out their required information to create new accounts.
- New user accounts will be stored in the user database to process and save their information.

2. Log in:

- After successfully creating a new account, users must log in to their account to make any orders. Additionally, users can view their account information and make modifications to it.
- Log-in details will be stored in the user database to process and update information.

3. Process Food:

- This process involves composting leftover food that has been shared by clients into a manure and listing the manure after composting for sale on the website.
- It also involves requesting restaurants to check if there is available food for donation to non-profit organization, and they should notify the system if there are any available meals.

4. Manage Order:

This process involves users who can order surplus food and manure such as clients and farmers, and the non-profit organizations that order food donations.

5. Manage Payment:

- Only users who have ordered manure or surplus food must complete their payment, which will be processed and stored in the payment database to issue payment confirmation. Accreditation of paid orders will then commence. Excluded from payment are non-profit organizations, restaurants, and clients who share their surplus food with our system.

6. System Management:

- Administrators access the system information database using tools to manage users, products. They have access to all the reports, monitor and confirm orders, and make any necessary system configurations. They are also responsible for technical support and dispute resolution. During this process, they will receive updates on any control changes made to user accounts or products, as well as configuration statuses.
- Status updates and control signals will be stored in the system information database to process and issue configuration and real-time data receipts.

7. Generate Daily Reports:

- Administrators will receive reports daily, and they will receive a report PDF document. Generated and modified reports will be stored in the reports database.

3 System Design

3.1 Input and output forms design

- Sign-up/Login

Input:

- In Figure 3.1, registration page will pop up when users press on the register tab, then the users have to specify what kind of users they are (clients, farmers, restaurants, and non-profit organizations).
- After registering, as shown in Figure 3.1.2, on the website, users will log in easily. After logging in, the user will be redirected to the home page, as shown in Figure 3.3

The image shows a registration form for a platform called "GIVE&GET". At the top left is the logo, which consists of two orange interlocking circles. To the right of the logo are three links: "ABOUT" and "CONTACT". Below these are two large, rounded rectangular buttons: "LOG IN" on the left and "REGISTER" on the right. The "REGISTER" button is highlighted with a darker shade. The form itself has several input fields: "FIRST NAME", "LAST NAME", "EMAIL", "LOCATION", "MOBILE NUMBER", "PASSWORD", and "CONFIRM PASSWORD". Below these fields is a dropdown menu labeled "USER TYPE" with the option "CLIENTS" selected. There is also a small checkbox labeled "REMEMBER PASSWORD". At the bottom of the form is a large, rounded rectangular "SIGN UP" button.

Figure 3.1 User's registration screen form

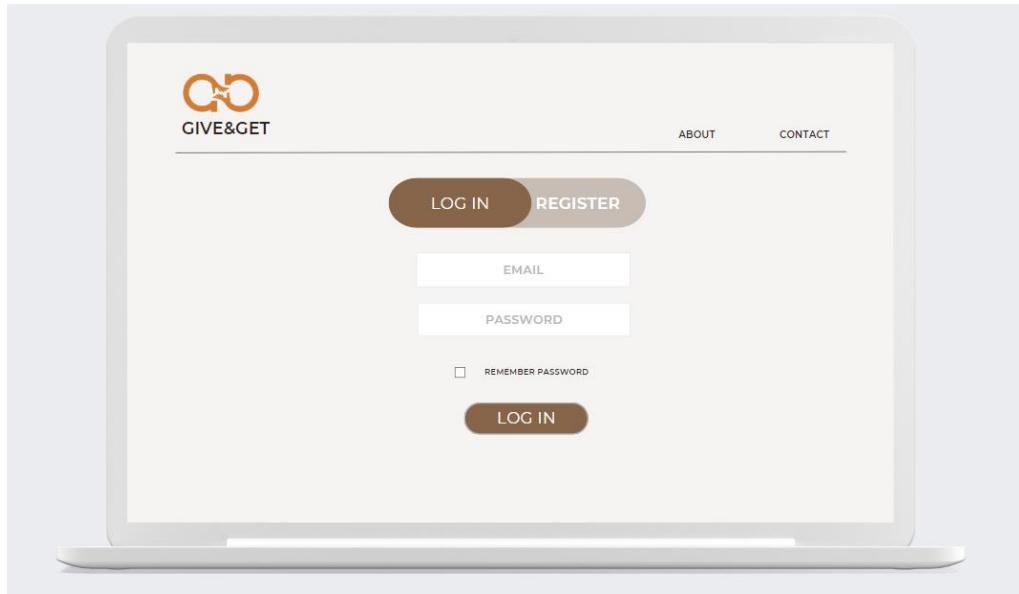


Figure 3.2 User's login screen form

Output:

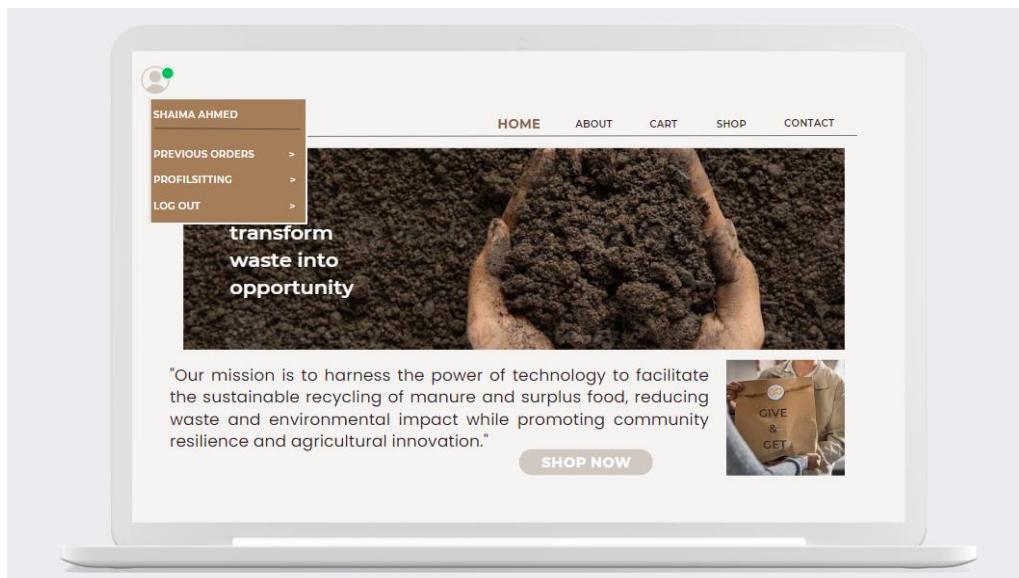


Figure 3.3 Users' registration/login output

- Add to cart (see Figure 3.4 & Figure 3.5):
- Users, based on their type, can add to cart the items they want to order, after selecting items they want to order, they will be redirected to the checkout page, Figure 3.1.6, to complete the checkout process.

Input:

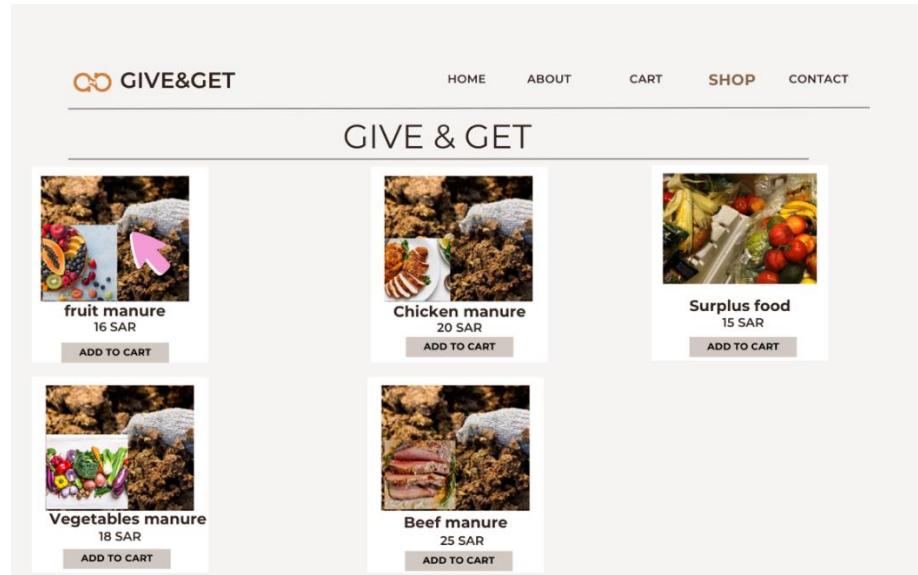
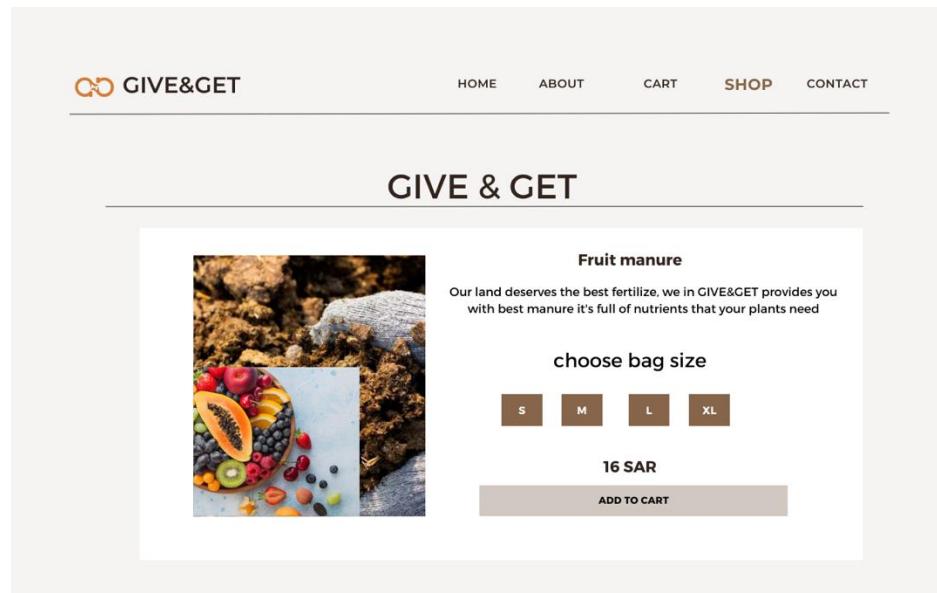


Figure 3.4 Adding to cart



(a)

Figure 3.5 Adding to cart details selection by customer

The screenshot shows a shopping cart page for 'GIVE&GET'. At the top, there's a navigation bar with links for HOME, ABOUT, CART, SHOP, and CONTACT. Below the navigation is a table with four columns: Item, Price, Quantity, and Total. A single item is listed: a small yellow bag with a green logo, priced at 10 SR, with a quantity of 1. The total is 10 SR. Below the table, there are fields for Subtotal (10 SR), Sales Tax (15 SR), and a Coupon Code input field with a 'Add Coupon' link. The Grand Total is displayed as 15 SR. A message at the bottom says 'CONGRATS,, YOU'RE EIGIBLE FOR FREE SHIPPING' with a delivery truck icon. A 'Check out' button is at the bottom right.

(b)

Figure 3.6 The output of adding to cart

- Paying (see Figure 3.7)
- After completing payment process, Figure 3.1.6, the user will be shown a successful payment page.

The screenshot shows a payment selection details page on a laptop. On the left, there's a form for 'BILLING ADDRESS' with fields for Full Name (YOUR NAME), Email (EMAIL@example.com), Address (ADDRESS), City (CITY), State (STATE), Zip (ZIP), Exp Year (EXP YEAR), and CVV. There's also a checkbox for 'SHIPPING ADDRESS SAME AS THE BILLING'. On the right, there's a 'PAYMENT' section showing 'ACCEPTED CARDS' (VISA, MasterCard, American Express, Discover) and a 'CREDITCARD NUMBER' field containing 1111-2222-3333-4444. To the right of the address form is a 'CART' summary showing two items: ITEM 1 (5 SR) and ITEM 2 (10 SR), totaling 15 SR. A 'CONTINUE TO CHECKOUT' button is at the bottom.

Figure 3.7 payment selection details

Output:

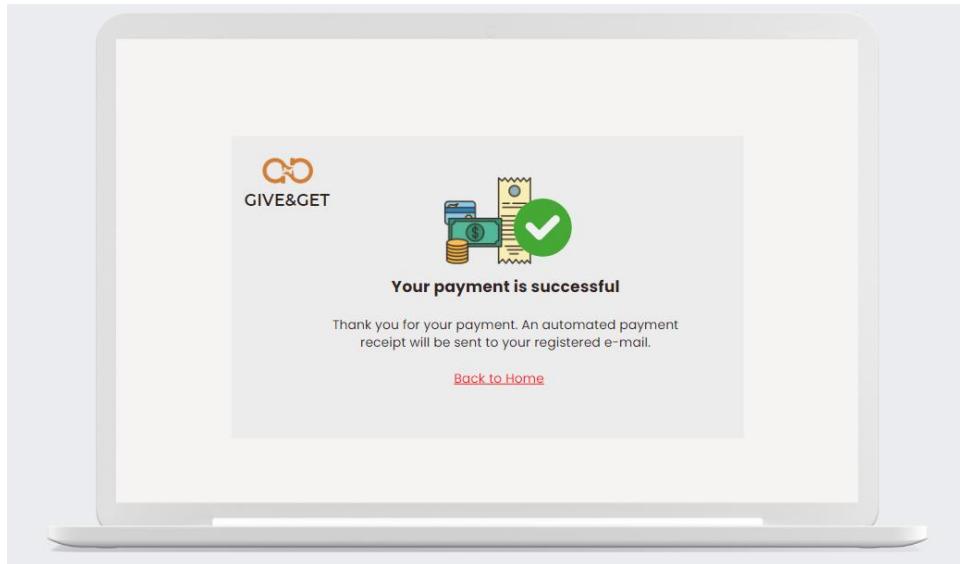


Figure 3.8 output of a successful payment

3.2 Prototype design

Figure 3.2.1 to Figure 3.2.7 a prototype showing the basic user interface design covering the major functions of Give & Get website.

The image shows a web page for "GIVE&GET". At the top left is the "GIVE&GET" logo. To the right are links for "ABOUT" and "CONTACT". Below the header is a registration form. It has two buttons at the top: "LOG IN" (disabled) and "REGISTER" (highlighted). The registration form consists of seven input fields: "FIRST NAME", "LAST NAME", "EMAIL", "LOCATION", "MOBILE NUMBER", "PASSWORD", and "CONFIRM PASSWORD". Below these fields is a small checkbox labeled "REMEMBER PASSWORD". At the bottom right of the form is a brown "SIGN UP" button.

Figure 3.9 Login/Signup page

Input name: Register

Who will use it: clients, farmers, restaurants, and non-profit organizations

Why is it used: it is used to identify users.

Description: this page will pop up when users press on the register tab, then the users have to specify what kind of users they are (clients, farmers, restaurants, or non-profit organizations).

As shown in Figure 3.10, the user will see the basic user interface design after logging in, and will discover the various functions available on the website.

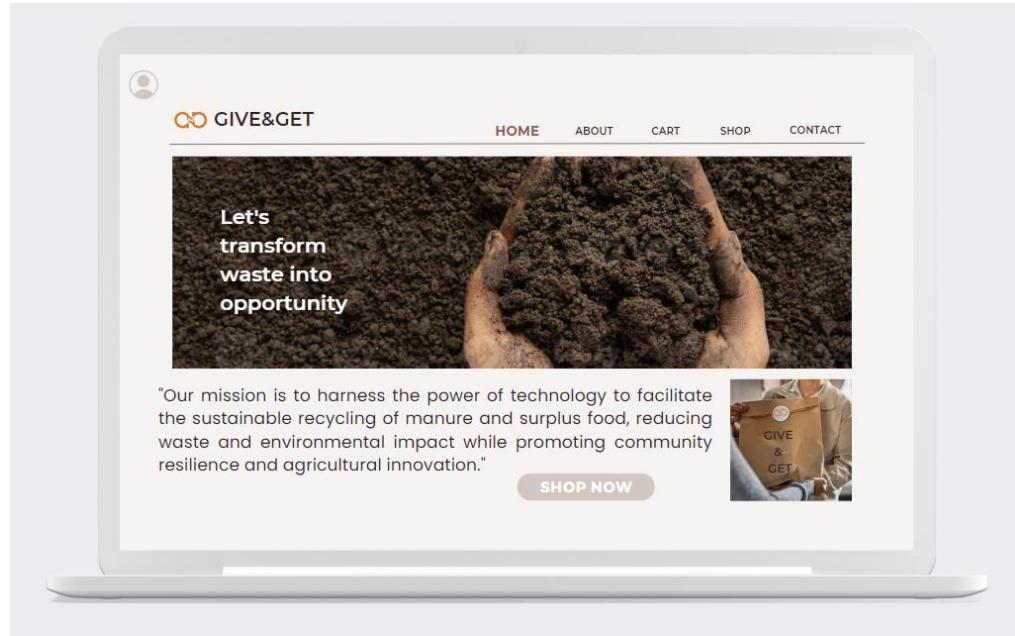


Figure 3.10 Home page

Figure 3.2.3 shows the About Us page to introduce the website to users.

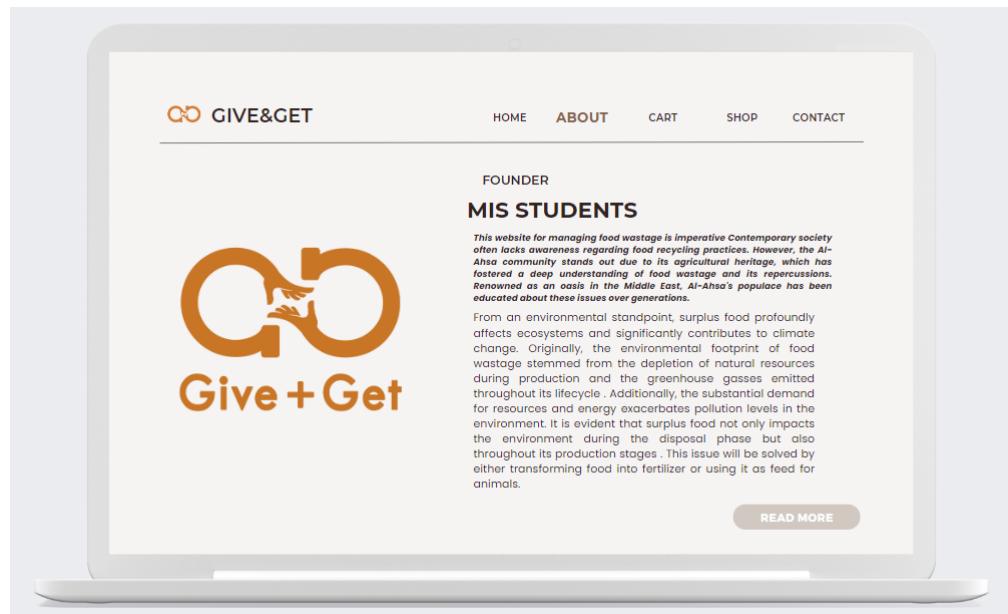


Figure 3.11 About us page

The screenshot shows the shopping cart page for the "GIVE&GET" website. At the top, there's a navigation bar with links for HOME, ABOUT, CART, SHOP, and CONTACT. Below the navigation is a table with four columns: Item, Price, Quantity, and Total. A single item, a yellow bag of fertilizer, is listed with a price of 10 SR, a quantity of 1, and a total of 10 SR. Below the table, there's a summary section with Subtotal (10 SR), Sales Tax (15 SR), and a Grand Total of 15 SR. There's also a link to Add Coupon. At the bottom, a message says "CONGRATS,, YOU'RE ELIGIBLE FOR FREE SHIPPING" followed by a small truck icon, and a "Check out" button.

Figure 3.12 Cart page

Figure 3.13 shows the products that users can browse and shop from.

The screenshot shows the products page for the "GIVE & GET" website. At the top, there's a navigation bar with links for HOME, ABOUT, CART, SHOP, and CONTACT. The main heading is "GIVE & GET". Below the heading, there are six product cards arranged in two rows of three. Each card includes a small image, the product name, its price, and an "ADD TO CART" button.

Image	Name	Price	Action
	fruit manure	16 SAR	ADD TO CART
	Chicken manure	20 SAR	ADD TO CART
	Surplus food	15 SAR	ADD TO CART
	Vegetables manure	18 SAR	ADD TO CART
	Beef manure	25 SAR	ADD TO CART

Figure 3.13 Products page

Figure 3.14 shows the Contact Us page to enable users to communicate with us.

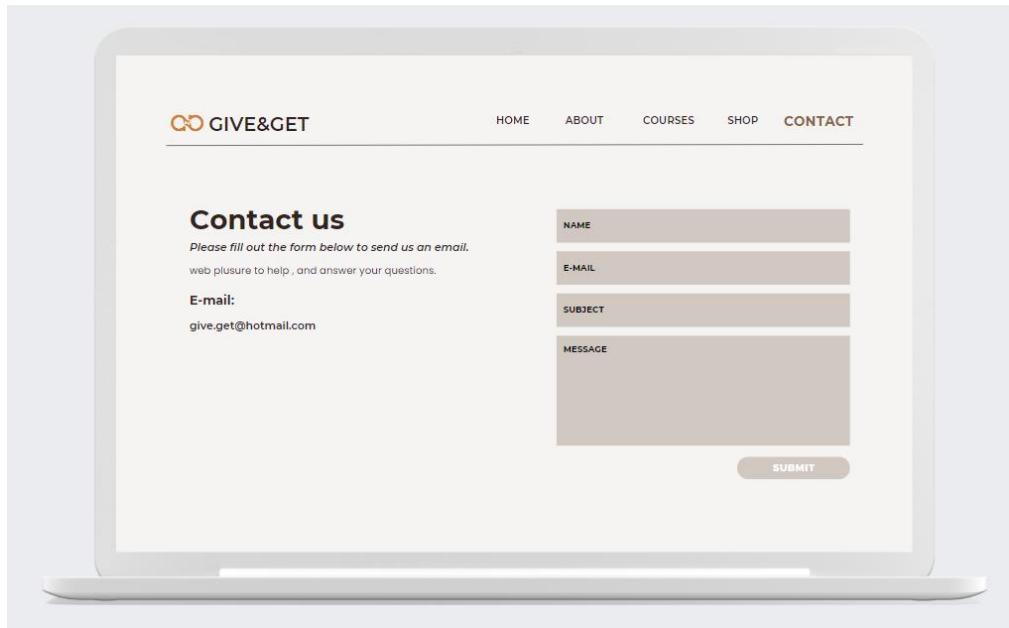


Figure 3.14 contact us page

Figure 3.15 illustrates that users can access their profile information.

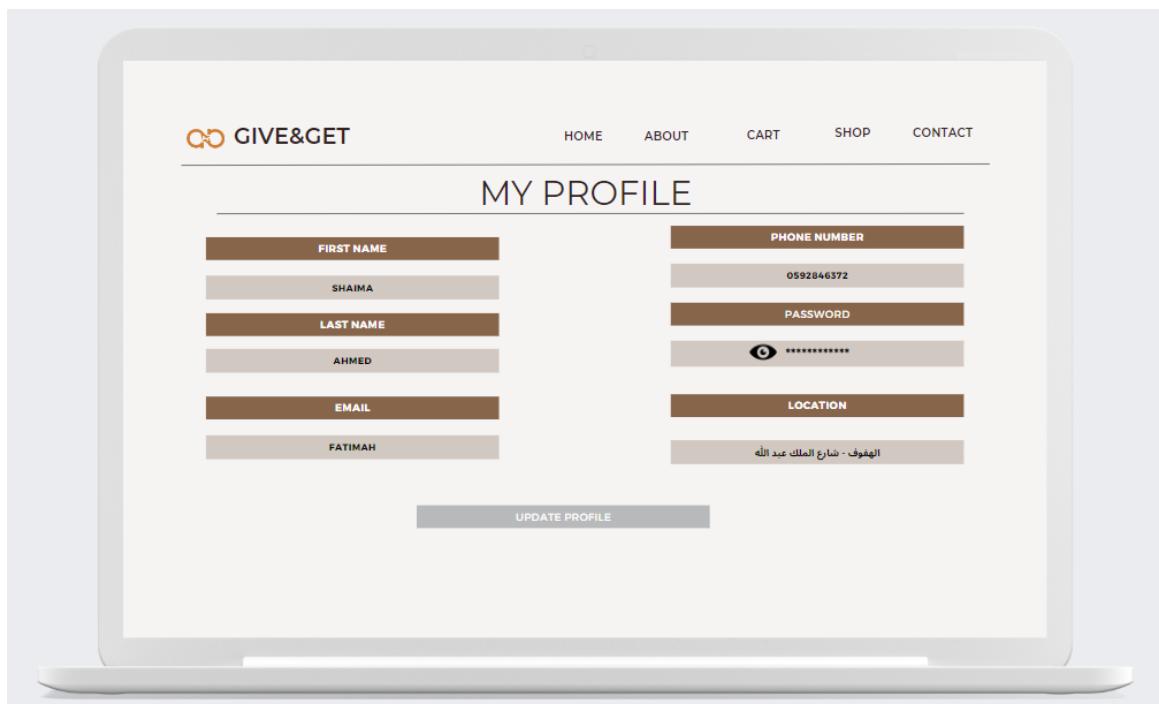


Figure 3.15 Profile page

Figure 3.2.8 shows that users can review the previous orders.

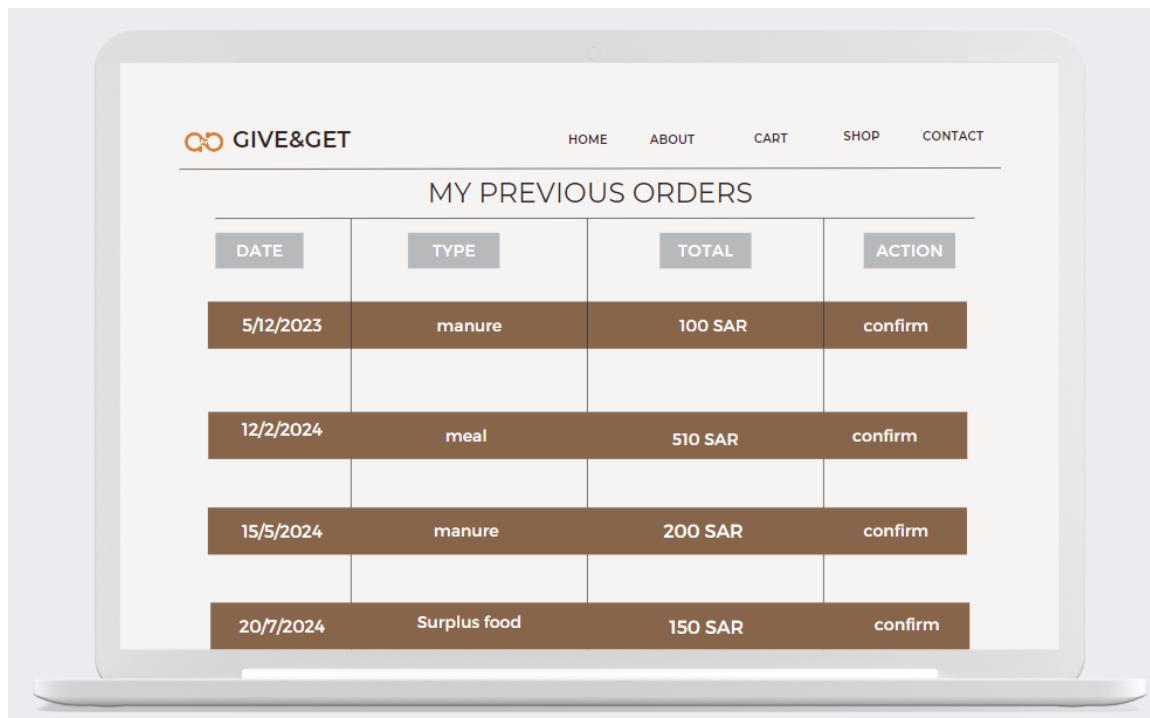


Figure 3.16 Previous order page

3.3 Entity Relationship Diagram

Figure 3.17 below is the entity relationship diagram of Give & Get that contains the following entities: employee, admin, branch, client, farmer, non-profit organization, restaurant, manure, and food item.

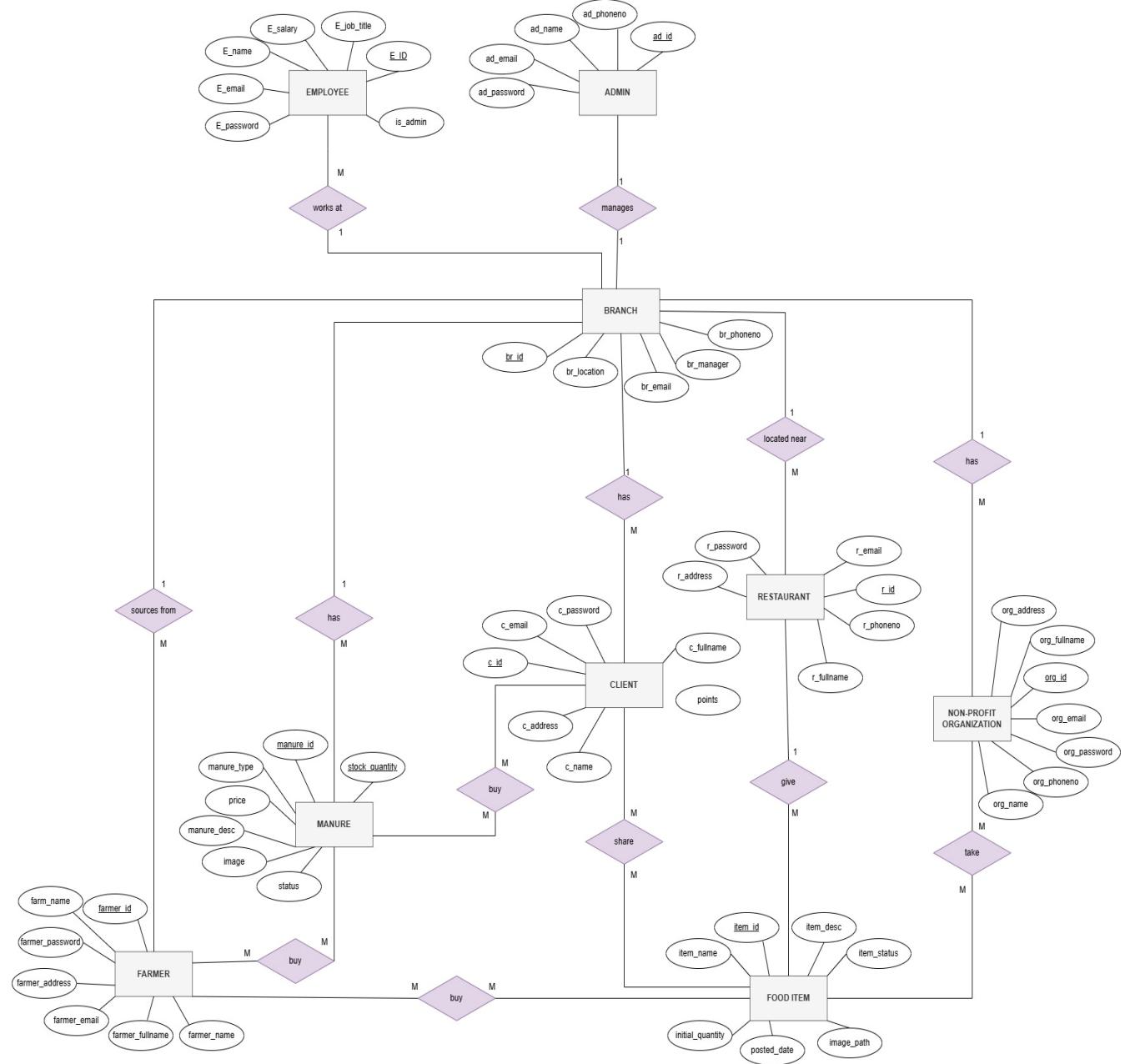


Figure 3.17 Entity Relationship Diagram

3.4 Normalization

Normalization is an essential process in database management for reducing redundancy and ensuring data integrity.

1NF (First Normal Form):

First Normal Form ensures that there are no atomic values meaning no multivalued attributes or duplicated columns in the same table. In [Figure 3.18](#), all the tables are in the first normal form since there are no multivalued attributes and no duplicated columns.

CLIENT	c_id	c_address	c_password	c_email	c_name	c_fullname	points
FARMER	farmer_id	farmer_address	farmer_password	farmer_email	farmer_name	farmer_fullname	farm_name
NON-PROFIT ORGANIZATION	org_id	org_address	org_email	org_password	org_phoneno	org_name	org_fullname
RESTAURANT	r_id	r_address	r_email	r_password	r_phoneno	r_name	r_fullname
FOOD ITEM	item_id	item_name	item_desc	item_status	image	posted_date	initial_quantity
EMPLOYEE	E_id	E_job_title	E_salary	E_password	E_email	E_name	br_id
MANURE	manure_id	manure_type	manure_desc	price	image	stock_quantity	status
BRANCH	br_id	br_location	br_email	br_manager	br_phoneno	ad_id	
ADMIN	ad_id	ad_email	ad_password	ad_name	ad_phoneno		
DONATIONS	DonationID	DonationDate	QuantityDonated	donation_status	item_id	org_id	
CLIENT ORDERS	order_id	c_id	manure_id	quantity	invoice_id	order_status	order_date
SHARED FOOD ITEMS	SharedID	ShareDate	QuantityShared	item_id	c_id		
LEFTOVER ITEMS	LF_item_id	LF_item_name	LF_item_desc	LF_quantity	LF_image_path	LF_availability	LF_price
LEFTOVER ORDERS	LFO_id	farmer_id	LF_item_id	LFO_quantity	LFO_invoice_id	LFO_order_status	LFO_order_date
ORDERS	f_order_id	farmer_id	manure_id	f_quantity	f_invoice_id	f_order_status	f_order_date

[Figure 3.18](#) 1NF Diagram

2NF (Second Normal Form):

2NF removes redundancy by ensuring that every piece of data depends on the full primary key so the relations are already in 2NF.

3NF (Third Normal Form):

3NF removes transitive dependencies. This means that all attributes depend on the primary key and not the non-primary key. As for our relations, they are already in 3NF.

3.5 Logical Database

Figure 3.19 shows the logical database design and illustrates the relationship between entities.

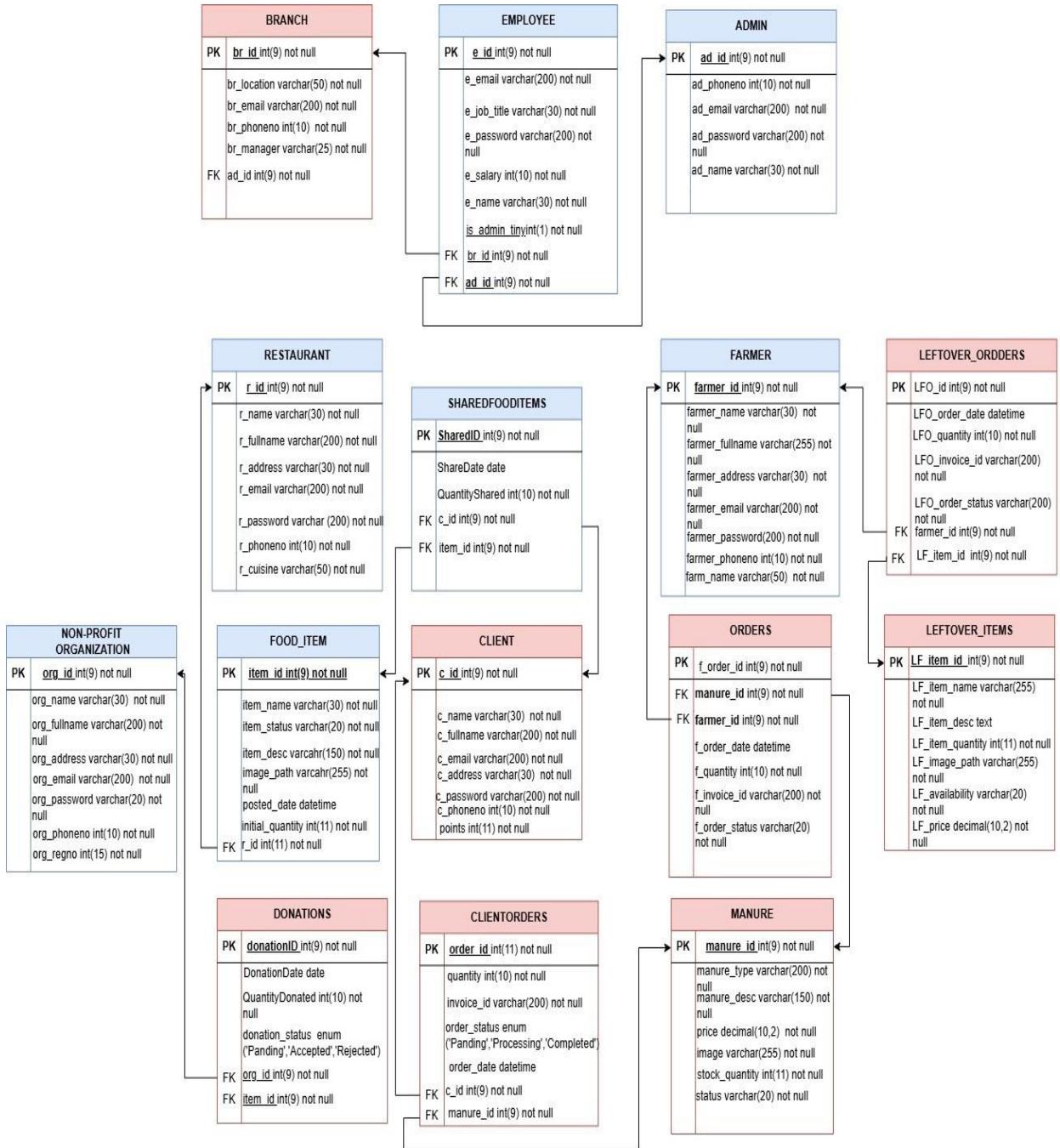


Figure 3.19 Logical Database

3.6 Physical Database

Figures from 3.20 to 3.35 illustrate the physical database design of the system including all tables (entities) and their attributes.

Give and Get Database

Table	Action	Rows	Type	Collation	Size	Overhead
admin		1	InnoDB	latin1_swedish_ci	16.0 Kib	-
branch		3	InnoDB	latin1_swedish_ci	96.0 Kib	-
client		13	InnoDB	latin1_swedish_ci	32.0 Kib	-
clientorders		13	InnoDB	latin1_swedish_ci	48.0 Kib	-
donations		17	InnoDB	latin1_swedish_ci	64.0 Kib	-
employee		3	InnoDB	latin1_swedish_ci	48.0 Kib	-
farmer		9	InnoDB	latin1_swedish_ci	32.0 Kib	-
food_item		34	InnoDB	latin1_swedish_ci	32.0 Kib	-
leftover_items		4	InnoDB	latin1_swedish_ci	16.0 Kib	-
leftover_orders		6	InnoDB	latin1_swedish_ci	48.0 Kib	-
manure		5	InnoDB	latin1_swedish_ci	48.0 Kib	-
nonprofit_org		10	InnoDB	latin1_swedish_ci	32.0 Kib	-
orders		11	InnoDB	latin1_swedish_ci	64.0 Kib	-
restaurant		10	InnoDB	latin1_swedish_ci	32.0 Kib	-
sharedfooditems		24	InnoDB	latin1_swedish_ci	48.0 Kib	-
15 tables		163	InnoDB	latin1_swedish_ci	656.0 Kib	0 B
Sum						

Figure 3.20 Give&Get Database

The screenshot shows the MySQL Workbench interface with the 'admin' table selected. The top navigation bar includes tabs for 'Server: 127.0.0.1', 'Database: giveandget', and 'Table: admin'. Below the navigation are various management buttons: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Tracking. The main area displays the 'Table structure' tab, which lists the columns of the 'admin' table. The columns are: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. The 'ad_id' column is defined as int(9), 'ad_phoneno' as int(10), 'ad_email' as varchar(200) with collation latin1_swedish_ci, 'ad_password' as varchar(200) with collation latin1_swedish_ci, and 'ad_name' as varchar(30) with collation latin1_swedish_ci. The 'Extra' column for 'ad_id' indicates it is AUTO_INCREMENT. Action buttons for each column include Change, Drop, Primary, Unique, Index, Spatial, and Fulltext. At the bottom of the table view, there are buttons for 'Add to central columns' and 'Remove from central columns'.

Figure 3.21 admin table

The screenshot shows the MySQL Workbench interface with the 'branch' table selected. The top navigation bar includes tabs for 'Server: 127.0.0.1', 'Database: giveandget', and 'Table: branch'. Below the navigation are various management buttons: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Tracking. The main area displays the 'Table structure' tab, which lists the columns of the 'branch' table. The columns are: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. The 'br_id' column is defined as int(9), 'br_location' as varchar(50) with collation latin1_swedish_ci, 'br_email' as varchar(200) with collation latin1_swedish_ci, 'br_phoneno' as int(10), and 'br_manager' as varchar(25) with collation latin1_swedish_ci. The 'ad_id' column is defined as int(9). The 'Extra' column for 'br_id' indicates it is AUTO_INCREMENT. Action buttons for each column include Change, Drop, Primary, Unique, Index, Spatial, and Fulltext. At the bottom of the table view, there are buttons for 'Add to central columns' and 'Remove from central columns'.

Figure 3.22 branch table

Server: 127.0.0.1 » Database: giveandget » Table: client

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	c_id	int(9)			No	None		AUTO_INCREMENT	Change Drop More
2	c_fullname	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
3	c_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
4	c_address	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	c_email	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
6	c_password	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
7	c_phoneno	int(10)			No	None			Change Drop More
8	points	int(11)			Yes	0			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Figure 3.23 client table

Server: 127.0.0.1 » Database: giveandget » Table: clientorders

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	order_id	int(11)			No	None		AUTO_INCREMENT	Change D
2	c_id	int(9)			No	None			Change D
3	manure_id	int(9)			No	None			Change D
4	quantity	int(10)			No	None			Change D
5	invoice_id	varchar(200)	latin1_swedish_ci		No	None			Change D
6	order_status	enum('Pending', 'Processing', 'Completed', 'Cancelled')	latin1_swedish_ci		Yes	Pending			Change D
7	order_date	datetime			No	None			Change D

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Figure 3.24 client orders table

Server: 127.0.0.1 » Database: giveandget » Table: employee

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	e_id	int(9)			No	None		AUTO_INCREMENT	Change Drop More
2	e_email	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
3	e_job_title	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
4	e_password	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
5	e_salary	int(10)			No	None			Change Drop More
6	e_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
7	br_id	int(9)			No	None			Change Drop More
8	is_admin	tinyint(1)			Yes	0			Change Drop More
9	ad_id	int(11)			Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial
 Add to central columns Remove from central columns

Figure 3.25 employee table

Server: 127.0.0.1 » Database: giveandget » Table: farmer

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	farmer_id	int(9)			No	None		AUTO_INCREMENT	Change Drop More
2	farmer_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	farmer_fullname	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
4	farmer_address	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	farmer_email	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
6	farmer_password	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
7	farmer_phoneno	int(10)			No	None			Change Drop More
8	farm_name	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial More
 Add to central columns Remove from central columns

Figure 3.26 farmer table

Server: 127.0.0.1 » Database: giveandget » Table: food_item

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	item_id	int(9)			No	None		AUTO_INCREMENT	Change Drop More
2	item_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	item_status	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
4	item_desc	varchar(150)	latin1_swedish_ci		No	None			Change Drop More
5	image_path	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
6	posted_date	datetime			Yes	NULL			Change Drop More
7	initial_quantity	int(11)			Yes	NULL			Change Drop More
8	r_id	int(11)			Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial More
 Add to central columns Remove from central columns

Figure 3.27 food item table

Server: 127.0.0.1 » Database: giveandget » Table: manure

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	manure_id	int(9)			No	None		AUTO_INCREMENT	Change Drop More
2	manure_type	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
3	manure_desc	varchar(150)	latin1_swedish_ci		No	None			Change Drop More
4	price	decimal(10,2)			No	0.00			Change Drop More
5	image	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
6	stock_quantity	int(11)			Yes	0			Change Drop More
7	status	varchar(20)	latin1_swedish_ci		Yes	Available			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial More
 Add to central columns Remove from central columns

Figure 3.28 manure table

Server: 127.0.0.1 » Database: giveandget » Table: nonprofit_org

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	org_id	int(9)			No	None		AUTO_INCREMENT	Change Drop More
2	org_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	org_fullname	varchar(200)	latin1_swedish_ci		Yes	NULL			Change Drop More
4	org_address	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	org_email	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
6	org_password	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
7	org_phoneno	int(10)			No	None			Change Drop More
8	org_rego	int(15)			No	None			Change Drop More

With selected: Change Drop Primary Unique Index Spatial

Figure 3.29 non-profit organization table

Server: 127.0.0.1 » Database: giveandget » Table: restaurant

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	r_id	int(9)			No	None		AUTO_INCREMENT	Change Drop More
2	r_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	r_fullname	varchar(200)	latin1_swedish_ci		Yes	NULL			Change Drop More
4	r_address	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	r_email	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
6	r_password	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
7	r_phoneno	int(10)			No	None			Change Drop More
8	r_cuisine	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

With selected: Change Drop Primary Unique Index Spatial

Figure 3.30 restaurant table

Server: 127.0.0.1 » Database: giveandget » Table: donations

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	donationID	int(9)			No	None		AUTO_INCREMENT	Change More
2	DonationDate	date			No	None			Change More
3	QuantityDonated	int(10)			No	None			Change More
4	org_id	int(11)			Yes	NULL			Change More
5	item_id	int(9)			No	None			Change More
6	donation_status	enum('Pending', 'Accepted', 'Rejected')	latin1_swedish_ci		Yes	Pending			Change More

With selected: Change Drop Primary Unique Index Spatial Fulltext

Figure 3.31 donations table

Server: 127.0.0.1 » Database: giveandget » Table: sharedfooditems

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	SharedID	int(9)			No	None		AUTO_INCREMENT	Change Drop More
2	ShareDate	date			No	None			Change Drop More
3	QuantityShared	int(10)			No	None			Change Drop More
4	c_id	int(9)			No	None			Change Drop More
5	item_id	int(9)			No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial
 Add to central columns Remove from central columns

Figure 3.32 shared food items table

Server: 127.0.0.1 » Database: giveandget » Table: leftover_items

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	LF_item_id	int(9)			No	None		AUTO_INCREMENT	Change Drop More
2	LF_item_name	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
3	LF_item_desc	text	latin1_swedish_ci		Yes	NULL			Change Drop More
4	LF_quantity	int(11)			No	None			Change Drop More
5	LF_image_path	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
6	LF_availability	varchar(20)	latin1_swedish_ci		Yes	Available			Change Drop More
7	LF_price	decimal(10,2)			No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext
 Add to central columns Remove from central columns

Figure 3.33 leftover items table

Server: 127.0.0.1 » Database: giveandget » Table: leftover_orders

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	farmer_id	int(9)			No	None			Change Drop More
2	LF_item_id	int(9)			No	None			Change Drop More
3	LFO_order_date	datetime			No	None			Change Drop More
4	LFO_quantity	int(10)			No	None			Change Drop More
5	LFO_invoice_id	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
6	LFO_order_status	varchar(200)	latin1_swedish_ci		No	Pending			Change Drop More
7	LFO_id	int(9)			No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext
 Add to central columns Remove from central columns

Figure 3.34 leftover orders table

Server: 127.0.0.1 » Database: giveandget » Table: orders

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	manure_id	int(9)			No	None			Change Drop More
<input type="checkbox"/>	2	farmer_id	int(9)			No	None			Change Drop More
<input type="checkbox"/>	3	f_order_date	datetime			No	None			Change Drop More
<input type="checkbox"/>	4	f_quantity	int(10)			No	None			Change Drop More
<input type="checkbox"/>	5	f_invoice_id	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	6	f_order_status	varchar(20)	latin1_swedish_ci		No	Pending			Change Drop More
<input type="checkbox"/>	7	f_order_id	int(9)			No	None	AUTO_INCREMENT		Change Drop More

↑ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Add to central columns Remove from central columns

Figure 3.35 orders table

4 Development

4.1 Tools/ technologies selected for implementation

- **HTML**

HTML is employed for the structure of each webpage as it represents the base of the front end that helps in providing the structure for the web pages [4]. It was used to markup content, create forms, and structure navigation within the web application. We chose it because it's known for its simplicity, universality, and compatibility in combination with technologies.

- **CSS**

CSS is used to style HTML content which helps to control the layout, design, and responsive behavior of the web application [4]. For this project, it was used to customize the appearance and layout for Give & Get web pages, help in making the system visually appropriate and user-friendly to attract users and makes them enjoy the use of it.

- **JavaScript**

JavaScript is employed for frontend development to make Give & Get web pages interactive [4]. It was used to check user inputs accuracy, processes submissions, and perform dynamic actions like displaying available Manure to buy, and available food for charities to request.

- **PHP**

PHP is used to process and manage data on the server as a server-side scripting language [4]. PHP main responsibilities is storing user information once they create accounts and retrieving their information when they use the system later, also PHP helps in displaying available manure for purchase, and handling requests for food donations from charities, help to ensure up-to-date information about the manure and available food for charitable organizations, while also focusing on managing the logistics of transactions and requests within Give&Get system.

- **PhpMyAdmin**

The aid of PhpMyAdmin, Give&Get database was worked with, designed, and managed more easily with a user-friendly interface, which allows us to access and use the

database directly without the need for command-line tools, which helps in avoiding major inconvenience in working.

- **XAMPP**

XAMPP is used to create a local web server environment, to speed up the development process and avoid risk as much as possible, test code locally, and manage websites before launching it, executing SQL queries, importing or exporting data. It enables us to efficiently manage relationships between tables, ensuring data integrity. Additionally, it ensures to secure the database during development and after deployment.

4.2 Web system navigation design

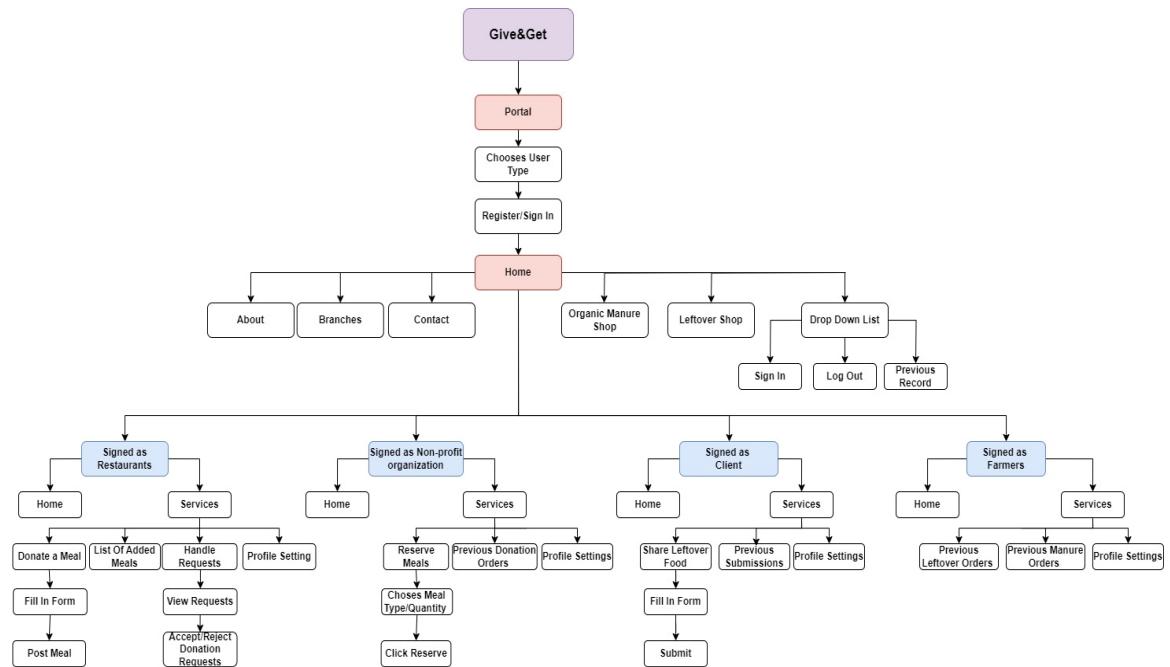


Figure 4.1 Web System Navigation Design

4.3 User Interface Design

Give&Get system user interface (UI) was developed with focus on simplicity and efficiency and designed to ensure an easy user experience for all users, including farmers, clients, restaurants, and non-profit organizations.

1. Portal Page:

Once users enter Give & Get system, they will be greeted with a portal page where each user must select their user type (farmer, client, restaurant, or non-profit organization),

as each user type has different home page which contain individual setting and provide different services. We design this portal to ensure that the system tailors the subsequent interface to the specific needs of each user type. After choosing their user type that indicates their different roles, users will log in securely to access personalized features and services.

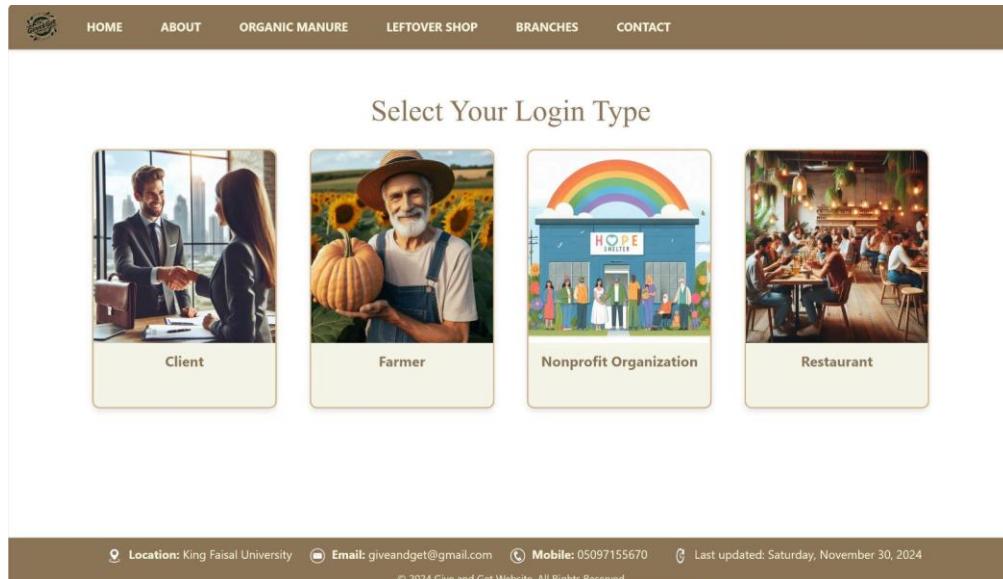


Figure 4.2 Portal Page

2. Home Page:

Once logged in, the user is directed to the home page, which features the following elements:

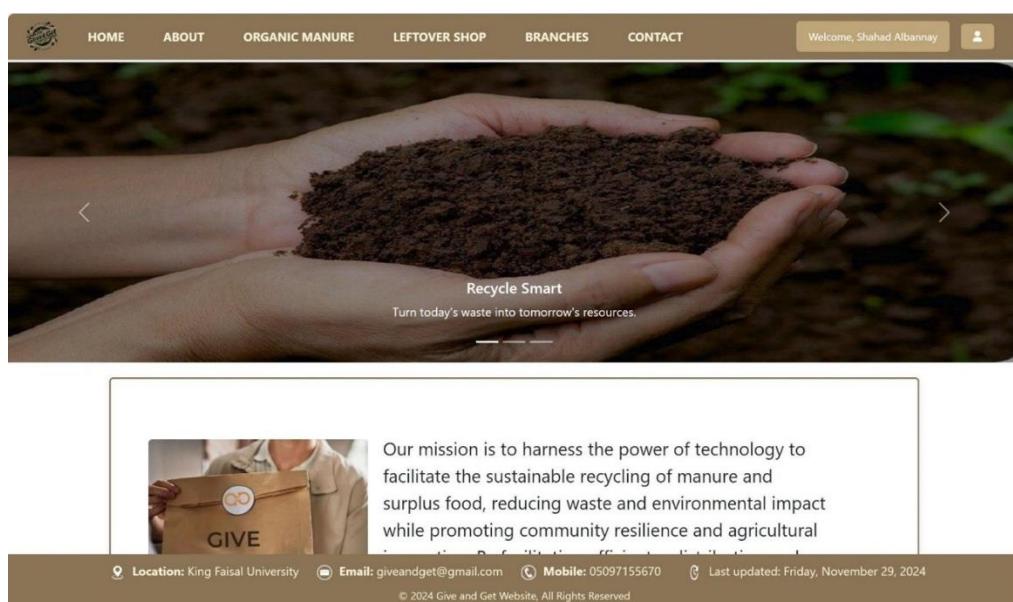


Figure 4.3 Home Page



Figure 4.4 Home Page

- **About page**

Figure 4.5 About Page

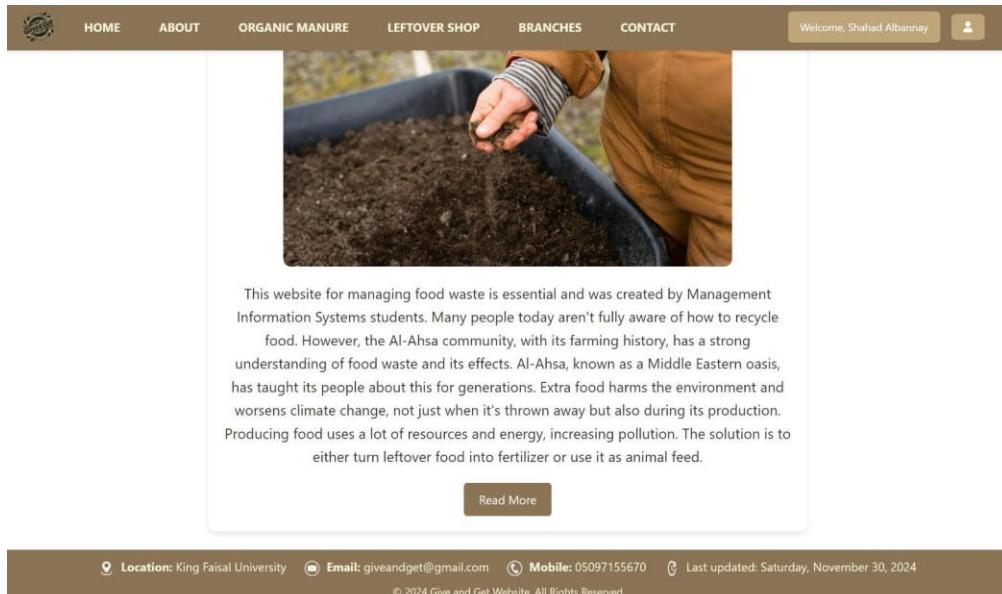


Figure 4.6 About Page

- **Organic Manure page**

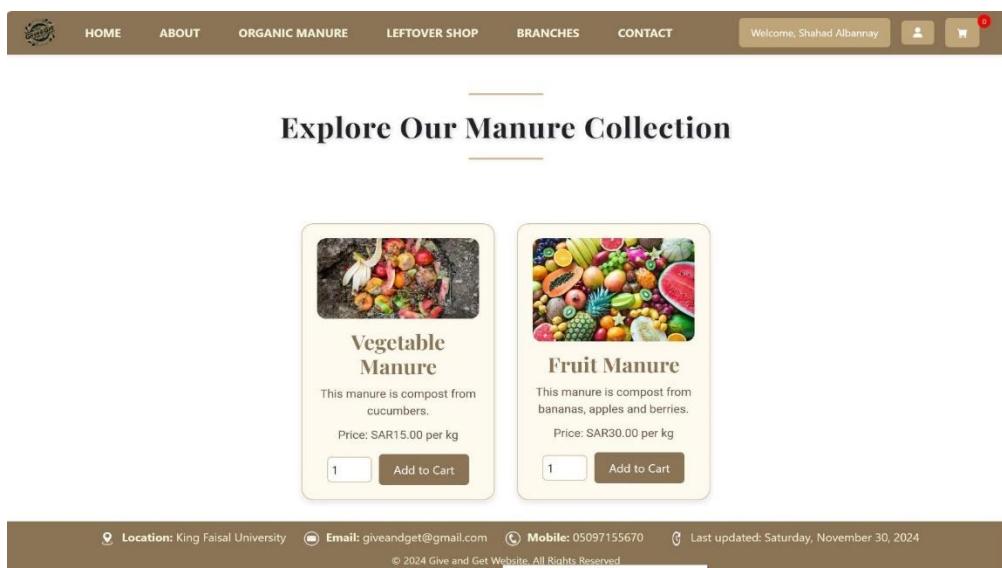


Figure 4.7 Manure Shop Page

- **Leftover Shop page**

The screenshot shows the 'LEFTOVER SHOP' section of the website. It features three product cards:

- Veggies Leftover**: Cucumber. Price: SAR15.00 per kg. Quantity: 1. Add to Cart button.
- Grains Leftover**: Rice and chickpeas. Price: SAR15.00 per kg. Quantity: 1. Add to Cart button.
- Meat Leftover**: Meat. Price: SAR12.00 per kg. Quantity: 1. Add to Cart button.

At the bottom, there is a footer bar with location, email, mobile number, and update information.

Figure 4.8 Leftover Shop Page

- o **Branches page**

The screenshot shows the 'BRANCHES' section of the website. It includes a map of the region with branch locations marked and three buttons for specific branches:

- Al Hasa Branch**
- Dammam Branch**
- Khobar Branch**

At the bottom, there is a footer bar with location, email, mobile number, and update information.

Figure 4.9 Branch Page

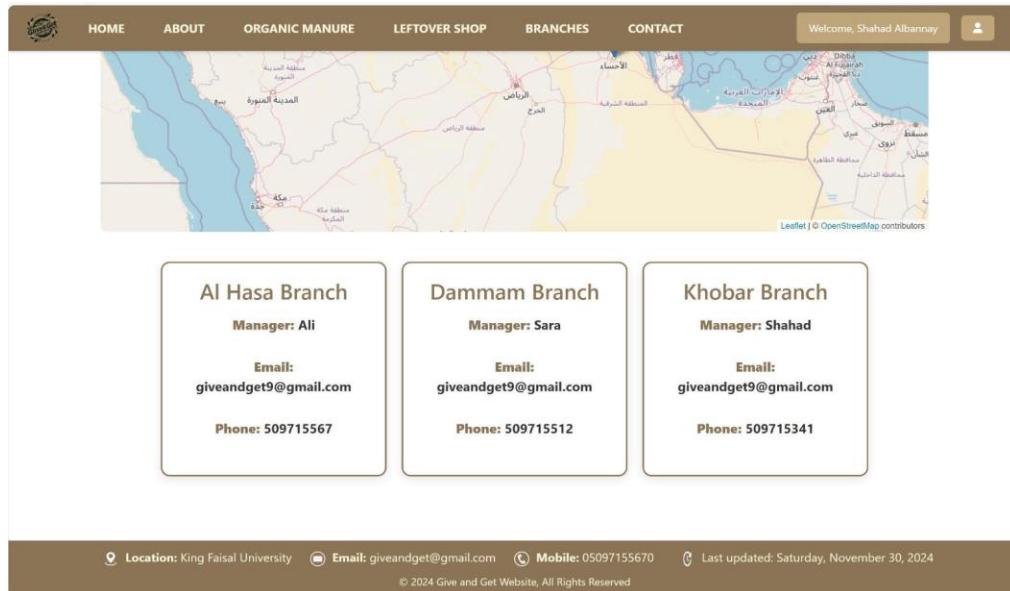


Figure 4.10 Branch Page

- **Contact page**

The screenshot shows a contact form page. The header includes a logo, navigation links (HOME, ABOUT, ORGANIC MANURE, LEFTOVER SHOP, BRANCHES, CONTACT), and a welcome message "Welcome, Guest". The main content is titled "Contact Us" and includes a note: "fill out the form below to send us an email." Below this is an "E-mail" section with the address "giveandget9@gmail.com". To the right is a large form with four input fields: "Name", "E-Mail", "Subject", and "Message", each with a red info icon. A "send" button is at the bottom of the form. The footer is identical to the one in Figure 4.10.

Figure 4.11 Contact US Page

- **FAQ page**

The screenshot shows the 'Frequently Asked Questions' section of the website. At the top, there is a navigation bar with links for HOME, ABOUT, ORGANIC MANURE, LEFTOVER SHOP, BRANCHES, and CONTACT. A 'Welcome, Guest' button is also present. Below the navigation is a title 'Frequently Asked Questions'. Underneath, a question 'What is Give&Get website?' is expanded, revealing a detailed answer about the platform's mission to reduce food waste and protect the environment through surplus food collection and reuse. Below this, several other questions are listed, each with a collapse/expand arrow:

- How can I register to Give&Get website?
- Is the Give&Get system secure?
- How can I control my account information?
- How can I contact customer support?
- How can I share leftover food through the website?
- How can I share untouched food through the website? Note: accessible to restaurants only.
- How can I request donations? Note: accessible to nonprofit organizations only.
- How can I purchase manure or leftover food? Note: accessible to farmers only.

At the bottom of the page, there is footer information including location (King Faisal University), email (giveandget9@gmail.com), mobile number (05097155670), a FAQ link, and a note indicating the last update was on Saturday, December 7, 2024. Copyright information for 2024 Give and Get Website, All Rights Reserved is also present.

Figure 4.12 Help and FAQ Page

○ **Setting page**

The screenshot shows the 'Profile Settings' page. At the top, there is a navigation bar with links for HOME, ABOUT, ORGANIC MANURE, LEFTOVER SHOP, BRANCHES, and CONTACT. A 'Welcome, Shahad AlBannay' button is also present. Below the navigation is a title 'Profile Settings' and a 'Total Point: 150' indicator. The page is divided into two main sections: 'Personal Information' and 'Reset Password'.

Personal Information: This section contains fields for Username (sh123), Email (shahadalbannay@outlook.com), Address (hasa), and Phone number (502829747). It includes a 'Save Information' button at the bottom left and a 'Delete Account' button at the bottom center.

Reset Password: This section contains fields for Current Password and New Password, along with a 'Save Password' button.

Figure 4.13 Settings Page

○ **Restaurant's Services Page**

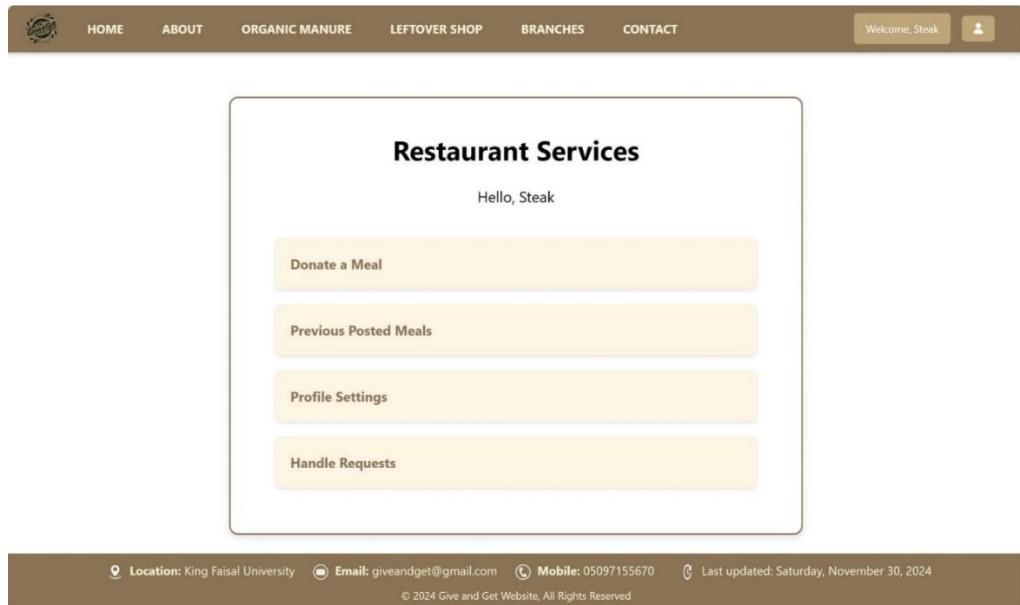


Figure 4.14 Restaurant Services Page

- Nonprofit organization's Services Page

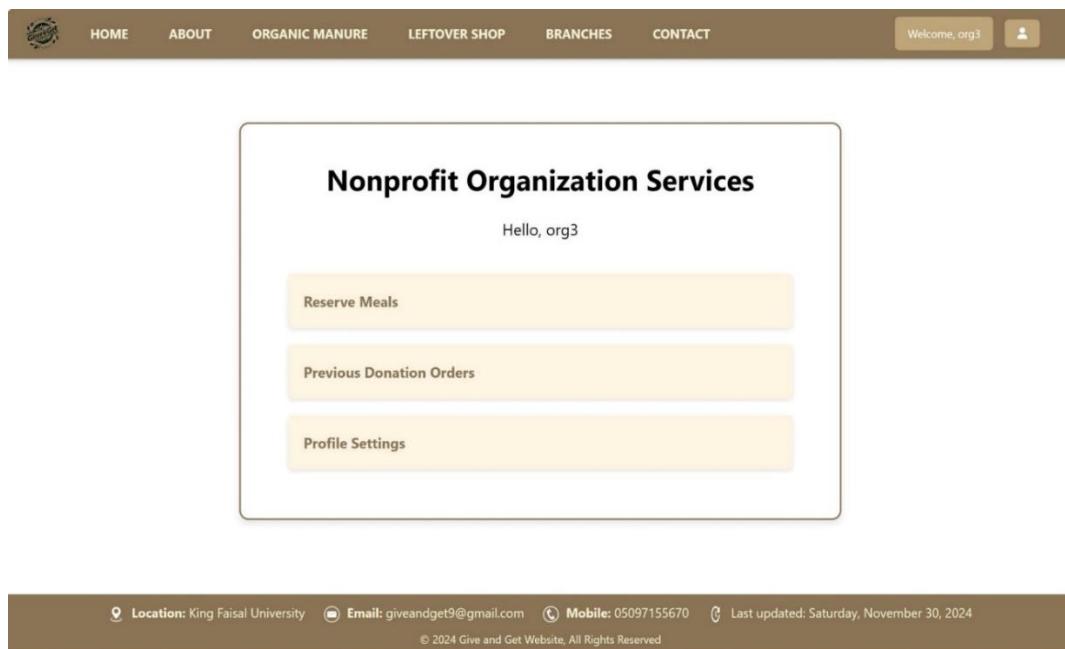


Figure 4.15 Nonprofit Organization Services Page

- **Farmer's Services Page**

The screenshot shows a website header with a brown background. On the left is a circular logo with a stylized tree. To its right are navigation links: HOME, ABOUT, ORGANIC MANURE, LEFTOVER SHOP, BRANCHES, and CONTACT. A welcome message "Welcome, Palm Tree" and a user icon are on the far right. The main content area has a white background with a rounded rectangle border. At the top of this box is the heading "Farmer Services". Below it is a greeting "Hello, Palm Tree farm!". Three yellow rectangular buttons are listed vertically: "Previous Manure Orders", "Previous Leftover Orders", and "Profile Settings". At the bottom of the page is a brown footer bar containing location information ("Location: King Faisal University"), contact details ("Email: giveandget9@gmail.com", "Mobile: 05097155670"), a "Last updated" timestamp ("Saturday, November 30, 2024"), and a copyright notice ("© 2024 Give and Get Website, All Rights Reserved").

Figure 4.16 Nonprofit Organization Services Page

- **Client's Services Page**

The screenshot shows a website header with a brown background. On the left is a circular logo with a stylized tree. To its right are navigation links: HOME, ABOUT, ORGANIC MANURE, LEFTOVER SHOP, BRANCHES, and CONTACT. A welcome message "Welcome, Shahad Albannay" and a user icon are on the far right. The main content area has a white background with a rounded rectangle border. At the top of this box is the heading "Client Services". Below it is a greeting "Hello, Shahad Albannay". Three yellow rectangular buttons are listed vertically: "Share Leftover Food", "Previous Submissions", and "Profile Settings". At the bottom of the page is a brown footer bar containing location information ("Location: King Faisal University"), contact details ("Email: giveandget9@gmail.com", "Mobile: 05097155670"), a "Last updated" timestamp ("Wednesday, December 4, 2024"), and a copyright notice ("© 2024 Give and Get Website, All Rights Reserved").

Figure 4.17 Client Services Page

Usability Enhancements

- **Clear Navigation:** The header is important to both new and previous users because it is one of the first elements that users will lay eyes on, and helps to provide clear,

consistent navigation throughout the platform, allowing users to understand how to navigate efficiently and quickly. This is important as it saves users time and effort on searching for what they want on the website.

- **Form Validation:** Forms on the website (such as login, registration, and contact forms) have tools to ensure validation and accuracy of the entered users' data. This excludes errors from happening and helps users quickly correct mistakes. These processes are important to make sure to only accept real and up-to-date information.
- **Responsive Design:** The UI is designed to be responsive, allow users to access and navigate the website easily on multiple devices, from desktops to mobile phones, and to achieve the optimal appearance regardless of the device being used.

4.4 Database connectivity

- **Installing XAMPP**

The first step in the creation of Give&Get database was accessing the XAMPP Control Panel to install XAMPP on a local computer and launching the Apache and MySQL services to enable us to access PhpMyAdmin and use it to create, edit, and manage the database and tables, and carry out other necessary tasks.

- **Creation of Databases**

We developed a database called "Give&Get" using phpMyAdmin and SQL. This database represents the main location for organizing and storing different datasets. Next, we create tables (such as admin and branch) to methodically arrange and store the data.

Table	Action	Rows	Type	Collation	Size	Overhead
admin		1	InnoDB	latin1_swedish_ci	16.0 KiB	-
branch		3	InnoDB	latin1_swedish_ci	96.0 KiB	-
client		12	InnoDB	latin1_swedish_ci	32.0 KiB	-
clientorders		13	InnoDB	latin1_swedish_ci	48.0 KiB	-
donations		17	InnoDB	latin1_swedish_ci	64.0 KiB	-
employee		3	InnoDB	latin1_swedish_ci	48.0 KiB	-
farmer		9	InnoDB	latin1_swedish_ci	32.0 KiB	-
food_item		34	InnoDB	latin1_swedish_ci	32.0 KiB	-
leftover_items		4	InnoDB	latin1_swedish_ci	16.0 KiB	-
leftover_orders		6	InnoDB	latin1_swedish_ci	48.0 KiB	-
manure		5	InnoDB	latin1_swedish_ci	48.0 KiB	-
nonprofit_org		9	InnoDB	latin1_swedish_ci	32.0 KiB	-
orders		10	InnoDB	latin1_swedish_ci	64.0 KiB	-
restaurant		10	InnoDB	latin1_swedish_ci	32.0 KiB	-
sharedfooditems		24	InnoDB	latin1_swedish_ci	48.0 KiB	-
15 tables	Sum				160 InnoDB latin1_swedish_ci	656.0 KiB
						0 B

Check all

Print Data dictionary

Figure 4.18 Give and Get tables

- "Admin": To store admin information.
- "Employee": To store employees' information.
- "Client": To store clients' information.
- "Farmer": To store farmers' information.
- "Restaurant": To store restaurant information.
- "Nonprofit_org": To store nonprofit organization's information.
- "Branch": To store all branch's locations and data.
- "Manure": To store manure information.
- "Donations": To save requested and reserved meals' information from nonprofit organizations.
- "Orders": To save farmers purchases information about manure and leftover food.
- "Leftover_items": To save information about items posted by the admin to the leftover shop.
- "Leftover_orders": To store orders from farmers in the leftover shop.

- "Clientorders": To store orders from clients in the organic manure shop
- "Food_item": to store submitted used food by clients or posted untouched food from restaurants for donations.
- "Sharedfooditems": To save information about the shared food by clients.

5 Testing and Conversion

5.1 Test Plan

Test requirements

Use Cases

- After registration, Clients log into the system, order manure, share surplus food, view order details, and review their transactions.
- After registration, Farmers log into the system, order manure or surplus food, and view order details.
- After registration, Restaurants log into the system, add notifications about donations, share surplus food, and reviews their transactions.
- After registration, Non-profit organizations log into the system, request donations, and view request details.
- Administrator manages all users and orders, edits and archives products, and follows up on monthly reports.

Functional Requirements:

Functional testing in Give & Get is designed to ensure the safe operation of each system feature for all users. The focus is on core functionality such as managing donations, sharing surplus food, and requesting manure:

1. Manure Ordering by clients and Farmers:
 - Verify that clients and farmers possess the ability to browse and order manure.
 - Ensure the process of recording order details is accurate and provide payment confirmation.
2. Surplus food sharing by clients and restaurants:
 - Verify the ability to share surplus food in the system with clients and restaurants.
 - Ensure the functionality of food sharing, including ease of input and listing visibility.
3. Food Donation by Restaurants:

- Verify that restaurants possess the ability to notify the system about food availability for donation.
- Confirm the correctness and visibility of donation records to non-profit organizations.

4. Admin Management of System Operations:

- Confirm the admin's ability to access and control everything within the system such as managing users, approving donations, and editing system listings.
- Ensure seamless functionality without any occurrence of problems connected to generating reports or resolving disputes.

Non-Functional Requirements

“Give & Get” incorporates various non-functional requirements to enhance system performance and user experience.

- ✓ Usability: Ensure a user-friendly interface that simplify how users interact and navigate on the website.
- ✓ Performance: Ensure the system can operate efficiently even under high usage or pressure.
- ✓ System Security: Protect the data and its transactions for each user from any unauthorized access.
- ✓ Database update mechanism: Ensure to provide continuous updates for the shared food and manure lists.

Test strategy

The design testing strategy that evaluates and ensure the comprehensive of its functionalities for Give & Get system, the approach involves a systematic examination of both base and additional functionalities:

1. Evaluate Core Functionality: Make sure to verify how the essential features are working by testing the website fundamentals such as registration, login, and order manure, donation, or share surplus food.
2. Test additional Functionality: Make sure to verify how the additional features are working by subsequent testing that extends to the additional functions of the system to address additional features such as donation scheduling.
3. Data Integrity Assessment: Help in verifying the transaction for all the data that are in the system operations ensuring consistency and accuracy.
4. Error Handling: look for all possible incorrect inputs and sudden user behavior and manage them.
5. Functional Testing: Make sure that all the features are working as planned and deal efficiently with issues by providing all the possible solutions.

“Give & Get” system aims to ensure reliable functionality, data accuracy, and an optimized user experience by following these structured approaches.

5.2 Unit testing

Users Sign up [1]

Table 5.1 User Registration Case

Test Case Name	Registration Success
Purpose	Verify successful client registration
Component Name	login.php
Component Type (method/object)	\$_REQUEST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Click on “Register here”. 2. Fill out valid information on the registration form. 3. Click the "SIGN UP" button. 4. Account is created.
Input Provided	Username, Full Name, Email, Mobile Number, Location, Password.
Expected Outcome	A new user account is created, and the user is redirected to the login page.
Actual Outcome	The account is created successfully and inserted into the database.
Status (Pass/Fail)	Pass
Comments	-

Users Sign up [2]

Table 5.2 User Registration Failure Case

Test Case Name	Registration Failure
Purpose	Verifies that the system handles duplicate email or username during signup.
Component Name	login.php
Component (method/object) Type	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Click on “Register here”. 2. Fill out an already registered email or username, enforce password requirements (must be at least 6 characters or required), and ensure the phone number is exactly 10 digits, along with other information on the registration form. 3. Click the "SIGN UP" button. 4. Account is created.
Input Provided	Username, Full Name, Email, Mobile Number, Location, Password.
Expected Outcome	System displays an error message indicating that the either email or username is already in use, or Mobile Number or Password are not meeting the required characters.
Actual Outcome	The account isn't created and displays one or more of the following messages: 'This Email Already Exists.', 'This Username Already Exists.', 'Password is required or must be at least 6 characters.', and 'Phone Number must be exactly 10 digits.'
Status (Pass/Fail)	Pass
Comments	-

Users Sign up [1]

Table 5.3 User Registration Case

Test Case Name	Registration Success
Purpose	Verify successful farmer registration
Component Name	login.php
Component (method/object) Type	\$_REQUEST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Click on “Register here”. 2. Fill out valid information on the registration form. 3. Click the "SIGN UP" button. 4. Account is created.
Input Provided	Username, Full Name, Email, Mobile Number, Location, Password, Farm Name.
Expected Outcome	A new user account is created, and the user is redirected to the login page.
Actual Outcome	The account is created successfully and inserted into the database, and the user is redirected to the login page.
Status (Pass/Fail)	Pass
Comments	-

Users Sign up [2]

Table 5.4 User Registration Failure Case

Test Case Name	Registration Failure
Purpose	Verifies that the system handles duplicate email or username, enforces password requirements (must be at least 6 characters or required), and ensures the phone number is exactly 10 digits during signup.
Component Name	login.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Click on "Register here". 2. Fill out an already registered email or username, enforce password requirements (must be at least 6 characters or required), and ensure the phone number is exactly 10 digits, along with other information on the registration form. 3. Click the "SIGN UP" button. 4. Account is created.
Input Provided	Username, Full Name, Email, Mobile Number, Location, Password, Farm Name.
Expected Outcome	System displays an error message indicating that the either email or username is already in use, or Mobile Number or Password are not meeting the required characters.
Actual Outcome	The account isn't created and displays one or more of the following messages: 'This Email Already Exists.', 'This Username Already Exists.', 'Password is required or must be at least 6 characters.', and 'Phone Number must be exactly 10 digits.'
Status (Pass/Fail)	Pass
Comments	-

Users Sign up [1]

Table 5.5 User Registration Case

Test Case Name	Registration Success
Purpose	Verify successful nonprofit organization registration
Component Name	login.php
Component Type (method/object)	\$_REQUEST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Click on "Register here". 2. Fill out valid information on the registration form. 3. Click the "SIGN UP" button. 4. Account is created.
Input Provided	Username, Full Name, Email, Mobile Number, Location, Password, Commercial Registration Number.
Expected Outcome	A new user account is created, and the user is redirected to the login page.

Actual Outcome	The account is created successfully and inserted into the database, and the user is redirected to the login page.
Status (Pass/Fail)	Pass
Comments	-

Users Sign up [2]

Table 5.6 User Registration Failure Case

Test Case Name	Registration Failure
Purpose	Verifies that the system handles duplicate email or username, enforces password requirements (must be at least 6 characters or required), and ensures the phone number is exactly 10 digits during signup.
Component Name	login.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<p>5. Click on "Register here".</p> <p>6. Fill out an already registered email or username, enforce password requirements (must be at least 6 characters or required), and ensure the phone number is exactly 10 digits, along with other information on the registration form.</p> <p>7. Click the "SIGN UP" button.</p> <p>8. Account is created.</p>
Input Provided	Username, Full Name, Email, Mobile Number, Location, Password, Commercial Registration Number.
Expected Outcome	System displays an error message indicating that the either email or username is already in use, or Mobile Number or Password are not meeting the required characters.
Actual Outcome	The account isn't created and displays one or more of the following messages: 'This Email Already Exists.', 'This Username Already Exists.', 'Password is required or must be at least 6 characters.', and 'Phone Number must be exactly 10 digits.'
Status (Pass/Fail)	Pass
Comments	-

Users Sign up [1]

Table 5.7 User Registration Case

Test Case Name	Registration Success
Purpose	Verify successful restaurant registration
Component Name	login.php

Component Type (method/object)	\$_REQUEST method
Test Description (Steps)	<p>5. Click on "Register here".</p> <p>6. Fill out valid information on the registration form.</p> <p>7. Click the "SIGN UP" button.</p> <p>8. Account is created.</p>
Input Provided	Username, Restaurant Name, Email, Mobile Number, Location, Password, Type of Cuisine.
Expected Outcome	A new user account is created, and the user is redirected to the login page.
Actual Outcome	The account is created successfully and inserted into the database, and the user is redirected to the login page.
Status (Pass/Fail)	Pass
Comments	-

Users Sign up [2]

Table 5.8 User Registration Failure Case

Test Case Name	Registration Failure
Purpose	Verifies that the system handles duplicate email or username, enforces password requirements (must be at least 6 characters or required), and ensures the phone number is exactly 10 digits during signup.
Component Name	login.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<p>1. Click on "Register here".</p> <p>2. Fill out an already registered email or username, enforce password requirements (must be at least 6 characters or required), and ensure the phone number is exactly 10 digits, along with other information on the registration form.</p> <p>3. Click the "SIGN UP" button.</p> <p>4. Account is created.</p>
Input Provided	Username, Restaurant Name, Email, Mobile Number, Location, Password, Type of Cuisine.
Expected Outcome	System displays an error message indicating that the either email or username is already in use, or Mobile Number or Password are not meeting the required characters.
Actual Outcome	The account isn't created and displays one or more of the following messages: 'This Email Already Exists.', 'This Username Already Exists.', 'Password is required or must be at least 6 characters.', and 'Phone Number must be exactly 10 digits.'

Status (Pass/Fail)	Pass
Comments	-

Users Sign in [1]

Table 5.9 Admin sign in case

Test Case Name	Admin log in Success
Purpose	Verify successful admin sign-in.
Component Name	login.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Click on "Log in". 2. Enter a valid admin email and password. 3. Click the "Log In" button.
Input Provided	email, Password.
Expected Outcome	Admin is logged in and redirected to the panel.
Actual Outcome	The admin logs in successfully and is redirected to the panel.
Status (Pass/Fail)	Pass
Comments	-

Users Sign in [2]

Table 5.10 Employee sign in case

Test Case Name	Employee Sign in Success
Purpose	Verify successful employee sign-in
Component Name	login.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Click on "Log in". 2. Enter a valid email and password. 3. Click the "Log In" button.
Input Provided	Email, Password.
Expected Outcome	Employee is logged in and redirected to the panel.
Actual Outcome	The employee logs in successfully and is redirected to the panel.
Status (Pass/Fail)	Pass

Comments	-
----------	---

Users Sign in [3]

Table 5.11 Client sign in case

Test Case Name	Client Sign in Success
Purpose	Verify successful client sign-in
Component Name	login.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	1. Enter a valid username and password. 2. Click the "Log In" button.
Input Provided	Username, Password.
Expected Outcome	Client is logged in and is redirected to the client services page.
Actual Outcome	The client logs in successfully and is redirected to the client services.
Status (Pass/Fail)	Pass
Comments	-

Users Sign in [4]

Table 5.12 farmer sign in case

Test Case Name	Farmer Sign in Success
Purpose	Verify successful farmer sign-in
Component Name	login.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	1. Enter a valid username and password. 2. Click the "Log In" button.
Input Provided	Username, Password.
Expected Outcome	Farmer is logged in and is redirected to the farmer services page.
Actual Outcome	The farmer logs in successfully and is redirected to the farmer services.
Status (Pass/Fail)	Pass
Comments	-

Users Sign in [5]

Table 5.13 Admin sign in case

Test Case Name	Nonprofit Organization Sign in Success
Purpose	Verify successful nonprofit organization sign-in
Component Name	login.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	1. Enter a valid username and password. 2. Click the "Log In" button.
Input Provided	Username, Password.
Expected Outcome	Nonprofit organization is logged in and is redirected to the nonprofit organization services page.
Actual Outcome	The nonprofit organization logs in successfully and is redirected to the nonprofit organization services.
Status (Pass/Fail)	Pass
Comments	-

Users Sign in [6]

Table 5.14 restaurant sign in case

Test Case Name	Restaurant Sign in Success
Purpose	Verify successful restaurant sign-in
Component Name	login.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	1. Enter a valid username and password. 2. Click the "Log In" button.
Input Provided	Username, Password.
Expected Outcome	Restaurant is logged in and is redirected to the restaurant services page.
Actual Outcome	The restaurant logs in successfully and is redirected to the restaurant services.
Status (Pass/Fail)	Pass

Comments	-
----------	---

Restaurant Posting Meal [1]

Table 5.15 Posting Meals case

Test Case Name	Successful Posting Meals Reservation
Purpose	Verify successful posting meals
Component Name	restaurant_post_meal.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the post meal page. 2. Fill in the meal name, description, initial quantity fields. 3. Upload a photo of the meal. 4. Post the meal.
Input Provided	Meal Name, Description of the Meal, Quantity and Photo.
Expected Outcome	The meal is successfully posted for nonprofit organizations, and a confirmation message will be displayed: 'Meal posted successfully'.
Actual Outcome	The meal is successfully posted, and the restaurant will receive a confirmation message: 'Meal posted successfully'.
Status (Pass/Fail)	Pass
Comments	-

Restaurant Posting Meal [2]

Table 5.16 Posting Meals failure

Test Case Name	Unsuccessful Restaurant Posting Meal
Purpose	Verify unsuccessful Restaurant Posting Meal
Component Name	restaurant_post_meal.php
Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the post meal page. 2. Fill in the meal name, description, initial quantity fields. 3. Not to upload a photo of the meal. 4. Post the meal.
Input Provided	Meal Name, Description of the Meal, and Quantity.
Expected Outcome	System denies posting meal with an error message indicating unavailability.

Actual Outcome	The meal isn't posted, and the following message is shown (Failed to post meal)
Status (Pass/Fail)	Pass
Comments	-

Restaurant Handling Requests [1]

Table 5.17 Handling Requests case

Test Case Name	Successful Handling Requests
Purpose	Verify successful handling requests
Component Name	approving_requests.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the handling requests page. 2. Read the request's information. 3. Choose 'Accept' or 'Reject'.
Input Provided	No input is provided, but the restaurant presses 'Accept' or 'Reject'
Expected Outcome	The requests are successfully accepted or rejected
Actual Outcome	The requests are successfully accepted or rejected, and a restaurant will be redirected to the same page "approving_requests.php".
Status (Pass/Fail)	Pass
Comments	-

Restaurant Handling Requests [2]

Table 5.18 Handling Requests failure

Test Case Name	Unsuccessful Handling Requests
Purpose	Verify Unsuccessful Restaurant Handling Requests
Component Name	approving_requests.php
Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the handling requests page. 2. Read the request's information. 3. Choose 'Accept' or 'Reject'

Input Provided	No input is provided, but the restaurant presses 'Accept' or 'Reject'.
Expected Outcome	The system prevents updating the request's status to 'Accept' or 'Reject,' ensuring the request remains visible on the restaurant handling requests page, and the database is not updated with the new status.
Actual Outcome	The request does not disappear and updated, and the restaurant isn't redirected to the same page.
Status (Pass/Fail)	Pass
Comments	-

Nonprofit Organization Meal Reservation [1]

Table 5.19 Meal Reservation case

Test Case Name	Successful Meal Reservation
Purpose	Verify successful meal reservation
Component Name	Reserve_meal.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the meal reservation page. 2. Review the meal details provided. 3. Enter the desired quantity for reservation. 4. Visit the previous donations page to verify if the reservation requests have been accepted or rejected by the restaurant.
Input Provided	Quantity, pressing "Reserve" button.
Expected Outcome	The reservation has been sent to the restaurant as a request successfully.
Actual Outcome	The reservation or request is successfully accepted or rejected.
Status (Pass/Fail)	Pass
Comments	-

Nonprofit Organization Meal Reservation [2]

Table 5.20 Meal Reservation failure

Test Case Name	Unsuccessful Meal Reservation
Purpose	Verify Unsuccessful Meal Reservation
Component Name	Reserve_meal.php

Component (method/object)	Type	\$_Post method
Test Description (Steps)		1. Navigate to the meal reservation page. 2. Review the meal details provided. 3. Enter the desired quantity for reservation. 4. Visit the previous donations page to verify if the reservation requests have been accepted or rejected by the restaurant.
Input Provided		Quantity, pressing “Reserve” button.
Expected Outcome		The system denies the reservation or request due to not enough stock of the meal.
Actual Outcome		The reservation request is unsuccessful, and the nonprofit organization will receive the message: 'Not enough quantity available.'
Status (Pass/Fail)		Pass
Comments		-

Client Posting Leftover Food [1]

Table 5.21 Posting Leftover Food case

Test Case Name	Successful Posting Leftover Food
Purpose	Verify successful Posting Leftover Food
Component Name	Client_food_submission.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	1. Navigate to the client food submission page. 2. Select what type of food is the client submitting. 3. Fill in the food name, description, quantity fields. 4. Upload a photo of the submitted food. 5. Click ‘Submit’ to finalize the process.
Input Provided	Type of Food, Food Name, Description, Quantity, Photo.
Expected Outcome	The leftover food posting has been sent to the database successfully to either compost it or sell it as is. Moreover, the client will get points whenever he or she shares excessive food.
Actual Outcome	The posting is successfully shared with confirmation messages ‘Food item successfully shared with quantity’ and ‘Points successfully updated!’.
Status (Pass/Fail)	Pass
Comments	-

Client Posting Leftover Food [2]

Table 5.22 Posting Leftover Food failure

Test Case Name	Unsuccessful Posting Leftover Food
Purpose	Verify Unsuccessful Posting Leftover Food

Component Name	Client_food_submission.php
Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the client food submission page. 2. Select what type of food is the client submitting. 3. Fill in the food name, description, quantity fields. 4. Upload a photo of the submitted food. 5. Click 'Submit' to finalize the process.
Input Provided	Type of Food, Food Name, Description, Quantity, Photo.
Expected Outcome	The system denies the posting due to an error in updating points or inserting data in shared food history.
Actual Outcome	The posting is unsuccessful, and the client will receive the message: 'Error updating points', 'Error sharing food item in the history table', or 'Error inserting into food_item table'.
Status (Pass/Fail)	Pass
Comments	-

Client Purchasing Manure [1]

Table 5.2.23 Purchasing Manure case

Test Case Name	Successful Purchasing Manure
Purpose	Verify successful Purchasing Manure
Component Name	shop.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the organic manure page. 2. Add available items to the cart. 3. Click on the number above the cart icon to navigate to the cart page. 4. Fill in the address information and save it. 5. Review the items added to the cart, and optionally you can choose to apply points earned to the total price for a discount. 6. Click on 'Proceed to Check Out' to navigate to the checkout page, review the address and cart items, and choose the preferred payment method. 7. After completing payment process, client will be redirected to the bill page.
Input Provided	Address Information, Payment Details, Items to be Purchased.
Expected Outcome	The item is successfully purchased with a confirmation page that displays "Your order has been placed successfully", and redirected to the bill page
Actual Outcome	The item is successfully shared with confirmation page that displays "Your order has been placed successfully", "Thank you for shopping with us!", and "view your order details".

Status (Pass/Fail)	Pass
Comments	-

Client Purchasing Manure [2]

Table 5.23 Purchasing Manure failure

Test Case Name	Unsuccessful Purchasing Manure
Purpose	Verify Unsuccessful Purchasing Manure
Component Name	shop.php
Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the organic manure page. 2. Add available items to the cart. 3. Click on the number above the cart icon to navigate to the cart page. 4. Fill in the address information and save it. 5. Review the items added to the cart, and optionally you can choose to apply points earned to the total price for a discount. 6. Click on ‘Proceed to Check Out’ to navigate to the checkout page, review the address and cart items, and choose the preferred payment method. 7. After completing payment process, client will be redirected to the bill page.
Input Provided	Address Information, Payment Details, Items to be Purchased.
Expected Outcome	The system denies the purchasing process due to one of the following situations: incomplete address, unsuccessful payment process, or not enough stock.
Actual Outcome	The purchasing process is unsuccessful, and the client will not be able to continue the process and get the bill page.
Status (Pass/Fail)	Pass
Comments	-

Farmer Purchasing Manure [1]

Table 5.24 Posting Meals case

Test Case Name	Successful Purchasing Manure
Purpose	Verify successful Purchasing Manure
Component Name	shop.php
Component Type (method/object)	\$_POST method

Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the organic manure page. 2. Add available items to the cart. 3. Click on the number above the cart icon to navigate to the cart page. 4. Fill in the address information and save it. 5. Review the items added to the cart, and optionally you can choose to apply points earned to the total price for a discount. 6. Click on 'Proceed to Check Out' to navigate to the checkout page, review the address and cart items, and choose the preferred payment method. 7. After completing payment process, farmer will be redirected to the bill page.
Input Provided	Address Information, Payment Details, Items to be Purchased.
Expected Outcome	The item is successfully purchased with a confirmation page that displays "Your order has been placed successfully", and redirected to the bill page
Actual Outcome	The item is successfully shared with confirmation page that displays "Your order has been placed successfully", "Thank you for shopping with us!", and "view your order details".
Status (Pass/Fail)	Pass
Comments	-

Farmer Purchasing Manure [2]

Table 5.25 Farmer Purchasing Manure failure

Test Case Name	Unsuccessful Purchasing Manure
Purpose	Verify Unsuccessful Purchasing Manure
Component Name	shop.php
Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the organic manure page. 2. Add available items to the cart. 3. Click on the number above the cart icon to navigate to the cart page. 4. Fill in the address information and save it. 5. Review the items added to the cart, and optionally you can choose to apply points earned to the total price for a discount. 6. Click on 'Proceed to Check Out' to navigate to the checkout page, review the address and cart items, and choose the preferred payment method. 7. After completing payment process, farmer will be redirected to the bill page.
Input Provided	Address Information, Payment Details, Items to be Purchased.
Expected Outcome	The system denies the purchasing process due to one of the following situations: incomplete address, unsuccessful payment process, or not enough stock.

Actual Outcome	The purchasing process is unsuccessful, and the farmer will not be able to continue the process and get the bill page.
Status (Pass/Fail)	Pass
Comments	-

Farmer Purchasing Leftover [1]

Table 5.26 Farmer Purchasing Leftover case

Test Case Name	Successful Purchasing Leftover
Purpose	Verify successful Purchasing Leftover
Component Name	Leftover_shop.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the leftover shop page. 2. Add available items to the cart. 3. Click on the number above the cart icon to navigate to the cart page. 4. Fill in the address information and save it. 5. Review the items added to the cart, and optionally you can choose to apply points earned to the total price for a discount. 6. Click on ‘Proceed to Check Out’ to navigate to the checkout page, review the address and cart items, and choose the preferred payment method. 7. After completing payment process, farmer will be redirected to the bill page.
Input Provided	Address Information, Payment Details, Items to be Purchased.
Expected Outcome	The item is successfully purchased with a confirmation page that displays “Your order has been placed successfully”, and redirected to the bill page
Actual Outcome	The item is successfully shared with confirmation page that displays “Your order has been placed successfully”, “Thank you for shopping with us!”, and “view your order details”.
Status (Pass/Fail)	Pass
Comments	-

Farmer Purchasing Leftover [2]

Table 5.2.27 Purchasing Leftover failure

Test Case Name	Unsuccessful Purchasing Leftover
Purpose	Verify Unsuccessful Purchasing Manure
Component Name	Leftover_shop.php

Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the leftover shop page. 2. Add available items to the cart. 3. Click on the number above the cart icon to navigate to the cart page. 4. Fill in the address information and save it. 5. Review the items added to the cart, and optionally you can choose to apply points earned to the total price for a discount. 6. Click on ‘Proceed to Check Out’ to navigate to the checkout page, review the address and cart items, and choose the preferred payment method. 7. After completing payment process, farmer will be redirected to the bill page.
Input Provided	Address Information, Payment Details, Items to be Purchased.
Expected Outcome	The system denies the purchasing process due to one of the following situations: incomplete address, unsuccessful payment process, or not enough stock.
Actual Outcome	The purchasing process is unsuccessful, and the farmer will not be able to continue the process and get the bill page.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Leftover Shop [1]

Table 5.27 Admin Managing Leftover Shop case

Test Case Name	Successful Managing Leftover Shop
Purpose	Verify successful Managing Leftover Shop
Component Name	admin_leftover_upload.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the managing leftover items page. 2. Fill in the item name, description, quantity, price, and status: unavailable or available. 3. Upload a Photo of the Item 4. Submit items.
Input Provided	Item Name, Description, Quantity, Price, Status, Photo.
Expected Outcome	The item is successfully uploaded with details necessary in the shop that displays “Item uploaded successfully!”.

Actual Outcome	The item is successfully uploaded with a confirmation message that displays “Item uploaded successfully!”, and the item is displayed in the Manage Existing Items table.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Leftover Shop [2]

Table 5.28 Admin Managing Leftover Shop failure

Test Case Name	Unsuccessful Admin Managing Leftover Shop
Purpose	Verify Unsuccessful Admin Managing Leftover Shop
Component Name	admin_leftover_upload.php
Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the managing leftover items page. 2. Fill in the item name, description, quantity, price, and status: unavailable or available. 3. Upload a Photo of the Item 4. Submit items.
Input Provided	Item Name, Description, Quantity, Price, Status, Photo.
Expected Outcome	The system denies the item upload process if any fields are incomplete, the photo extension is invalid, or the item fails to meet database requirements, ensuring that only valid submissions are accepted.
Actual Outcome	The item upload process fails, preventing the item from appearing in the existing items list. The admin will receive a notification displaying either “Failed to upload item” or “Failed to upload image,” providing clarity on the issue encountered.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Manure Shop [1]

Table 5.29 Admin Managing Manure Shop case

Test Case Name	Successful Managing Manure Shop
Purpose	Verify successful Managing Manure Shop
Component Name	admin_manage_products.php
Component Type (method/object)	\$_POST method

Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the managing manure items page. 2. Fill in the item name, description, quantity, price, and status: unavailable or available. 3. Upload a Photo of the Item 4. Submit items.
Input Provided	Item Name, Description, Quantity, Price, Status, Photo.
Expected Outcome	The item is successfully uploaded with details necessary in the shop that displays “New manure item added successfully”.
Actual Outcome	The item is successfully uploaded with a confirmation message that displays “New manure item added successfully!”, and the item is displayed in the Manage Existing Items table.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Manure Shop [2]

Table 5.30 Admin Managing Manure Shop failure

Test Case Name	Unsuccessful Admin Managing Manure Shop
Purpose	Verify Unsuccessful Admin Managing Manure Shop
Component Name	admin_manage_products.php
Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the managing manure items page. 2. Fill in the item name, description, quantity, price, and status: unavailable or available. 3. Upload a Photo of the Item 4. Submit items.
Input Provided	Item Name, Description, Quantity, Price, Status, Photo.
Expected Outcome	The system denies the item upload process if any fields are incomplete, the photo extension is invalid, or the item fails to meet database requirements, ensuring that only valid submissions are accepted.
Actual Outcome	The item upload process fails, preventing the item from appearing in the existing items list. The admin will receive a notification displaying either “Error adding manure item” or “Error uploading the image file”, providing clarity on the issue encountered.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Client Manure Orders [1]

Table 5.31 Admin Managing Client Manure Orders case

Test Case Name	Successful Managing Client Manure Orders
Purpose	Verify successful Managing Client Manure Orders
Component Name	Manage_orders.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the Manage Orders Page: Access the "Manage Client Orders" section in the admin panel. 2. Review Order Details: Check the order list, including status categories: Pending, Processing, Completed, and Cancelled. 3. Update Order Status: Click on the current status and select a new status to update the order.
Input Provided	Order status (Pending, Processing, Completed, and Cancelled).
Expected Outcome	The order's status is successfully updated.
Actual Outcome	The order's status is successfully updated for the client and in the database.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Client Manure Orders [2]

Table 5.32 Admin Managing Client Manure Orders failure

Test Case Name	Unsuccessful Admin Managing Client Manure Orders
Purpose	Verify Unsuccessful Managing Client Manure Orders
Component Name	Manage_orders.php
Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the Manage Orders Page: Access the "Manage Client Orders" section in the admin panel. 2. Review Order Details: Check the order list, including status categories: Pending, Processing, Completed, and Cancelled. 3. Update Order Status: Click on the current status and select a new status to update the order.
Input Provided	Order status (Pending, Processing, Completed, and Cancelled).
Expected Outcome	The order's status is unsuccessfully updated.
Actual Outcome	The order's status is unsuccessfully updated for the client and in the database.

Status (Pass/Fail)	Pass
Comments	-

Admin Managing Farmer Manure Orders [1]

Table 5.33 Admin Managing Farmer Manure Orders case

Test Case Name	Successful Managing Farmer Manure Orders
Purpose	Verify successful Managing Farmer Manure Orders
Component Name	Manage_farmers_orders.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the Manage Orders Page: Access the "Manage Farmers Orders" section in the admin panel. 2. Review Order Details: Check the order list, including status categories: Pending, Processing, Completed, and Cancelled. 3. Update Order Status: Click on the current status and select a new status to update the order.
Input Provided	Order status (Pending, Processing, Completed, and Cancelled).
Expected Outcome	The order's status is successfully updated.
Actual Outcome	The order's status is successfully updated for the farmer and in the database.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Farmer Manure Orders [2]

Table 5.34 Admin Managing Farmer Manure Orders failure

Test Case Name	Unsuccessful Admin Managing Farmer Manure Orders
Purpose	Verify Unsuccessful Managing Farmer Manure Orders
Component Name	Manage_farmers_orders.php
Component Type (method/object)	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the Manage Orders Page: Access the "Manage Farmers Orders" section in the admin panel. 2. Review Order Details: Check the order list, including status categories: Pending, Processing, Completed, and Cancelled. 3. Update Order Status: Click on the current status and select a new status to update the order.

Input Provided	Order status (Pending, Processing, Completed, and Cancelled).
Expected Outcome	The order's status is unsuccessfully updated.
Actual Outcome	The order's status is unsuccessfully updated for the farmer and in the database.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Farmer Leftover Orders [1]

Table 5.35 Admin Managing Farmer Leftover Orders case

Test Case Name	Successful Managing Farmer Leftover Orders
Purpose	Verify successful Managing Farmer Leftover Orders
Component Name	Manage_leftover.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<p>Navigate to the Manage Orders Page: Access the "Manage Leftover Orders" section in the admin panel.</p> <p>Review Order Details: Check the order list, including status categories: Pending, Processing, Completed, and Cancelled.</p> <p>Update Order Status: Click on the current status and select a new status to update the order.</p>
Input Provided	Order status (Pending, Processing, Completed, and Cancelled).
Expected Outcome	The order's status is successfully updated.
Actual Outcome	The order's status is successfully updated for the farmer and in the database.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Farmer Leftover Orders [2]

Table 5.36 Admin Managing Farmer Leftover Orders failure

Test Case Name	Unsuccessful Admin Managing Farmer Leftover Orders
Purpose	Verify Unsuccessful Managing Farmer Leftover Orders
Component Name	Manage_leftover.php
Component Type (method/object)	\$_Post method

Test Description (Steps)	<ol style="list-style-type: none"> 1. Navigate to the Manage Orders Page: Access the "Manage Leftover Orders" section in the admin panel. 2. Review Order Details: Check the order list, including status categories: Pending, Processing, Completed, and Cancelled. 3. Update Order Status: Click on the current status and select a new status to update the order.
Input Provided	Order status (Pending, Processing, Completed, and Cancelled).
Expected Outcome	The order's status is unsuccessfully updated.
Actual Outcome	The order's status is unsuccessfully updated for the farmer and in the database.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Daily Reports [1]

Table 5.37 Admin Managing Daily Reports case

Test Case Name	Successful Managing Daily Reports
Purpose	Verify successful managing daily reports
Component Name	reports.php
Component Type (method/object)	\$_POST method
Test Description (Steps)	<p>Open the admin panel and go to the "Daily Reports" section to access the reports.</p> <p>Review Report Details: Verify that the report corresponds to the current date and review the total number of orders for today.</p> <p>Download Today's Report: Click the 'Download PDF' button to save a copy of today's report for record-keeping.</p>
Input Provided	Clicking on the 'Download PDF' button.
Expected Outcome	The report of today's orders is displayed on the page and can be downloaded as a PDF file.
Actual Outcome	The report of today's orders is displayed on the page, providing a summary of all transactions, and can be easily downloaded as a PDF for record-keeping or further analysis.
Status (Pass/Fail)	Pass
Comments	-

Admin Managing Daily Reports [2]

Table 5.2.39 Admin Managing Daily Reports failure

Test Case Name	Unsuccessful Admin Managing Daily Reports
----------------	---

Purpose	Verify Unsuccessful Managing Daily Reports
Component Name	reports.php
Component (method/object) Type	\$_Post method
Test Description (Steps)	<ol style="list-style-type: none"> 1. Open the admin panel and go to the "Daily Reports" section to access the reports. 2. Review Report Details: Verify that the report corresponds to the current date and review the total number of orders for today. 3. Download Today's Report: Click the 'Download PDF' button to save a copy of today's report for record-keeping
Input Provided	Clicking on the 'Download PDF' button.
Expected Outcome	The report of today's orders could not be displayed on the page due to a system error or missing data.
Actual Outcome	The report of today's orders could not be displayed on the page due to a system error or missing data. As a result, the summary of transactions is unavailable, and the option to download the report as a PDF for record-keeping or further analysis is disabled.
Status (Pass/Fail)	Pass
Comments	-

5.3 Integration Testing

Donate or share your leftover food

Table 5.38 Integration Testing

Test Case Name	Restaurants, clients are allowed to fill the form to share leftover food on their accounts.
Purpose	To enable restaurant and clients post the form of share leftover food
External dependencies	portal.php , login.php , register.php , index.php , client_services.php , restaurant_services.php , restaurant_post_meal.php , client_food_submission.php
Test description (steps)	<ol style="list-style-type: none"> 1. Click on the portal of the Give&Get website 2. Choose restaurant or client based on the user type 3. Registers if you're a new user if you already have an account click log in 4. Then home page will appear to you 5. Hover on your photo a drop-down menu will appear 6. Choose services 7. if you were a client click on share leftover food if you were a restaurant click on donate a meal 8. finally fill out the form then click submit
Expected result	The clients and restaurants can fill the form of share leftover and posted on the website

Actual result	The clients and restaurants can fill the form of share leftover and posted on the website
Status(pass/fail)	Pass
comments	-

Display your previous order or donation

Table 5.3.2 Integration Testing

Test Case Name	All the users (farmer, client, restaurant, non-profit organization) can return to their previous order or donation
Purpose	Link our website with HCI using memorable concept, make users trust our website to safeguard the truth.
External dependencies	portal.php , login.php , register.php , index.php , client_services.php , restaurant_services.php , farmer_services , org_services , client_history.php , farmer_history.php , donations.php , restaurant_meal_history.php
Test description (steps)	<ol style="list-style-type: none"> 1. Click on the portal of the Give&Get website 2. Choose your user type 3. Registers if you're a new user if you already have an account click log in 4. Then home page will appear to you 5. Hover on your photo a drop-down menu will appear 6. Choose previous order 7. Your previous order or donation displays
Expected result	All the previous orders or donations should be displayed.
Actual result	All the previous order or donation should be displayed.
Status(pass/fail)	Pass
comments	-

Display /Edit the user's data

Table 5.39 Integration Testing

Test Case Name	All users (farmer, client, restaurant, non-profit organization) are allowed to edit their personal information.
----------------	--

Purpose	Users are encouraged to review and correct any errors and ensure the accuracy of the information they provide.
External dependencies	portal.php , login.php , register.php , index.php , client_services.php , restaurant_services.php , settings.php
Test description (steps)	<ol style="list-style-type: none"> 1. Click on the portal of the Give&Get website 2. Choose your user type 3. Registers if you're a new user if you already have an account click log in 4. Then home page will appear to you 5. Hover on your photo a drop-down menu will appear 6. Choose sitting 7. Edit/ display your information
Expected result	All users can edit /display their personal information
Actual result	All users can edit /display their personal information
Status(pass/fail)	Pass.
comments	-

Ordering from our shop

Table 5.40 Integration Testing

Test Case Name	Place an order from our website shop.
Purpose	Allowing the user to add the product into shop cart and pay then get discount
External dependencies	portal.php , login.php , register.php , index.php , leftover_shop.php , remove_from_cart.php , orders_success.php , shop.php , credit_card.php
Test description (steps)	<ol style="list-style-type: none"> 1. Click on the portal of the Give&Get website 2. Choose your user type 3. Registers if you're a new user if you already have an account click login 4. Then home page will appear to you 5. Click on the leftover shop 6. Choose the product and quantity you want and add them to shop cart 7. complete your shopping through payment page 8. get discount for your next shopping

Expected result	User can choose /add to cart the product they want and complete payment
Actual result	User can choose /add to cart the product they want and complete payment
Status(pass/fail)	Pass.
comments	-

Nonprofit organizations get donations

Table 5.41 Integration Testing

Test Case Name	Nonprofit organizations are allowed to receive donation from restaurants.
Purpose	Ensure that nonprofit organizations can select the suitable meal and specify the desired quantities.
External dependencies	portal.php , login.php , register.php , index.php , org_services , reserve_meal.php
Test description (steps)	<ol style="list-style-type: none"> 1. Click on the portal of the Give&Get website 2. Choose your user type 3. Registers if you're a new user if you already have an account click login 4. Then home page will appear to you 5. Hover on your photo a drop-down menu will appear 6. Choose services 7. Click on Reserve Meals 8. select the appropriate leftover and choose the desired quantity
Expected result	Nonprofit organization can receive /select the proper leftover and desired quantity
Actual result	Nonprofit organization can receive /select the proper leftover and desired quantity
Status(pass/fail)	Pass.
Comments	-

5.4 Conversion Plan

The Conversion Plan for Give and Get website is a food waste and donation management System aims to conversion the current food donation process into an enhance online platform that maximizes efficiency and community sharing impact on Saudi Arabia and vision of 2030 to reduce food wasting. The plan holds up efforts to decrease food waste, help charities, feed animals with rest of food, and create environmentally friendly manure.

The main objective is to connect donors (restaurants and individuals) with non-profit organizations, farmers, and other beneficiaries in an organized and user-friendly way. The users of Give and Get website could aid in converting food waste into manure and donate food.

The development required for digital solutions aligned with this conversion and donating of food to deal with world challenges like food waste and sustainable farming. By digitizing the process of food waste and donation management, we establish an easy, clear and efficient website that makes all stakeholders happy.

6 System Documentation

This section provides comprehensive documentation for the Give & Get system including user manuals, technical support manuals, and online help resources. The documentation aims to assist users in navigating the system effectively and troubleshooting common issues related to surplus food management.

6.1 User Manual

The user manual serves as a guide for clients, restaurants, farmers, and non-profit organizations using the Give & Get website to manage surplus food and fertilizer needs.

- **Getting Started**

- A. Registration**

1. Visit the Give & Get website.
2. Click on the "Register" button.
3. Select your user type (Client, Farmer, Restaurant, Non-Profit Organization).
4. Fill out the registration form with personal information and identification details.
5. Create a strong and unique password.
6. Click "Sign Up" to complete the registration process.
7. You are now registered

- B. Logging In**

1. After registration approval, you will be directed to the login page.
2. Click the "Login" button.
3. Enter your email and password.
4. Click "Login" to access your account.

- **About Us**

- This page gives you a brief overview of the site, its importance and services

- **User Dashboard**

Manage Profile

1. After logging in, click on your profile icon you can manage your profile
2. Select Profile Settings from the menu
3. Your profile information will be displayed
4. You can update your personal information such as name, email, location, and phone number.
5. You can change your current password

View Previous Orders

1. Click on "Previous Orders" to view your order history.
2. You can see the date, type, and status of each order.
3. Cancel orders if necessary.

Logout

- You can log out of the session

Clients:

1. Create Account & Log In
2. Request Compost
3. Pay Online Orders
4. Share Surplus Food
5. Receive Order and Payment Details.

Farmers:

1. Create Account & Log In
2. Order compost and surplus food
3. Pay for orders online
4. Receive order and payment details.

Restaurants:

1. Create Account & Log In
2. Receive a request for a meal donation (intangible),
3. Send a notification when food is available.

Non-profits:

1. Create Account & Log In
2. Request a food donation
3. Receive the donation they requested.

Access to notifications

1. Check the notifications section to get alerts about surplus food or fertilizers available.
2. Notifications will also inform you of scheduled pickups or deliveries.

- **System Administration:**

Database Access:

1. Administrators have full access to system data via dedicated administration tools.

User Management:

1. Add/edit/delete users and update account information.
2. Receive user updates.

Product Management:

1. Edit, add or delete product data such as (menus and compost).
2. Monitor Orders and Donations: Follow up and confirm orders.
3. Receive monthly reports and donation requests.
4. Technical Support and Dispute Resolution:
5. Administrators have the ability to address technical issues.
6. They can review disputes between users and suggest solutions.

Daily Report Generation:

- Administrators automatically receive daily reports and modify them if necessary.
- All generated or modified reports are stored in a separate reporting database.

Report Receipts:

- After generating a daily report, the administrator receives an electronic receipt confirming the report generation process.

6.2 Technical Support Manuals

In this section, administrators, technical support teams and users will be able to troubleshoot and fix technical issues, the support guide for our current project provides guidance on how to use the system features more effectively.

6.2.1 General Troubleshooting

A. Login Issues

- Issue: Users are unable to log in to their accounts.
- Solution:
 - a. Ensure that the correct email and password are entered.
 - b. Check for internet connection issues.
 - c. Reset the password by clicking the “Forgot Password” link on the login page.
 - d. Contact system support if the issue persists providing the email address and error message.

B. Slow Website Performance

- Issue: The website is loading slowly or experiencing delays.
- Solution:
 - a. Check the server status for any ongoing maintenance or outages.
 - b. Clear your browser cache and cookies.
 - c. Check your internet speed and try accessing from a different network.
 - d. Contact technical support if the issue persists.

C. Problems with ordering or payment

- Problem: Orders are not being processed correctly or payment errors occur.
- Solution:

- a. Ensure that the payment method is valid and connected to the account.
- b. Verify that the user is online while the payment is being processed.
- c. Check the status of the payment gateway for any interruptions.
- d. If the problem persists, notify technical support with detailed information about the order and payment.

6.2.2. User Account Management

A. Account Creation Issues

- Problem: Users are unable to create an account.
- Solution:
 - a. Ensure that all required fields in the registration form are filled out correctly.
 - b. Ensure that the email provided is not already in use by another account.
 - c. Advise the user to check for typos or errors in their email address.
 - d. If the problem persists, contact support regarding account creation issues.

B. Account Suspension or Deactivation

- Problem: Suspend or deactivate the user's account.
- Solution:
 - a. We recommend that you review your system logs for any suspicious activity.
 - b. If there are any violations of the Terms of Service, report the comment to the user.
 - c. If the comment is in error, escalate the issue to an administrator for review.
 - d. Reactivate the account if appropriate.

6.2.3. Notifications and Alerts

A. Missing Notifications

- Problem: Users are not receiving notifications about surplus food or fertilizer.

- Solution:
 - a. Ensure that the notifications feature is enabled in the user profile settings.
 - b. Verify if the notification system is active by reviewing the system backend logs.
 - c. Advise the user to check their email and SMS folders for missed notifications.
 - d. If notifications are not received, escalate the issue to the technical support team for further investigation.

6.2.4. System Administration

A. Access to User Data

- Problem: Administrators cannot access or manage user data.
- Solution:
 - a. Ensure that the administrator has the appropriate access rights and permissions.
 - b. Check the connection settings on the database to ensure proper access.
 - c. If the issue persists, escalate it to the system database administrator for troubleshooting.

B. Product Management

- Issue: Administrators cannot add, edit, or delete product data.
- Solution:
 - a. You should check the administrator's permissions to manage product data.
 - b. Ensure that the database is properly synchronized and updated.
 - c. Resolve any caching issues that may prevent product data updates from appearing on the website.
 - d. If the issue persists, contact the database support team for further assistance.

C. Report Generation Issues

- Issue: Administrators cannot create or edit monthly reports.
- Solution:
 - a. Ensure that the correct report parameters are specified.
 - b. Ensure that the administrator has the necessary permissions to create reports.
 - c. Check if there are any technical issues with the reporting database or server.
 - d. If the issue persists without resolution, notify the technical support team.

6.2.5. Advanced Support

- A. System Downtime or Major Errors**
- Problem: The entire system is experiencing critical downtime or errors.
- Solution:
 - a. Immediately check the server logs to determine the cause of the downtime.
 - b. Check if the problem is related to the hosting provider or internal system components.
 - c. Notify users via email or SMS about the downtime and the estimated resolution time.
 - d. Inform the support or hosting team about the problem

B. Data Recovery and Backup

- Problem: Data loss or corruption in the system.
- Solution:
 - a. Make sure to back up important system data.
 - b. In case of data loss, restore from the latest backup.
 - c. Inform users to ensure that their data is saved in their account profile to prevent.
 - d. If you encounter critical data problems, contact the database management team and inform them about it, so that they can take further action.

6.2.6. Contact Technical Support

- Problem: In case of problems that cannot be resolved using the steps described above.
- Solution:
 - a. You can contact the technical support team either by email or phone and describe the problem in detail.
 - b. You must provide the user information, the issue they are experiencing, and any error messages received. The support team will provide follow-up instructions or a way to raise the issue as needed.

6.3 Online Help and FAQs

- **What is Give & Get website?**

Give & Get is an online platform that offers new ways to support communities to achieve environmental sustainability by reducing food waste and protecting the environment from improper disposal. The Give & Get platform collects surplus food from users and reuses it in useful ways instead of wasting it. Inedible food is converted into agricultural manure and offered for sale, while edible food is redistributed by donating it to charities.

- **How can I register to Give & Get website?**

To register, visit Give & Get website, where you will be greeted with a portal where you must select your proper user type (client, farmer, restaurant, or non-profit organization), and after selecting, you will be able to register and create your account successfully.

- **Is the Give & Get system secure?**

Yes, we protect the privacy of all Give&Get users as we encrypt data and make continuous updates to the system.

- **How can I control my account information?**

To control your account, you must click on the setting option where you will be able to access your account setting, and you can change any of your personal information, then click on update to save the changes successfully.

- **How can I contact customer support?**

You can contact us in two ways, either by calling customer support directly on the phone number written on the footer of the website, or by clicking contact option on the header, then you will be assigned to another page where you must fill in the required fields and write your message and then click send.

- **How can I share leftover food through the website?**

You can share leftover food by signing up as a client, where you can then access the client services and choose the share leftover food option, then you will be forward to another page where you must fill in the required form to complete the process.

- **How can I share untouched food through the website? note: accessible to restaurants only.**

Restaurants can share untouched food by signing up, then access services and choose the donate a meal option, then they will be forward to another page where they must fill in the required form to complete the process.

- **How can I request donations? Note: accessible to nonprofit organizations only.**

Nonprofit organizations can request donations by signing up, then access their services and choose the reserve meals option, then they will be forward to another page where they can see the food posted by the restaurants and can select the meal type and quantity they need.

- **How can I purchase manure or leftover food? note: accessible to farmers only.**

Farmers can purchase manure or leftover food to benefit in gardening. For leftovers, they choose the leftover shop option on the header, then they will be forward to another

page where they can select the leftover type and quantity they need. The same processes apply to manure once they choose the organic manure option on the header, after this they must pay to confirm their purchase.

- **How can I view my previous records?**

Users can easily view their previous records and check their status by accessing their services and choose open previous (requests/purchases/ donations). The option label depends on the type of user registered on the system.

Major Contributions

Project's Idea	Shahad
Abstract	Shahad
1. Preliminary Investigation	
1.1 Problem Statement	Shahad
1.2 Solution	Shahad
1.3 Project Scope	Marwoh
1.4 Feasibility Study	Sokainh
1.5 Functional Requirements	Shaima
1.6 List of Staff's Functions	Shaima
1.7 Data Collection Techniques	Marwoh & Sokainh
1.8 System Requirements	Fatima
1.9 Technologies	Fatima
1.10 Project Timeline	Sokainh
2. System Analysis	
2.1 Context Diagram	Shahad & Sokainh
2.1.1 Context Diagram Description	Shahad, Sokainh & Fatima
2.2 Functional Decomposition Diagram	Shahad, Fatima, Sokainh & Shaima
2.3 Data Flow Diagram	Shahad, Sokainh, Shaima, Marwoh & Fatima
2.3.1 DFD Description	Shahad & Sokainh
3. System Design	
3.1 Input and Output Forms Design	Shaima & Fatima
3.2 Prototype Design	Shaima & Fatima
3.3 Entity Relationship Diagram	Shahad & Shaima
3.4 Normalization	Marwoh & Shahad
3.5 Logical Database Design	Shaima & Fatima
3.6 Physical Database Design	Shahad & Sokainh

4. System Development

4.1 Tools/ technologies selected for implementation	Sokainh & Shahad
4.2 Web system navigation design	Sokainh
4.3 User Interface Design	Sokainh & Shahad
4.4 Database Connectivity	Sokainh

5. Testing and Conversion

5.1 Test Plan	Shaima
5.2 Unit Testing	Shahad
5.3 Integration Testing	Fatimah
5.4 Conversion Plan	Fatimah

6. System Documentation

6.1 User Manual	Marwoh
6.2 Technical Support Manuals	Marwoh
6.3 Online Help and FAQs	Sokainh

Website	
Page	Names
Index.php	Fatimah, Shaima, Shahad, Sokainh
About.php	Everyone
Shop.php	Shahad
Leftover_shop.php	Shahad
Portal.php	Shahad
Reports.php	Shahad
admin_leftover_upload.php	Shahad
admin_manage_products.php	Shahad
admin_panel.php	Shahad
Branch2.php	Sokainh, Shahad
contactUS.php	Marwoh
Footer.php	Sokainh, Shahad

Online_and_faqs.php	Sokainh
manage_farmers_orders.php & orderHandle.php	Shahad
manage_leftover.php & HandlerOrder.php	Shahad
manage_orders.php & orderHandler.php	Shahad
Login.php (admin)	Shahad
Bill.php	Shaima, Shahad
Cart.php	Sokainh, Shahad
Checkout.php	Sokainh, Shahad
client_food_submission.php	Shahad
client_history.php	Shahad
client_services.php	Shaima, Fatimah
org_services.php	Shaima, Fatimah
farmer_services.php	Shaima, Fatimah
restaurant_services.php	Shaima, Fatimah
Credit_card.php	Shaima
Orders_success.php	Shaima
remove_from_cart.php	Shahad
Settings.php	Marwoh, Fatimah, Shahad
touched_food_sharing_history.php	Shahad
Login.php (employee)	Shahad
Checkout_leftover.php	Sokainh, Shahad
farmer_history.php	Shahad
farmer_leftover_history.php	Shahad
leftover_bill.php	Shaima, Shahad
leftover_cart.php	Sokainh, Shahad
leftover_cc.php	Shaima
leftover_remove.php	Shahad
reserve_meal.php	Shahad, Fatimah, Shaima

donations.php	Shahad
approving_requests.php & process_request.php	Shahad, Fatimah
restaurant_meal_history.php	Shahad
restaurant_post_meal.php	Shahad, Marwoh, Fatimah, Shaima
Login.php (farmer, client, restaurant, nonprofit organization)	Everyone
Register.php, registerfarm.php, regusterres.php, registerorg.php	Everyone

References

1. Jereme, I. A., Siwar, C., Begum, R. A., & Talib, B. A. (2016). Addressing the problems of food waste generation in Malaysia. International Journal of Advanced and Applied Sciences, 3(8), 68-77.
2. W. M. K. Trochim, "Knowledge base," Research Methods Knowledge Base. <https://conjointly.com/kb/>
3. "W3Schools.com." <https://www.w3schools.com/>
4. "What is a UML Diagram? | Different Types and Benefits | Miro," <https://miro.com/. https://miro.com/diagramming/what-is-a-uml-diagram/>
5. J. Mahmudovic, "Are surveys qualitative or quantitative? | SurveyLegend," SurveyLegend, Aug. 11, 2022. <https://www.surveylegend.com/types-of-surveys/quantitative-vs-qualitative/>