Project Report

On

“Food Sharing”



**GURU NANAK DEV UNIVERSITY, AMRITSAR**

*For the partial fulfilment of the requirement*

*For the award of degree of*

**“MASTER OF COMPUTER APPLICATIONS”**

(2019-2023)

***Submitted To: Submitted By:***

*“Ms/Mrs/Mr Teacher’s Name”* “Your Name”

(“Designation) **Roll No**. “Roll number”

“Course and Semester)

Department of Computer Science & Applications

Guru Nanak Dev University College

**DECLARATION**

The project report entitled “**Food Sharing**” was submitted by me to Guru Nanak Dev University for the degree of Bachelor in Engineering Sem. VI is the original piece of work and has not been submitted to any other university for the award of any degree. I also undertake that any quotation or a philosophy from the published or unpublished work of another person has been duly acknowledged in the work that I present in the project report.

#### **Place: GURU NANAK DEV UNIVERSITY**

#### **Signature of the Student:**

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **ACKNOWLEDGEMENT**

I have accumulated a large number of debts in preparing this project. While a brief acknowledgment here in no way writes them off, it is a small courtesy whose sentiments are sincere. I would like to extend my sincere thanks to all the people who helped me in different ways with the development of this project report. Without their continuous support and guidance, the completion of my project would have been impossible.

I extend my gratitude to “HOD’s Name”, Head of the Department, “Department Name”, for his continuous support and encouragement throughout the degree. I also wish to express my most sincere thanks to my supervisor **“Name of Supervisor’** for her invaluable guidance, advice, support and encouragement. I will carry out her guidance throughout my life.

I shall be falling behind in my duties if I do not place on record my sincere thanks to all those writers and authors from whose writings I have benefited.

I would like to express my special gratitude and thanks **O7 Services** Staff for giving me such attention and time.

In the end, I would also like to mention that this project would not have been possible but for the continuous support and guidance of my parents who gave me the strength and will to succeed. My thanks and appreciations also go to my friends in developing the project and the people who have willingly helped me out with their abilities.

Name of Student

**CHAPTER-1 Introduction**

**1.1 INTRODUCTION:**

The objective of the project is to provide designing and development. The website has three panels; Admin, Donor and Receiver. In this, the receiver can receive the selected food items from the selected category and can buy it from the donor with the admin's validation of admin. The admin can categorize the food items and can view requests. The request can be accepted or declined by the admin.

This project helps to meet the requirements of the receiver and to provide more food to people in need. The project is time saver and provides a wide range of availability.

### **1.2 PROBLEM DEFINITION**

The system is designed to reduce the manual workload of giving food to the required persons. The manual process is quite hectic and time-consuming. By using the conventional system, people need to note down the entire data and can’t qualify the items to be donated by which there is a lot more chance of error.

The application is used to reduce the time consumption and reduce errors. The website can store the data on the internet. .It also reduces the time of people. In the past, many people noted down all data on papers and that was not reliable, because it can be lost.

**EXISTING SYSTEM:**

The current food donation app only serves as an interface that bridges gaps between only the people who are trying to donate and NGO organizations. No one focused on the user side and user interest. So, by this proposed system we can increase the interest of users to donate food. Also, the conventional methodology is somewhat restraining the people to come forward as they don’t have such amount of time.

**1.3 PROPOSED SYSTEM:**

The proposed system consists of three panels namely Admin, Donor and Receiver. The donor can donate the food items which are categorized by the Admin. Receivers can login to their account and check the food categories and items to be sold. The Receivers can choose from wide range of varieties and can request for that particular food item easily.

In the proposed system, we proposed to computerize the above-mentioned activities. The main points focused in our proposed system are:

**There are 3 Module:**

* **ADMIN**
* **DONOR**
* **RECEIVER**

**ADMIN:**

Admin needs to first login to their account and can add the NGOs from which food is to be sold. The food is categorized and items are added which are to be donated. This is the main module of this website. He/she can view the requests submitted by the receivers and can accept or decline the requests generated by them.

**DONOR:**

Donor needs to login to their account and if he/she is not registered then they need to do registration and log in to their account. After logging in, they need to select the category of the food and then food items in which they want to donate their food. Not every type of food can be donated as it is supervised closely by the Admin.

**RECEIVER:**

Any person who wants to receive the food needs to register their account and log in from it. After logging in, they can check the categories of the food and can view the food items. After selecting a particular food item, they can select a specific food item that they want. This request is then viewed by the admin who can accept or decline the generated request.

**FEATURES**

* User friendly interface
* Fast access to database
* Less error prone
* Less hectic
* Search facility
* Friendly look and feel

**CHAPTER-2 Hardware & Software Requirements**

For this project minimum hardware and software requirement are listed below:

**2.1 Hardware Requirements:**

**Processor** **:** Intel X86

**RAM**  **:** 512MB/8 GB

**Hard Disk**  **:** 50 GB

**2.2 Software Requirements:**

**Front End Tool** **:** HTML, CSS, JAVASCRIPT, BOOTSTRAP

**Web Server :** PHP Server 8.0.0 (or Above).

**Back End Tool :** MYSQL

**Browser :** IE 7.0/Mozilla Firefox 6.0/Cross

**Operating System :** Windows Operating System/Linux

**Chapter 3**

**FEASIBILITY STUDY**

The idea behind the educational guide is to promote the demand of students for the best services regarding their time table, fee structure as a necessary complement to the effort to improve the home and class work of the students as well as the overall development of the student.

It provides information of all the students which are registered in the explicitly so that the students get the best of our facilities.

With the help of this project, a student can find a better platform and website according to his professional school studies. He can also record their data for future use. He can easily handle his own personal details. It can save the time of a student.

**3.1 ECONOMIC FEASIBILITY**:

It is economically feasible, it is free to use, the current user is responsible for entering the data into the database via a user interface provided to him, he/she can also able to show all the data in html tabular form.

**3.2 TECHNICAL FEASIBILITY:**

It is technically feasible, since the whole system is designed into the technologies like

Html. Css3, Javascript, Jquery , Ajax and JSON which are the most recent technologies to develop web based systems.

**3.3 BEHAVIOURAL FEASIBILITY**:

It is Behavioural feasible, since the system is providing a attractive user interface to the Operator /end user, so he feels very easy to work on it. Response to operator/end user is very fast and very good. Since, as we mentioned above, it requires much less cost, it uses computer work so it is very fast to operate and it is very easy for users to work on it.

**3.4 METHODOLOGY /PLANNING OF WORK**

The main objectives of our project are:

1). Admin checks all records of registered users.

2). Users can store the information like course details, Address etc.

3). Admin can manage the time table.

4). Students check their homework.

5). All users can edit and update their personal details.

6). Easy to maintain the record and save their time.

**3.5 Use Case Diagrams**

Use cases are used during the analysis phase of a project to identify system functionality. They separate the system into actors and use cases. Actors represent roles that are played by users of the system. Users may be humans, other computers, or even other software system.

**CHAPTER 4**

**SYSTEM ANALYSIS**

**4.1 DATA ANALYSIS**

Before developing this project, we first analyze the existing system of study. In the existing system all greetings are given manually. As we know, nowadays computers are used in every field. We can remove the manual work by using an automatic system. We see first that if it is feasible or not whether technically, economically, or operationally. We test whether it properly works or not. Its technical requirements are feasible or not. We analyzed the system properly and then started designing it. After Designing, we implement this project to see whether this project works properly or not. After implementing the project, we check whether there is any problem for the user while using this project. Prior to stating whether the system we have to develop is feasible or not we believe that we should emphasize on what is implied by the word “Analysis”. Analysis is the measure of how beneficial or practical the Development of the system will be to the organization. It is a preliminary survey for the system's investigation. It aims to provide information to facilitate a later in-depth investigation.

**Types**

There are various measures of analysis that helps to decide whether a particular project is feasible or not.

These measures include –

* + Operational Analysis
  + Technical Analysis
  + Economical Analysis

Each of these types will be explained in detail throughout the project report

**Operational analysis**

A proposed system is beneficial only if it can be turned into an information system that will meet the operational requirements of an organisation. A system often fails if it does not fit within existing operations and if users resist the change.

Important issues a systems developer must look into are: Will the new system be used if implemented in an organisation?

Are there any major barriers to implementation or is the proposed system accepted without destructive resistance?

The whole purpose of computerising it is to handle the work much more accurately and efficiently with less time consumption. There will be additional work to be completed, because now the website will have to maintain a database of both their admins as well as their Customers. Compared to the semi-computerized system the chances of avoiding errors in a computerised system is much higher because the user need not stress himself unnecessarily resulting in recklessness. Unlike the semi-computerized system there would be backup data for all the information concerning the daily transactions. Another important fact to be regarded is the security control, which is handled by the system. Since data regarding each Customer is confidential, security is a key issue. Information falling into the wrong hands could jeopardise the entire website organisation. Unlike in semi-computerized systems, the proposed system offers adequate control to protect against fraud and embezzlement and guarantees the accuracy and Security of data and information. This is handled by the system providing individuals with separate login names and passwords. The new system is user-friendlier, which enables the end-user to complete his/her work efficiently and accurately with interest. After taking the above fact into consideration we can state the operating of the proposed system is feasible.

**Economical Analysis**

In making recommendations a study of the economics of the proposed system should be made. Even though finding out the costs of the proposed project is difficult we assume and estimate the costs and benefits as follows. According to the computerised system we propose, the costs can be broken down in two categories.

1. Costs associated with the development of the system.
2. Costs associated with operating the system.

**Chapter 5**

**Technology used**

**5.1 HTML**

HTML Stands for HyperText Markup Language, where

* HyperText stands for Link between web pages.
* Markup Language means Text between tags that define the structure.

HTML is a markup language that is used to create web pages. It defines how the web page looks and how to display content with the help of elements. It forms or defines the structure of our Web Page, thus it forms or defines the structure of our Web Page. We must remember to save your file with .html extension.

**5.2 CSS**

**C**ascading **S**tyle **S**heets, fondly referred to as **CSS**, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

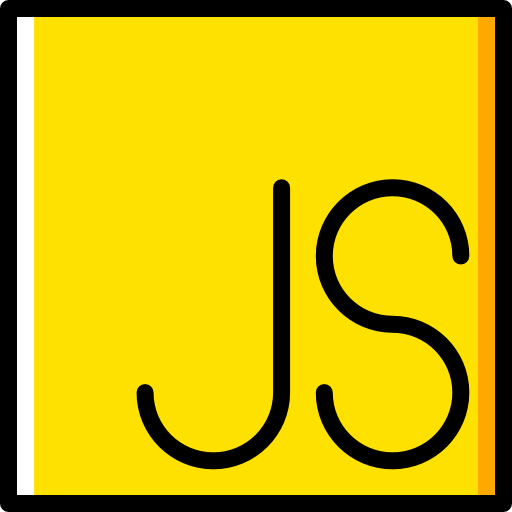


**WHY CSS?**

* **CSS saves time:** You can write CSS once and reuse the same sheet in multiple HTML pages.
* **Easy Maintenance:** To make a global change simply change the style, and all elements in all the webpages will be updated automatically.
* **Search Engines:** CSS is considered a clean coding technique, which means search engines won’t have to struggle to “read” its content.
* **Superior styles to HTML:** CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Offline Browsing:** CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.

**5.3 Javascript**

JavaScript is a lightweight, cross-platform, and interpreted scripting language. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments. JavaScript contains a standard library of objects, like Array, Date, and Maths, and a core set of language elements like operators, control structures, and statements.



* **Client-side:** It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as **mouse clicks**, **form input**, and **page navigation**. Useful libraries for the client-side are **AngularJS**, **ReactJS**, **VueJS** and so many others.
* **Server-side:** It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is **node.js**.

JavaScript can be added to your HTML file in two ways:

* **Internal JS:** We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
* **External JS:** We can write JavaScript code in another file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

**5.4 Introduction to PHP**

**What did you learn?**

**PHP** is a general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an [HTML](http://en.wikipedia.org/wiki/HTML) source document rather than calling an external file to process data. The code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a command-line interface capability and can be used in standalone graphical applications

**Advantages of PHP**

* PHP is accessible
* It's available for free
* It's available with documentation in many languages
* There are many support groups, forums, and teams supporting PHP
* There is a wealth of online information regarding PHP
* It's quick to develop in PHP
* PHP is loosely typed, which makes basic scripts much faster to develop with less attention to design
* PHP is flexible. Use OOP or not. Use naming convention(s) or not
* It runs on many different operating systems
* It can be optimised, even "compiled" for performance closer to that of more established compiled languages.

**What is the scope of PHP?**

* PHP is basically a scripting language used for web development. The websites created by PHP are dynamic and attractive
* Scope in PHP is really high as PHP is a language known in the world of technology for many years. So it has gained the maximum popularity in this era.

**5.5 Bootstrap**

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.



Why Bootstrap?

* Faster and Easier Web Development.
* It creates Platform-independent web pages.
* It creates Responsive Web-pages.
* It is designed to be responsive to mobile devices too.
* It is Free! Available on www.getbootstrap.com

How to use Bootstrap 4 on a webpage: There are two ways to include Bootstrap on the website.

* Include Bootstrap from the CDN link.
* Download Bootstrap from getbootstrap.com and use it.

**5.6 Xampp server**

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the Apache friends, and its native source code can be revised or modified by the audience. It consists of Apache HTTP server, Maria DB, and interpreters for different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as IA-32 package of windows and the x64 package of macOS and Linux.

What is XAMPP?

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stand for MySQL and Ps stands for PHP and Perl. It is an open-source package of web solutions that includes Apache distribution of many servers and common-line executable along with modules such as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computer and laptop before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB Is the most vividly used database developed by MySQL.

**5.7 MYSQL**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as Foreign Keys.

A Relational DataBase Management System (RDBMS) is a software that −

* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

## **RDBMS Terminology**

Before we proceed to explain the MySQL database system, let us revise a few definitions related to the database.

* Database − A database is a collection of tables, with related data.
* Table − A table is a matrix with data. A table in a database looks like a simple spreadsheet.
* Column − One column (data element) contains data of one and the same kind, for example the column postcode.
* Row − A row (= tuple, entry or record) is a group of related data, for example, the data of one subscription.
* Redundancy − Storing data twice, redundantly to make the system faster.
* Primary Key − A primary key is unique. A key value cannot occur twice in one table. With a key, you can only find one row.
* Foreign Key − A foreign key is the linking pin between two tables.
* Compound Key − A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* Index − An index in a database resembles an index at the back of a book.

* Referential Integrity − Referential Integrity makes sure that a foreign key value always points to an existing row.

## **MySQL Database**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

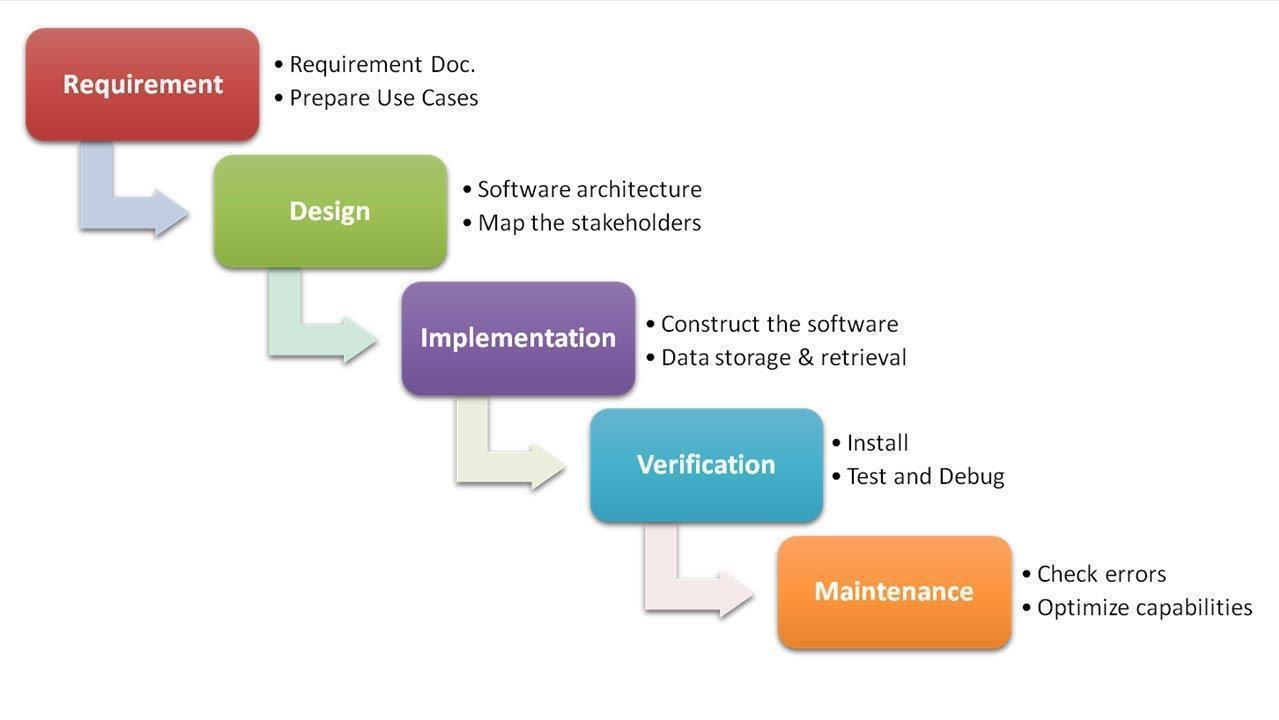
**CHAPTER 6**

**SOFTWARE PROCESS MODEL**

**Waterfall Model**

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Requirement Analysis, Design, Coding, Testing, Deployment, and Maintenance .

Following is a diagrammatic representation of different phases of the waterfall model.



The sequential phases in Waterfall model are:

* **Requirement Gathering and analysis**

All possible requirements of the system to be developed are captured in this phase. Requirements are a set of functionalities and constraints that the end-user (who will be using the system) expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be developed is also studied. Finally all requirements documented in a requirement specification doc.

* **System Design**

Before starting for actual coding, it is highly important to understand what we are going to create and what it should look like? The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

* **Implementation**

With inputs from system design, the work is divided into modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. Unit testing mainly verifies if the modules/units meet their specifications.

* **Integration and Testing**

All the units developed in the implementation phase are integrated into a system after testing of each unit. These units are integrated into a complete system during Integration phase and tested to check if all modules/units coordinate between each other and the system as a whole behaves as per the specifications. Post integration the entire system is tested for any faults and failures.

* **Maintenance**

This phase of "The Waterfall Model" is virtually never ending. There are some issues which come up in the client environment. Not all the problems come in picture directly but they arise from time to time and need to be solved. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for the previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap. The Waterfall model is the earliest SDLC approach that was used for software development.

**Waterfall Model Application**

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.

## **Waterfall Model Pros & Cons:**

## **Advantage**

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

### **Disadvantage**

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage. Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model.

**Why do we use the waterfall model?**

As it is a major project and being a beginner, we already have the requirements for our ongoing project. Waterfall model is considered to be of a downward approach and we don’t have to look up to the previous level that frequently, it’s beneficial for our project to complete it in a timely manner. Thus if we want to modify anything within our project after deployment, we can start from the initial phase. Thus it does not freeze the possibility for any kind of change.

**CHAPTER 7**

**DESIGN**

**7.1 SYSTEM DESIGN**

The most creative and challenging phase of SDLC is system design. The term design describes a final system and the process by which it is developed. It includes construction of programs and program testing.

The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution domain. Starting with what is needed; design takes us towards how to satisfy the needs. The design of the system is perhaps the most critical factor affecting the quality of the software. It has a major impact on the later phase, particularly testing and maintenance. The output of this phase is the design document. This document is similar to the blueprint or plan for the solution and is used later during implementation, testing and maintenance.

A systematic method has to achieve the beneficial result at the end. It includes starting with an average idea and developing it into a series of steps. The series of steps for successful system development are given below:

* Study the problem completely because first of all we should know the goal, which he has to achieve.
* We should see what kind of output we require and what kind of input we give so we can get the desired output from the system. It is a very challenging step of system development.
* According to the output requirement of the system the strength of various databases should be designed.
* Next, we should know what kind of program we should develop, which will lead us to reach our final goal.
* Then we write this individual program, which later on joining will solve the problem.
* Then we test these programs and make necessary corrections in them to achieve the target of the program.
* At last combining all these problems in the forms of a bar in the menu of windows, this will complete the software package for general insurance.

The three main objectives which the designer has to bear in mind are:-

1. How fast the design will be does the users work given particular hardware resources.
2. The extent to which the design is secure against human errors and machine malfunctions.
3. The ease with which the design allows the system to be changed.

**To meet these objectives analysts and programmers use a top-down and bottom-up design.**

* **TOP – DOWN DESIGN**

It is also known as system design, and aims to identify the modules that should be in a system. It starts with a large picture and moves to the details. The analyst and team members look at major functions that the system must provide and break these down into smaller and smaller activities.

* **BOTTOM – UP APPROACH**

It is also known as detailed design. It starts with details and then moves to the big picture. This approach is appropriate when users have specific requirements for output.

**CHAPTER 8**

**DFD: Data Flow Diagram**

Data Flow Diagrams were first developed by Larry Constantine as a way of expressing system requirements in a graphical form. DFD is also known as bubble chart and has a purpose of clarifying system requirements and identifying major transformations and will become the program in the system design.

Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destinations.

**Purpose:**

The purpose of data flow diagrams is to provide a semantic bridge between users and systems developers.

The diagrams are:

* Graphical, eliminating thousands of words.
* Logical representations, modeling WHAT a system does, rather than physical models showing HOW it does it.
* hierarchical, showing systems at any level of detail and
* Allowing user understanding and reviewing.

**DFD Symbols are as follows:**

* The External Entity symbol represents sources of data to the system or destinations of data from the system.

* The Data Flow symbol represents movement of data.



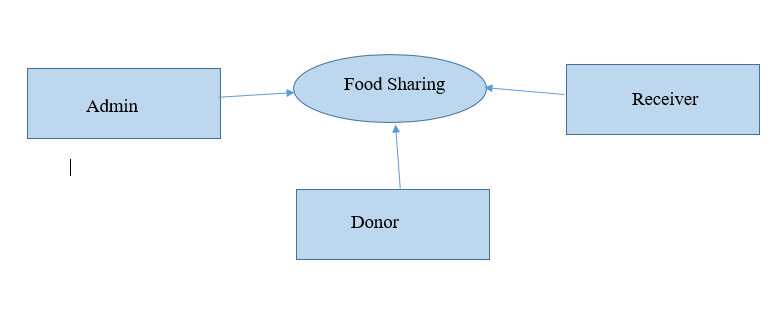


* The Data Store symbol represents data that is not moving (delayed data at rest).

* The Process symbol represents an activity that transforms or manipulates the data.

**DFD Level 0**

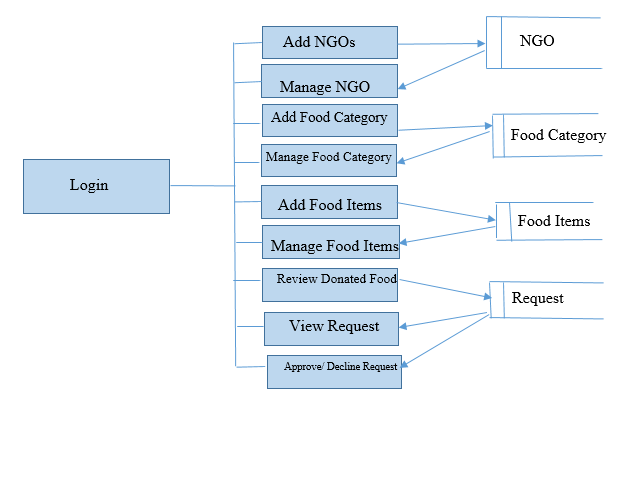
**Context Level Diagram**



Here User and Admin interact with the system for different purposes. Database contains all the information which users need.

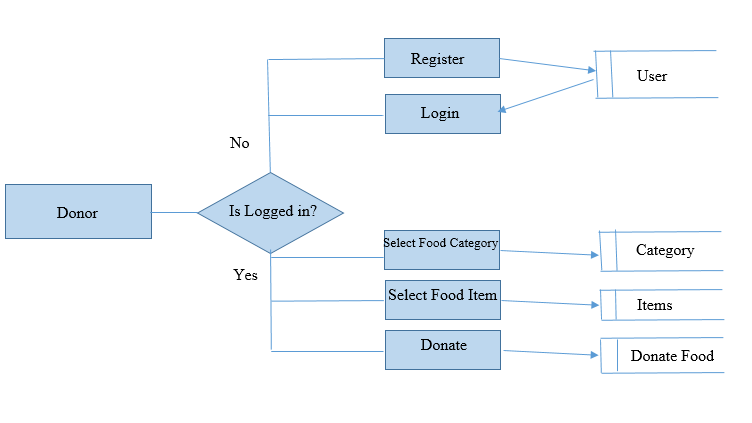
**Level 1 DFD:**

**DFD For Admin**



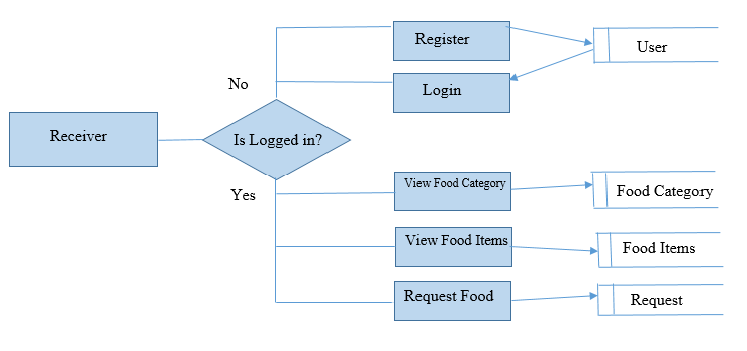
**Level 1 DFD:**

**DFD For Donor**

****

**Level 1 DFD:**

**DFD For Receiver**

****

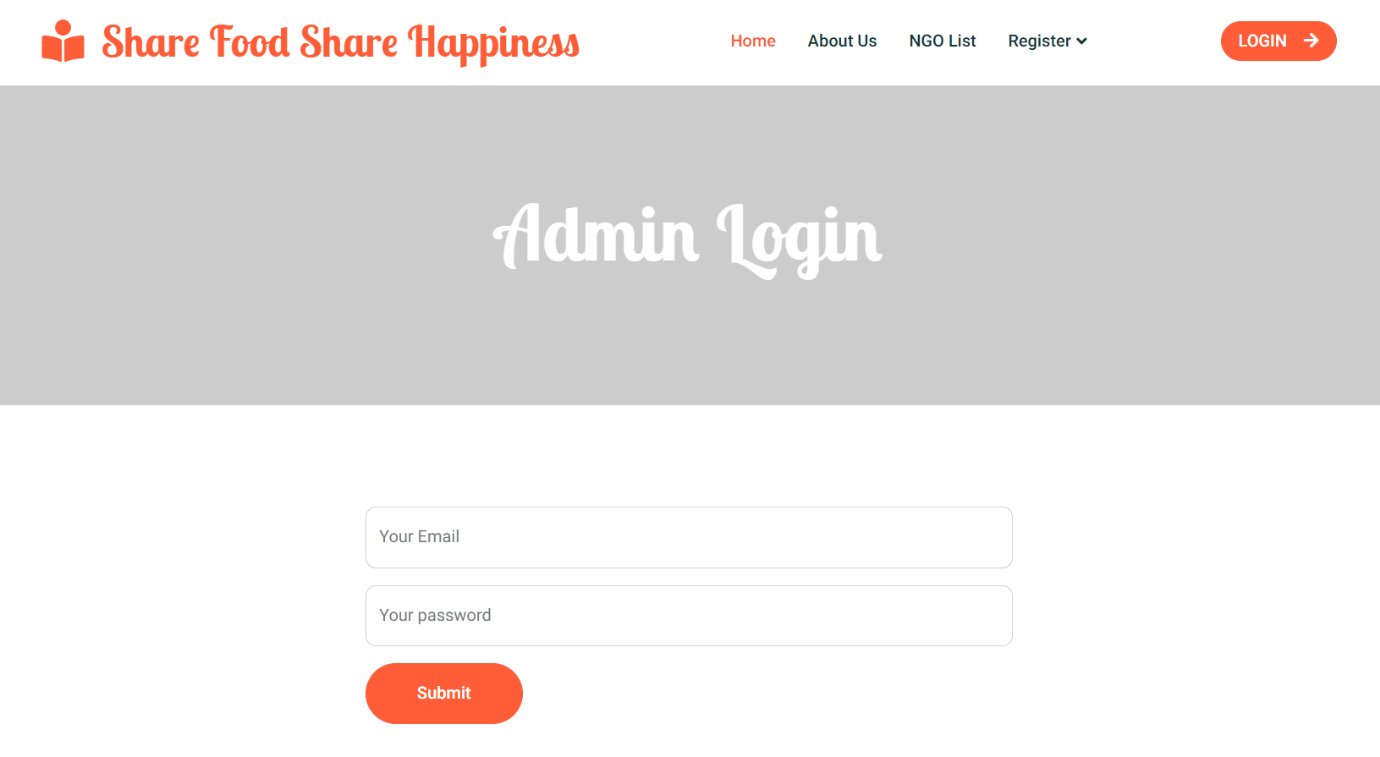
**CHAPTER 9**

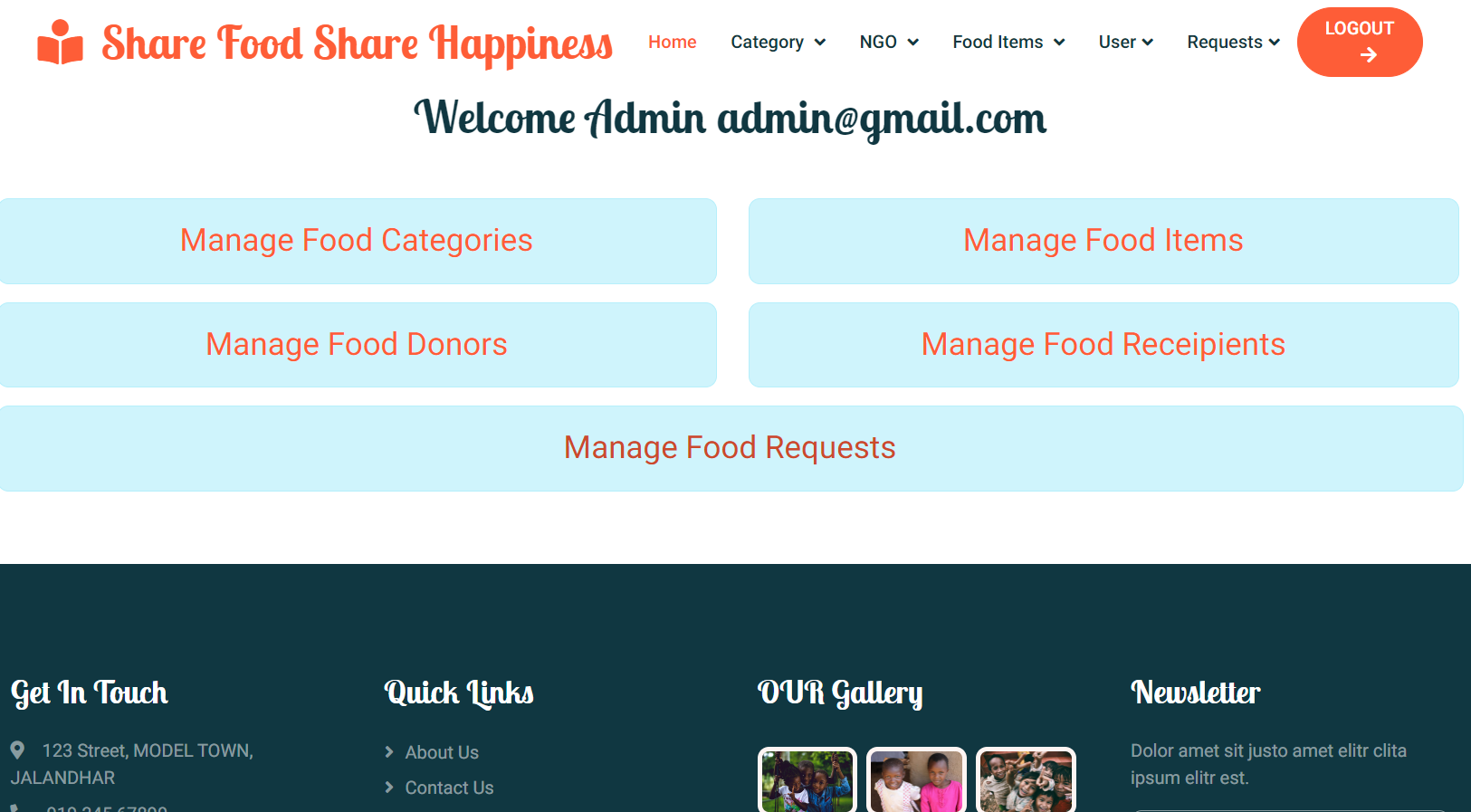
**SNAPSHOTS**

**9.1 Admin Panel**

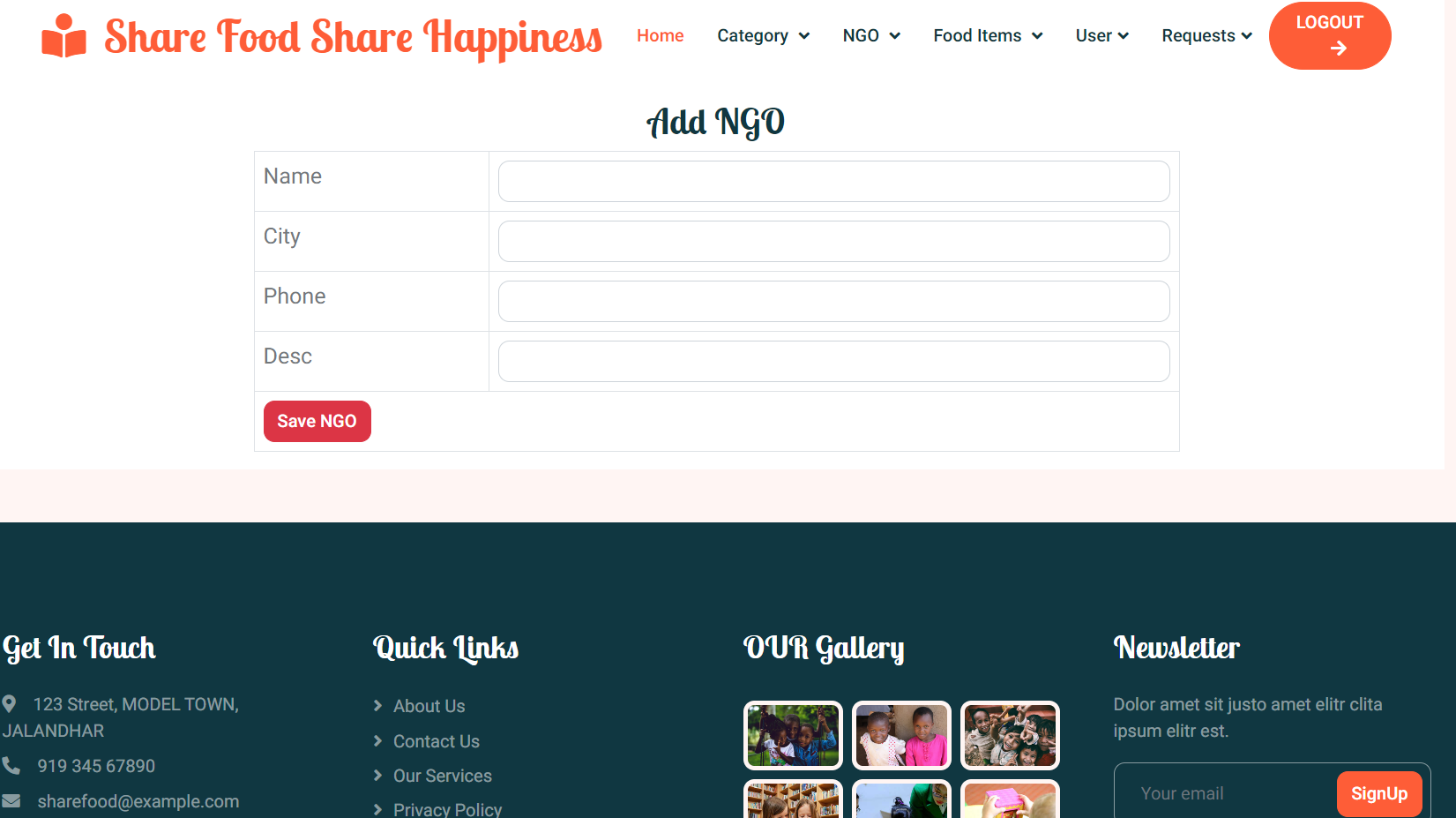
****

**9.2 Admin Login**

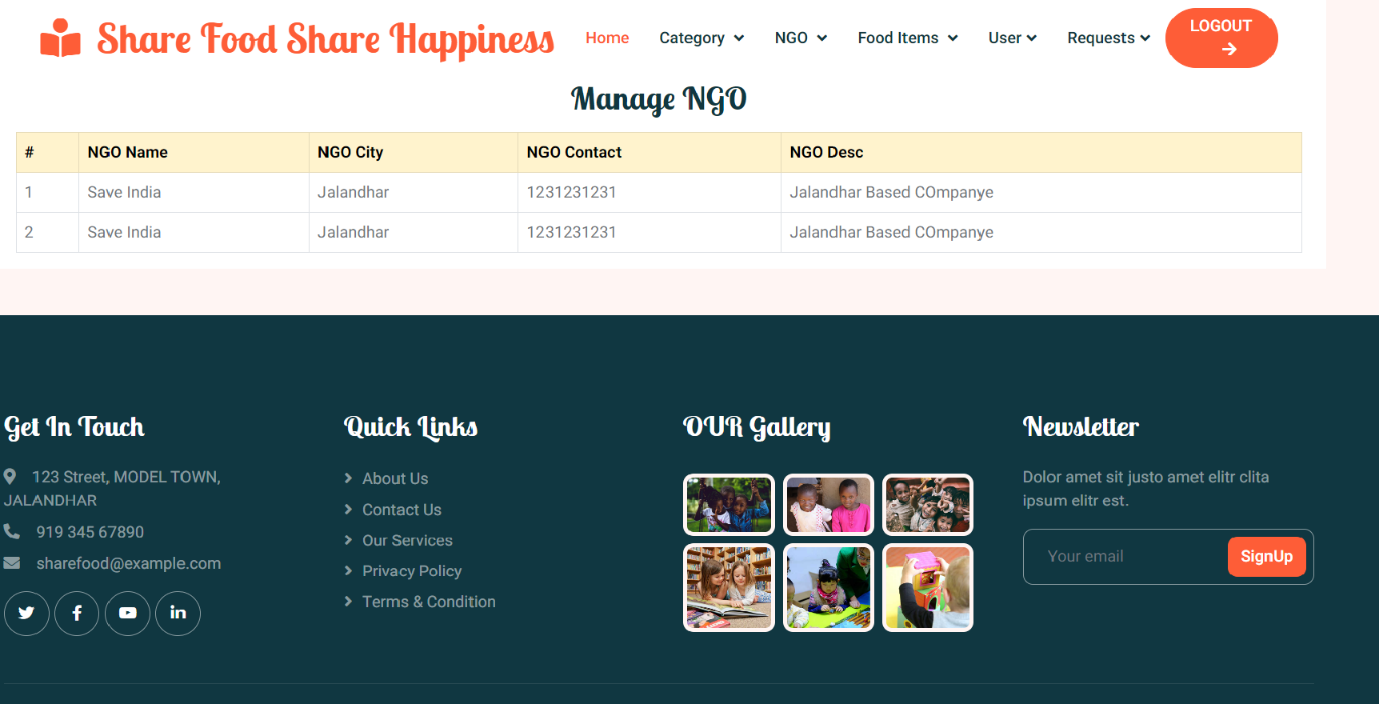
****

****

**9.3 Add NGO**

****

**9.4 Manage NGO**

****

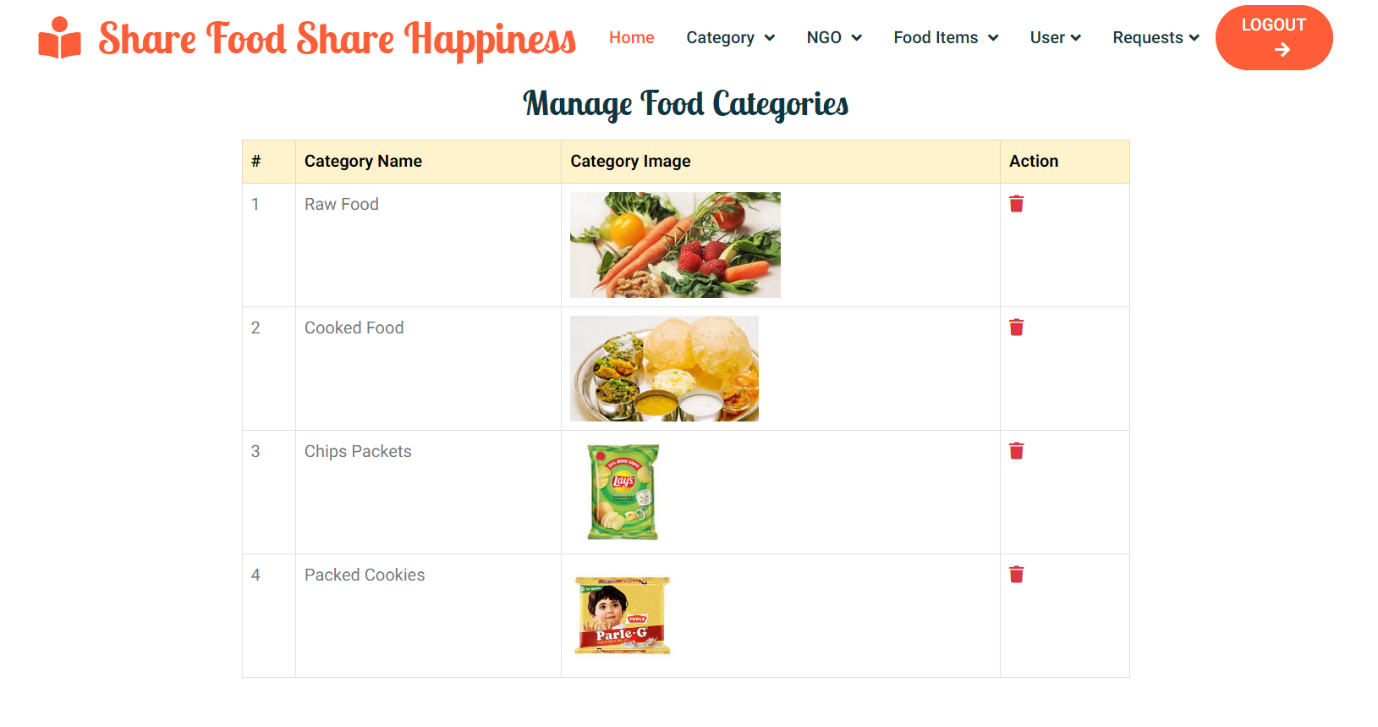
**9.5 Add Food Category**

****

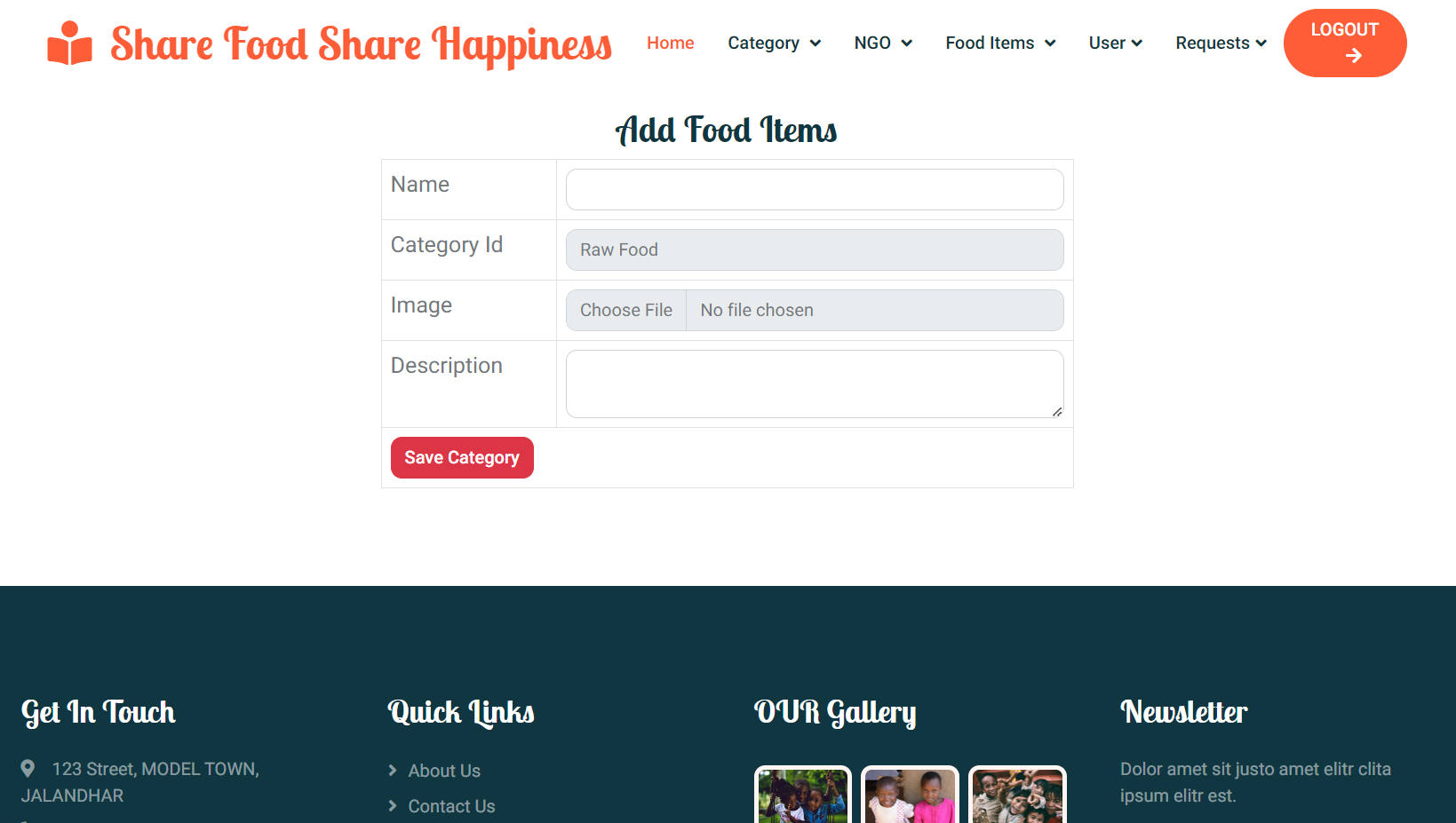
**9.6 Manage Food Category**

****

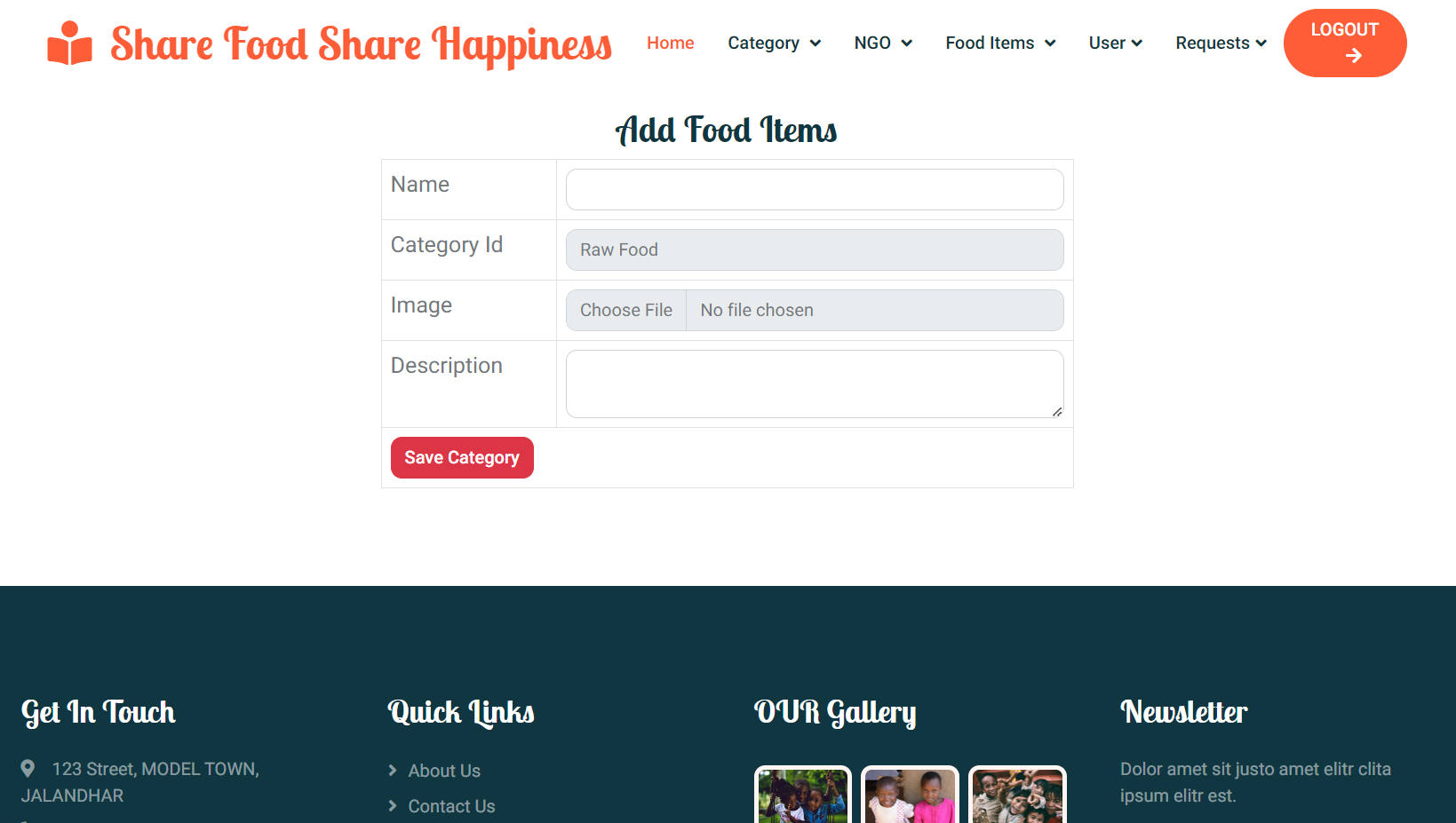
**9.7 Manage Food Category**

****

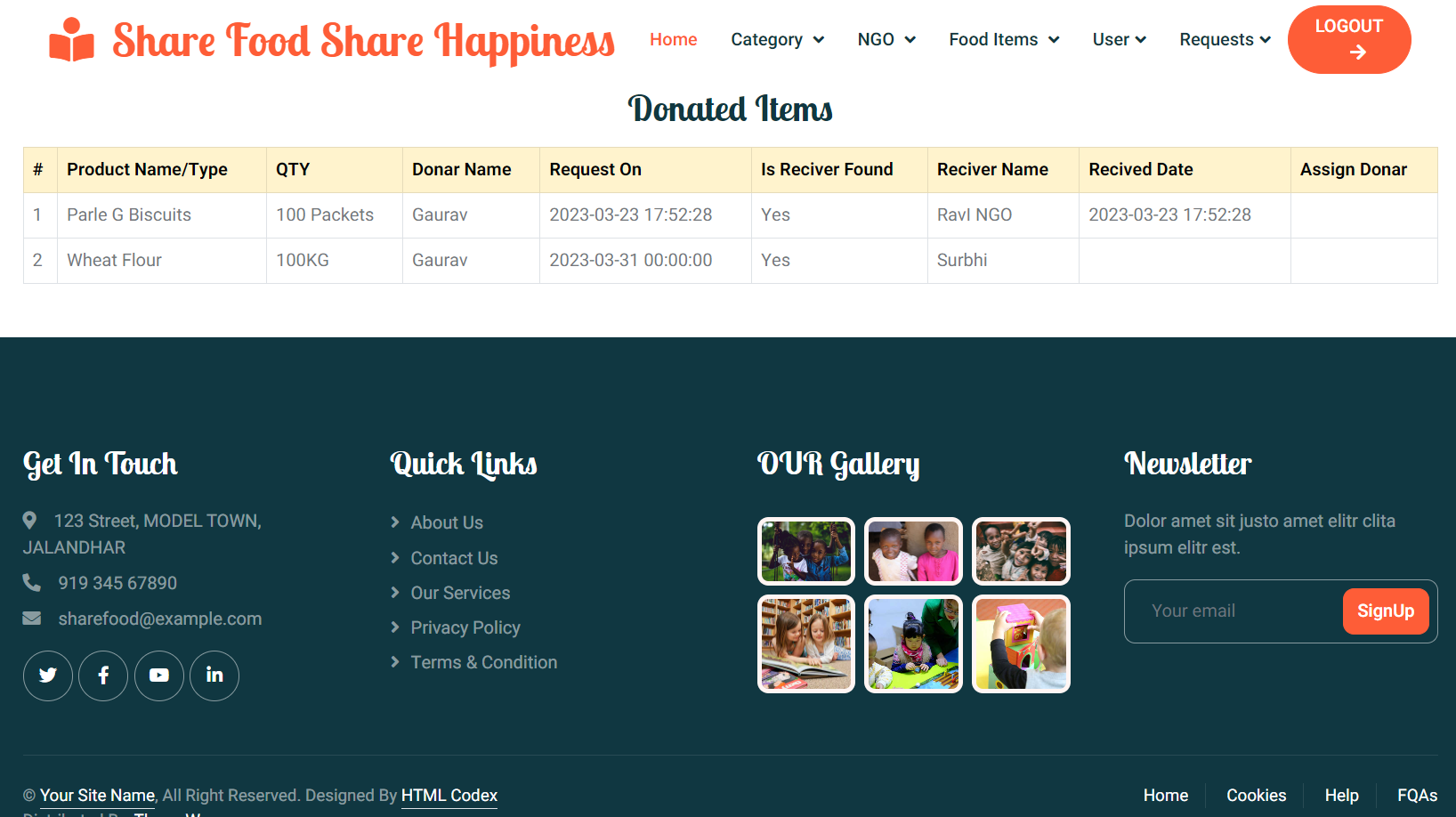
**9.8 Add Food Items**

****

**9.9 Donor List**

****

**9.10 Donated Items**

****

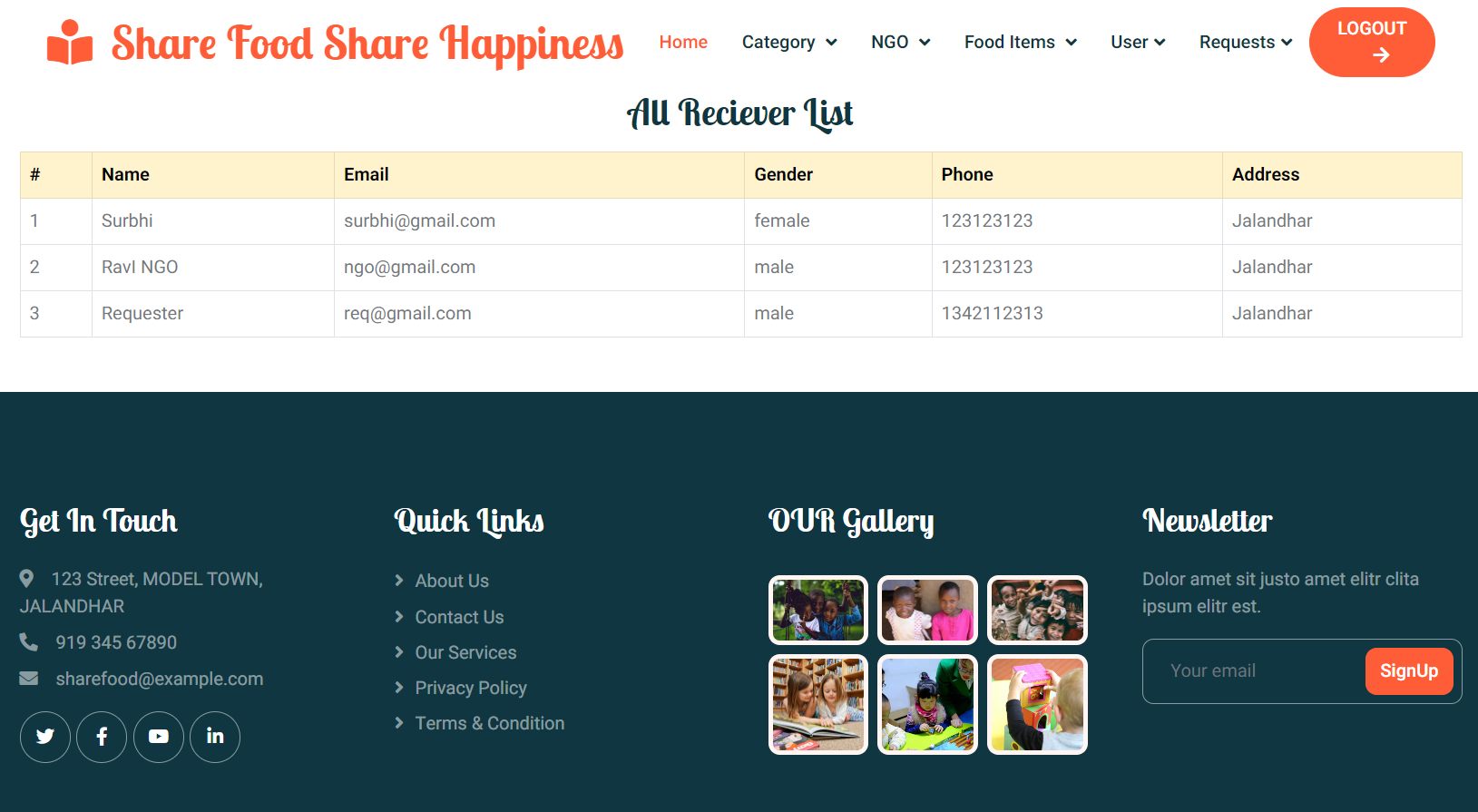
****

**9.11 Request**

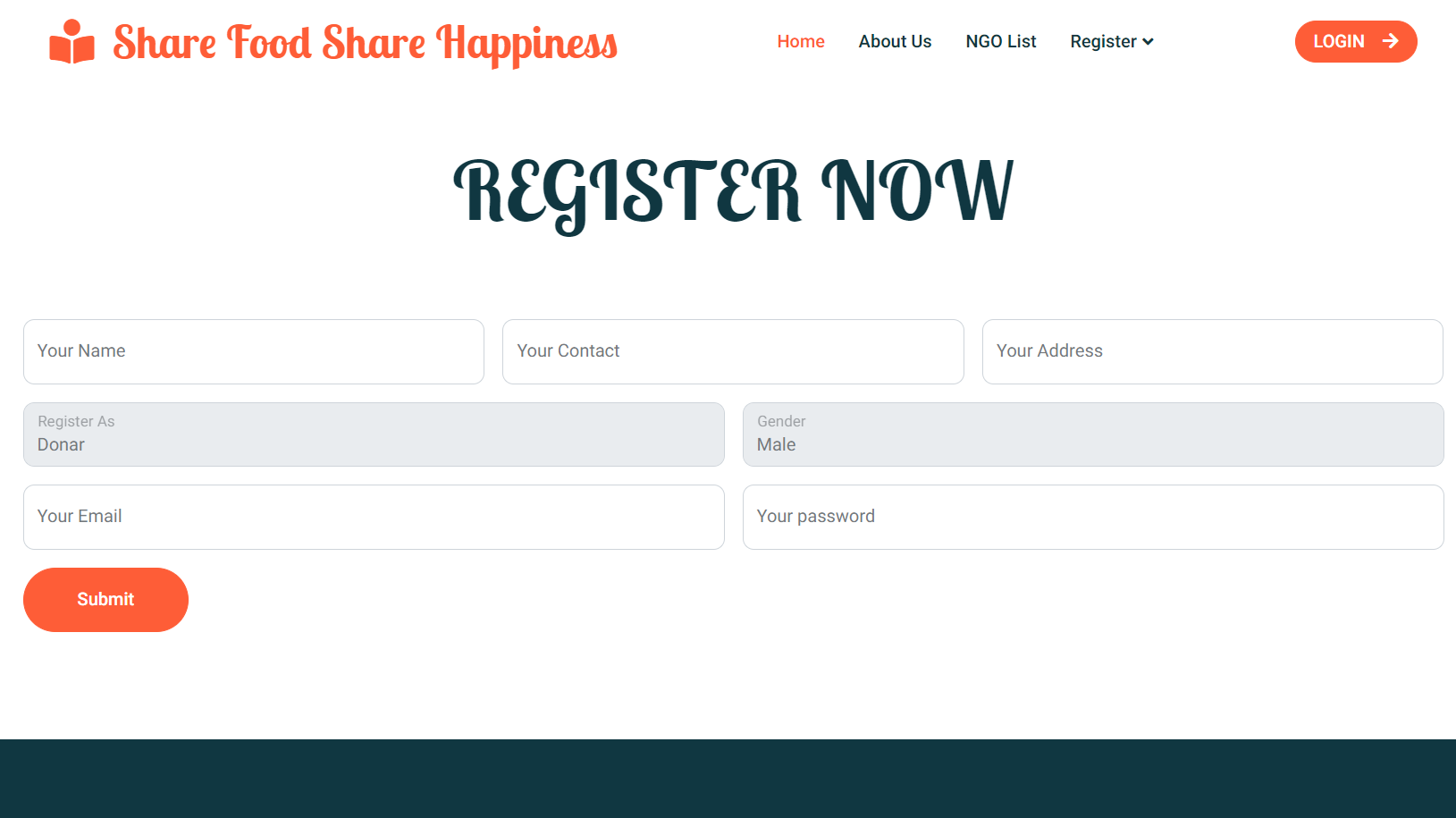
****

****

**9.12 Receiver List**

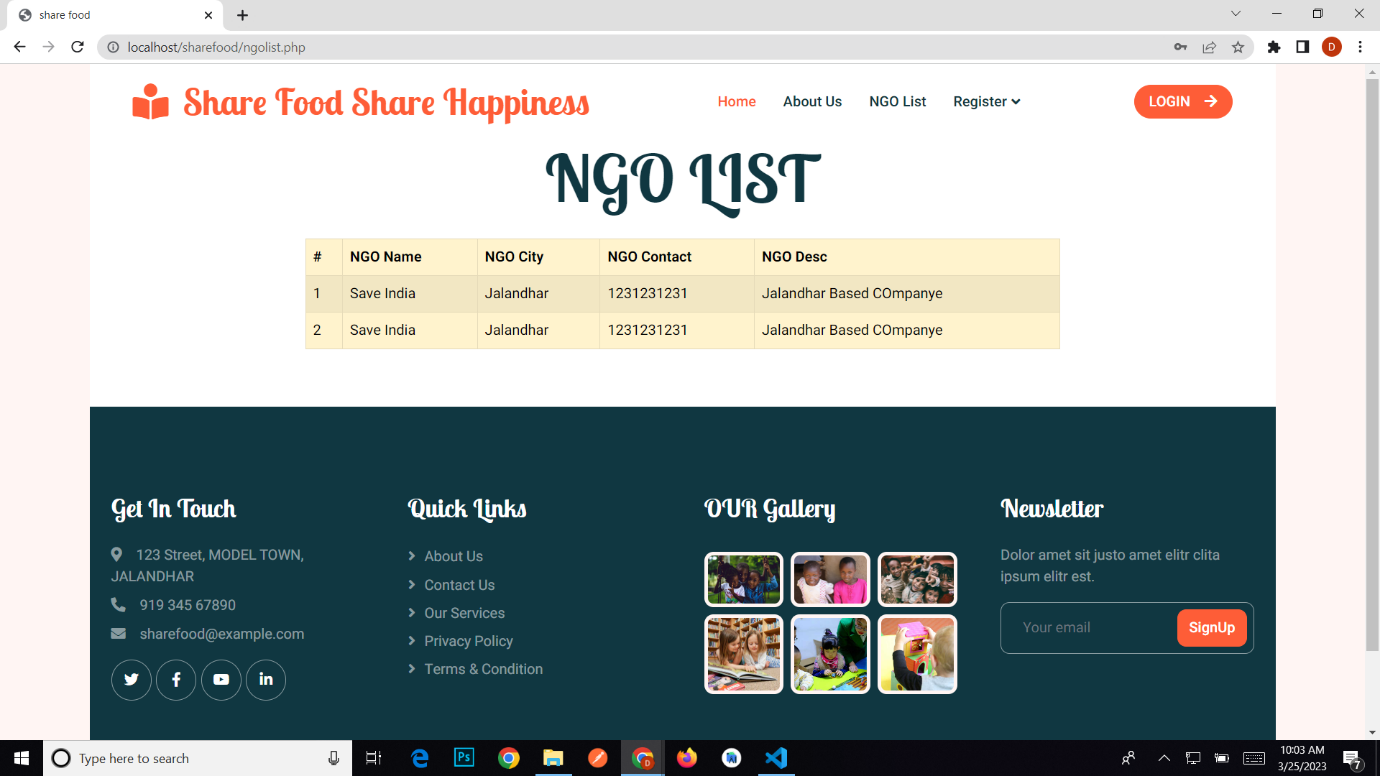
****

**9.13 Donor Register**

****

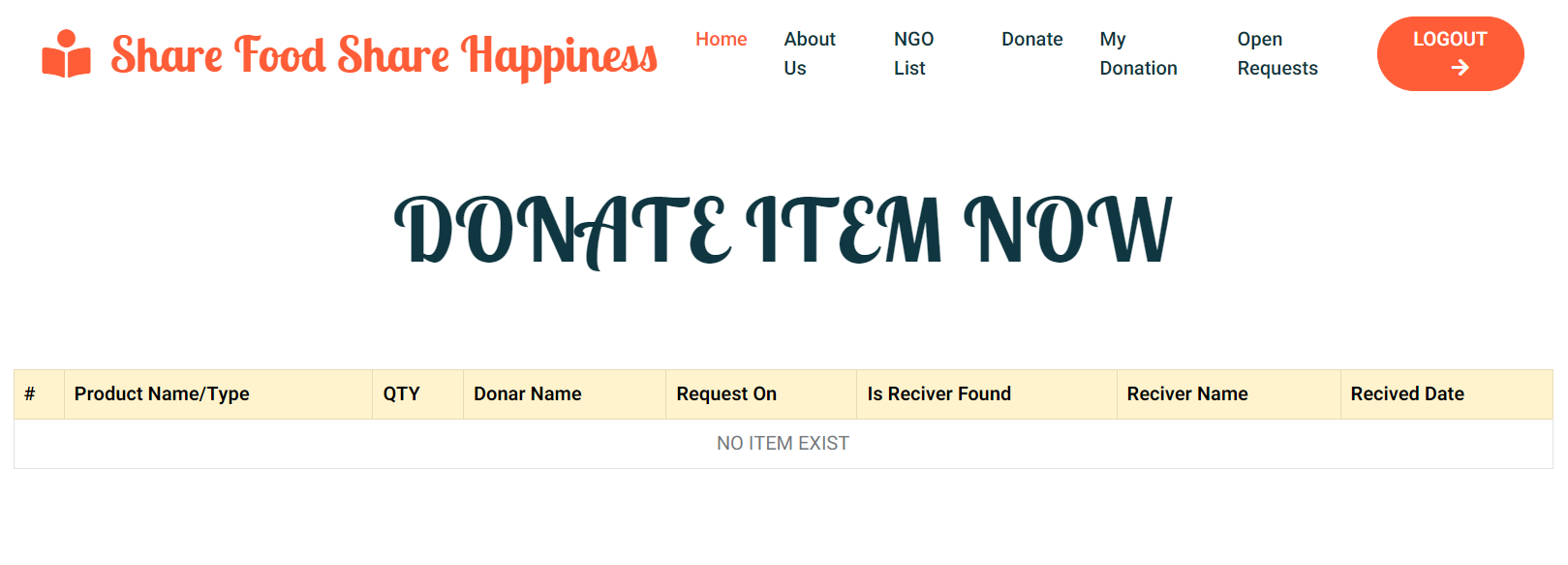
****

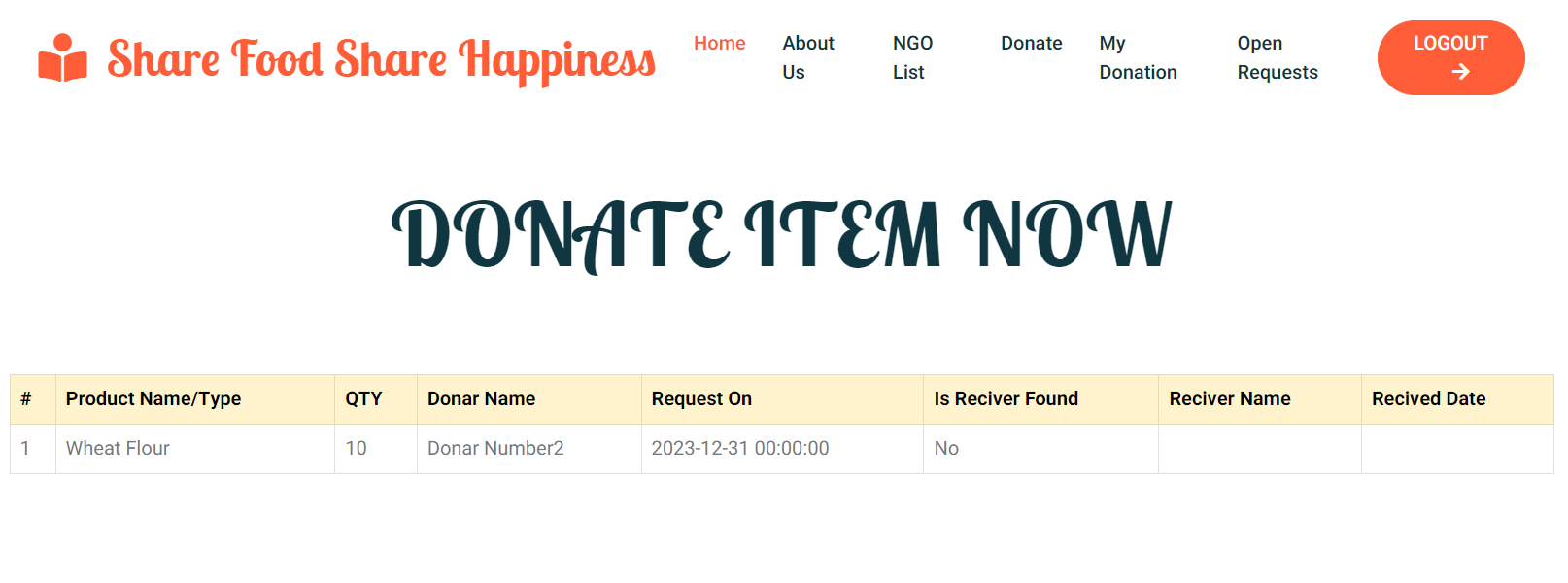
**9.14 NGOs’ List**

****

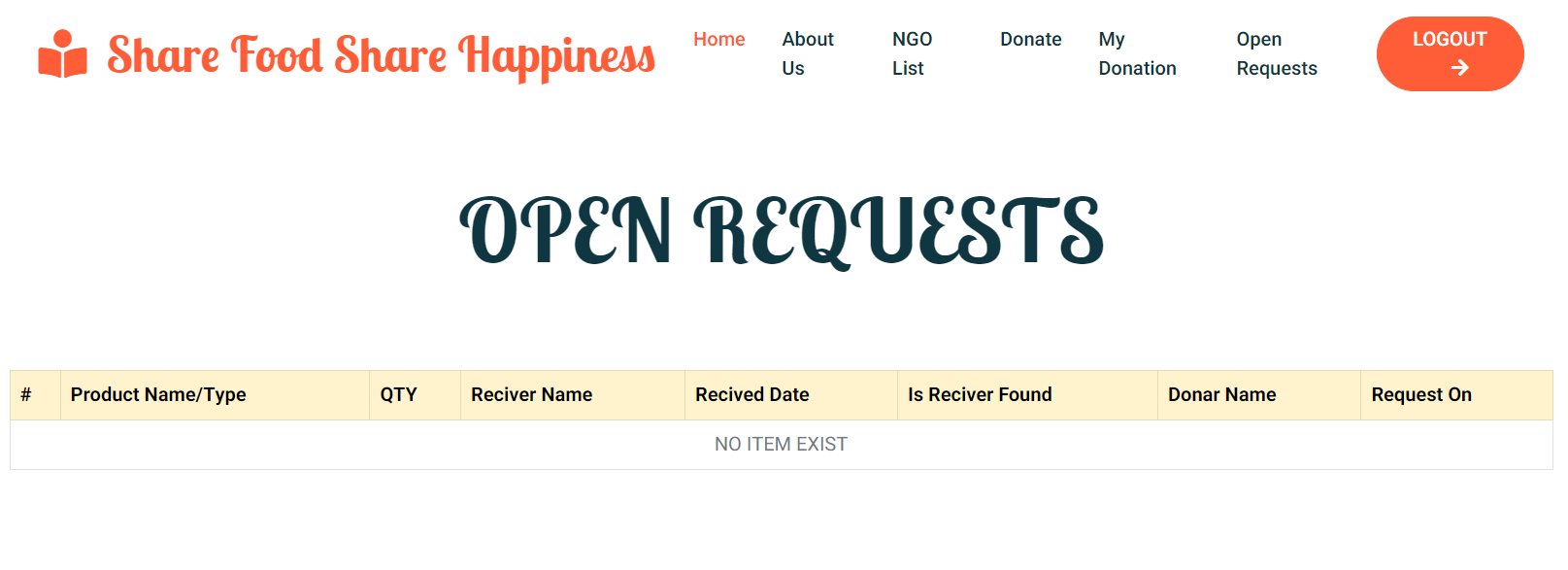
**9.15 Donate Items**

****

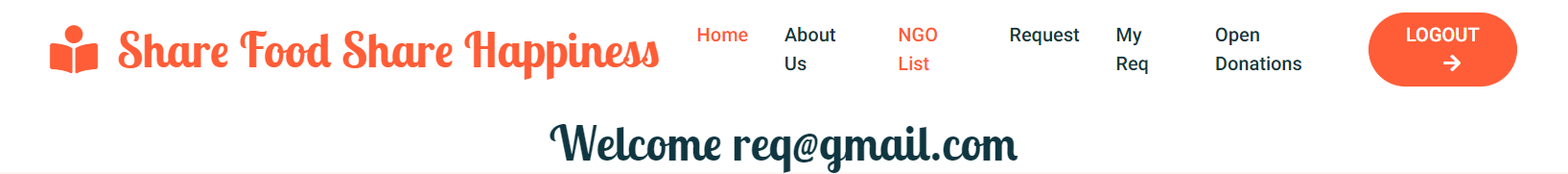
****

****

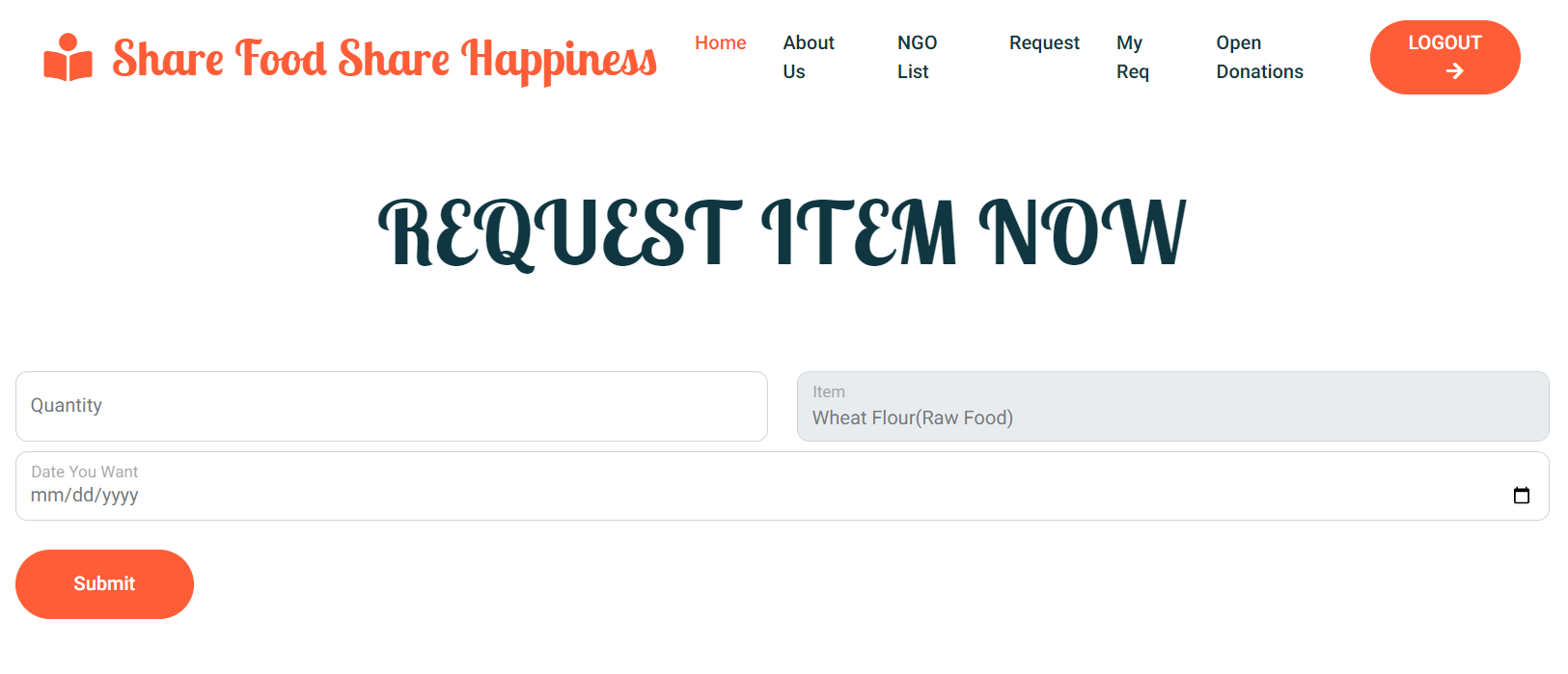
**9.16 Requests**

****

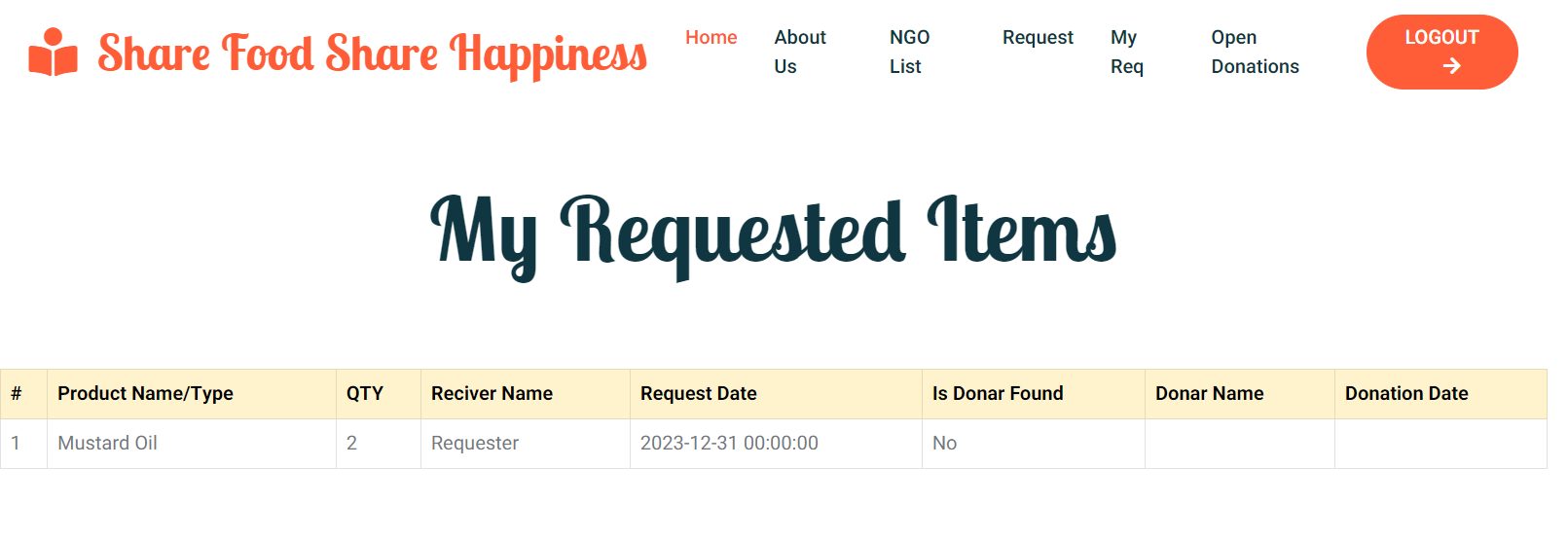
**9.17 Receiver Panel**

****

**9.18 Request Item**

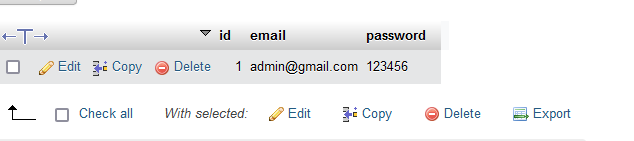
****

**9.19 Requested Items**

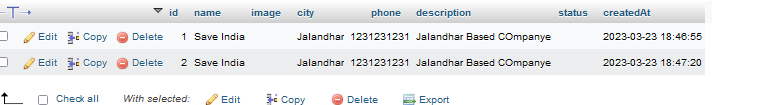
****

**9.20 Database Tables**

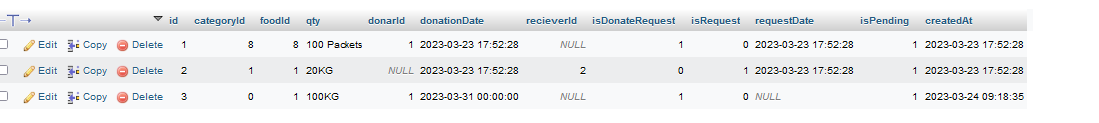
**9.20.1 Admin Table**



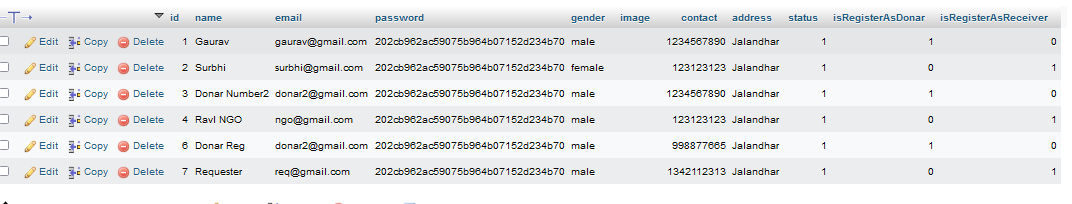
**9.20.2 NGO table**

****

**9.20.3 Request Table**

****

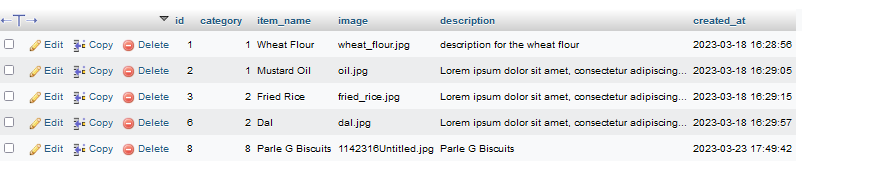
**9.20.4 Receiver Table**

****

**9.20.5 Category Table**

****

**9.20.6 Product Table**

****

**CHAPTER 10**

**TESTING**

**10.1 Introduction to Testing:**

Testing is the major quality control measure employed during software development. Testing is the process of executing a program with the intent of finding an error. No piece of code is completely ready unless it has been fully tested. This stage is very important as at this stage it is verified whether the code developed meets the requirement specifications or not. Moreover, all validations are also checked in the testing stage.

Testing is a process of executing a program with the intent of finding an error. A good test case is the one that has a probability of finding an as yet undiscovered error. If testing is conducted successfully (according to the objective stated) it will uncover errors in the software. As a secondary benefit, testing demonstrates that software function appears to be working according to the specification that performance requirement appears to have been met.

Testing is the set of activities that can be planned in advance and conducted systematically. It is an integral part of program development. It is in this stage, where we check that the program that has been coded, Perform according to the requirements. The purpose of doing the test is not to demonstrate that there are no errors in the program but to detect any bugs that may still exist.

In the testing stage, the main aim is to look for errors that unknowingly have occurred. It is a common misconception that the purpose of testing is to prove that a program is working correctly. This is a dangerous myth because it can lead to insufficient testing, and programs with hidden faults. Because the actual result and expected result may differ in the field of reality and it can be hazardous for a program.

The importance of software testing and its implications with respect to software quality cannot be over emphasized. Software testing is a crucial element of software quality and represents the ultimate review of specification design and coding.

The increasing visibility of motivating forces for well-planned thorough testing. It is not unusual for software development organizations to expend 40% of total project effort on testing.

**10.2 Test Strategy**

Implemented System is tested using Basic level of Testing that are:

**1) UNIT TESTING.**

**2) INTEGRATION TESTING.**

**3) SYSTEM TESTING.**

**4) ACCEPTANCE TESTING.**

These different levels of testing attempt to detect different types of faults. The relation of the faults introduced in different phases, and the different levels of testing are shown:

**1.) UNIT TESTING**

The first level of testing is unit testing. In this different modules are tested against the specifications produced during design for the modules. Unit testing is essential for verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules

**2.) INTEGRATION TESTING**

The next level of testing is often called integration testing. In this many tested modules are combined into sub-systems, which are then tested the goal here is to see if the modules can be integrated properly, the emphasis being on testing interfaces between modules. This activity can be considered as testing the design, and hence the emphasis on testing module interactions.

**3.) SYSTEM TESTING**

The next level of testing is system testing. Here the entire software system is tested. The reference document for this process is a requirement document, and the goal is to see if the software meets its requirements. This is essentially a validation exercise. And it was found that they all are working well to meet the Owners requirements.

**4.) ACCEPTANCE TESTING**

The last level of testing is acceptance testing. Acceptance testing is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here focuses on the external behavior of the system; the internal logic of the program is not emphasized.

**10.3 Test Cases**

For testing to be successful, proper selection of test cases is essential. There are two different approaches to selecting cases - functional testing and structural testing.

⮚ Non Functional testing of the software or the module to be tested is treated as a black box, and the test cases are decided based on the specifications of the system or the module. For this reason this type of testing is also called "black box testing" the focus here is on testing the external behavior of the system.

⮚ In structural testing the test cases are decided based on the logic of the module to be tested. A common approach here is to achieve some type of coverage of the statements in the code. One common coverage criterion is statement coverage, which requires that test cases be selected so that together they execute each statement exactly once.

**Test Case 1**

|  |  |
| --- | --- |
| Test Case Identification | Login Screen |
| Expected Results | It should display the message invalid login parameters. |
| Actual Results | It displays the error message invalid login parameters. |
| Remarks | Pass |

When User accidently enters a wrong username and password combination, then error message will display invalid username or password.

**Test Case 2**

|  |  |
| --- | --- |
| Test Case Identification | New Account Screen |
| Expected Results | It should display the message for the fields which is required to fill. |
| Actual Results | It displays the error message Please enter your name, Please enter your Phone Number etc. |
| Remarks | Pass |

When any user accidentally submits the data without filling in full details, then an error message will display.

**Test Case 3**

|  |  |
| --- | --- |
| Test Case Identification | New Account Screen |
| Expected Results | It should display the message Please enter the correct email. |
| Actual Results | It displays the error message Please enter the correct email. |
| Remarks | Pass |

When a user enters the wrong email address on create new account page, then error message will display “Please enter the correct email”.

**CHAPTER 11**

**IMPLEMENTATION**

System implementation generally benefits from high levels of user involvement and management support. User participation in the design and operation of information systems has several positive results. First, if users are heavily involved in systems design, they move opportunities to mold the system according to their priorities and business requirements, and more opportunities to control the outcome. Second, they are more likely to react positively to the change process. Incorporating user knowledge and expertise leads to better solutions. The relationship between users and information systems specialists has traditionally been a problem area for information systems implementation efforts. This is referred to as the user-designer communications gap. These differences lead to divergent organizational loyalties, approaches to problem-solving, and vocabularies. Examples of these differences or concerns are below:

**User Concerns**

● Will the system deliver the information I need for my work?

● How quickly can I access the data?

● How easily can I retrieve the data?

● How much clerical support will I need to enter data into the system?

● How will the operation of the system fit into my daily business schedule?

**Designer Concerns**

● How much disk storage space will the master file consume?

● How many lines of program code will it take to perform this function?

● How can we cut down on CPU time when we run the system?

● What are the most efficient ways of storing this data?

● What database management system should we use?

**CHAPTER 12**

**MAINTENANCE**

**Introduction to Software Maintenance**

Software maintenance denotes any changes made to a software product after it has been delivered to the customer. Maintenance is inevitable for almost any kind of product. It is practically impossible to make the software completely error-free because the input domain of most software products is very large and it is not practical to test the software exhaustively with respect to each value that the input data may assume. Maintenance is also needed to enhance the features of the software to add more functionality to it and to port to new platforms etc.

**Types of Software Maintenance**

Maintenance is fixing or enhancing a system. Many different types of maintenance must be performed on the system to ensure it continues to operate as expected. These include:

● **Adaptive maintenance -** making changes to increase system functionality to meet new requirements.

● **Corrective maintenance -** making changes to repair system defects and bugs observed while the system is in use.

● **Perfective maintenance -** making changes to enhance the system and improve such things as processing performance and usability.

● **Preventive maintenance -** making changes to reduce the chance of future system failures.

**CHAPTER 13**

**Conclusion**

The main goal of our project is that people are able to manage all the processes without any paperwork, i.e. users are not bound to the file system.

The main purpose of the website is to improve interaction.

● It will help to save their time

● With the help of these web pages users can also get knowledge of any course and perform the quiz to check their performance.

● Easy to use

**BIBLIOGRAPHY**

Websites:

● https://www.google.com

● https://www.stackoverflow.com