

A Project Report  
On  
**“FOOD SAVER WEBSITE”**  
Submitted to the  
**Department of Computer Applications**  
In partial fulfilment of the Course  
**Integrated Master of Computer Applications**  
Under the guidance of  
**Ms. Niya Roy**  
**Project Done**  
by  
**GAYATHRI DINESH**  
(Reg no: 203242110119)



**DEPARTMENT OF COMPUTER APPLICATIONS**  
**SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT**  
*December-2023*

# **SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT**



## **BONAFIDE CERTIFICATE**

Certified that the Project Work entitled

**“FOOD SAVER WEBISTE”**

**is a bonafide work done by**

**Gayathri Dinesh**

*In partial fulfillment of the requirement for the Award of*

**INTEGRATED MASTER OF COMPUTER  
APPLICATIONS**

**Degree From**  
Mahatma Gandhi University,  
Kottayam(2020-2025)

*Head of Department*

*Project Guide*

*Submitted for the Viva-Voce Examination held on.....*

**External Examiner1**

(Name & Signature)

**External Examiner2**

(Name & Signature)

# **SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT**



## **CERTIFICATE**

This is to certify that the project entitled "**“FOOD SAVER WEBSITE”**" has been successfully carried out by **GAYATHRI DINESH** (Reg no: 203242110119) in partial fulfilment of the Course **INTEGRATED MASTER OF COMPUTER APPLICATIONS.**

Date:

**HEAD OF DEPARTMENT**

# **SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT**



## **CERTIFICATE**

This is to certify that the project entitled "**“FOOD SAVER WEBSITE”**" has been successfully carried out by **GAYATHRI DINESH** (Reg no: 203242110119) in partial fulfilment of the course **INTEGRATED MASTER OF COMPUTER APPLICATIONS** under my guidance.

Date:

Niya Roy

INTERNAL GUIDE

# **SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT**



## **DECLARATION**

I, **GAYATHRI DINESH**, hereby declare that the project work entitled "**FOOD SAVER WEBISTE**" is an authenticated work carried out by me under the guidance of **Ms. Niya Roy** for the partial fulfilment of the course **INTEGRATED MASTER OF COMPUTER APPLICATIONS**. This work has not been submitted for similar purpose anywhere else except to **SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT**, affiliated to **M.G. UNIVERSITY, KOTTAYAM**.

I understand that detection of any such copying is liable to be punished in any way the school deems fit.

**Date:**

**Place:**

**GAYATHRI DINESH**

## **ACKNOWLEDGEMENT**

An endeavor over a long period can be successful with the advice, support, and blessings of many well-wishers. To acknowledge all of them is a blissful opportunity showered upon me by the Almighty. With great pleasure and privilege, I present here with full satisfaction, the Project report on "FOOD SAVER WEBSITE." I take this opportunity to express my gratitude and sincere thanks to all who helped me complete this project successfully.

I first thank God Almighty, who showered His immense blessings on my effort. I express my gratitude and sincere thanks to the Director, **Dr. Indu Nair**, for her kind consideration and valuable guidelines throughout our project work. Special thanks to **Dr. G Sashi Kumar**, our principal, for his unwavering support and encouragement.

I sincerely express my gratitude to **Dr. Anjana S Chandran**, HOD, Department of Computer Applications, SCMS School of Technology and Management (SSTM). I extend my sincere gratitude to **Ms. Niya Roy**, my project guide, for her valuable guidance, cooperation, and constant encouragement throughout the project work.

I also thank my friends and well-wishers who have provided their wholehearted support to me in this exercise. I believe that this endeavor has prepared me for taking up new challenging opportunities in the future.

**GAYATHRI DINESH**

# Table of Contents

1. EXECUTIVE SUMMARY .....	1
2. BACKGROUND .....	3
2.1. Existing System .....	3
2.2. Definition of Problem .....	3
2.3. Proposed System.....	3
3. PROJECT OVERVIEW .....	6
3.1. Objective of the project.....	6
3.2. Stakeholders.....	7
3.3. Scope of project .....	7
3.4. Feasibility Analysis.....	7
3.4.1 Feasibility study.....	7
3.4.2. Technical Feasibility.....	8
3.4.3. Operational Feasibility.....	8
3.4.4. Schedule Feasibility.....	9
3.4.5. Economic Feasibility .....	9
4 OVERALL PROJECT PLANNING .....	10
4.1. Development Environment .....	10
4.2. Constraints .....	10
4.3. Deliverables .....	11
4.4. Assumptions and Dependencies.....	11
4.5. Risks.....	12
4.6. Process Model.....	12
4.7. Test Strategy .....	12
4.7.1. System Testing.....	12
4.7.2. Types of Testing .....	13
4.8. Testing environment and tools.....	17
5. ITERATION PLANNING.....	18
5.1. Schedule.....	18
5.2. Risk .....	18
6. HIGH LEVEL SYSTEM ANALYSIS .....	19
6.1. User Characteristics .....	19
6.2. Summary of system features/Functional requirements .....	19
6.3. Non-Functional Requirements/Supplementary Specifications .....	21

6.4. Glossary .....	21
6.5. Business Rules .....	22
6.6. Use Case.....	22
6.7. Use-case Diagram .....	24
7. USE CASE MODEL.....	25
7.1. Use case Text.....	25
7.2. System Sequence Diagram.....	30
7.3. Operation Contracts .....	33
7.4. Reports .....	33
8. DESIGN MODEL.....	34
8.1. Class Diagram.....	34
8.2. UI Design .....	35
8.3. Theoretical Background.....	36
8.4. ER Diagram .....	37
8.5. Database Design.....	38
9. TESTING.....	42
9.1.Test cases .....	42
9.2. Test Report .....	48
9.3 Sample Code used for testing .....	52
10. TRANSITION.....	60
10.1 System Implementation.....	60
10.2 System Maintenance .....	61
10.2.1 Corrective maintenance .....	61
10.2.2 Adaptive maintenance.....	61
10.2.3 Preventive maintenance .....	62
11. ANNEXURE.....	63
11.1 References.....	63
11.2 Annexure I: User Interview Questionnaires.....	63
11.3 Conclusion .....	64
11.4 Sample Code .....	65
11.4.1 Screenshots .....	65
11.4.2 Sample Code.....	85

## 1. EXECUTIVE SUMMARY

"ShareSaveSustain" stands as an innovative online platform at the intersection of philanthropy and sustainability. This system empowers charity organizations to articulate specific food donation needs, allowing users to seamlessly contribute and support these causes. Simultaneously, food vendors can list items near expiration at discounted prices, fostering a culture of sustainability by minimizing food waste and providing users with cost-effective options. The platform welcomes guest users to explore discounted offerings and make direct monetary contributions to charitable organizations. With user-friendly features, including intuitive search functionality, "ShareSaveSustain" endeavors to create an engaged community where users, charities, and vendors collaborate to drive positive change in both social responsibility and sustainable practices.

"ShareSaveSustain" is more than just a platform; it's a community-driven initiative designed to reduce food waste while making a meaningful impact on societal well-being. By promoting collaboration between users, charities, and food vendors, the platform not only facilitates the efficient fulfillment of donation requests but also introduces an economic incentive for users through discounted food purchases. This dual-purpose approach aligns with the overarching goal of creating a sustainable and socially responsible ecosystem, making "ShareSaveSustain" a unique and impactful solution at the crossroads of charity, sustainability, and community engagement.

### Users of the System

- CHARITY ORGANISATIONS
- DONORS
- FOOD VENDORS
- GUEST USERS
- ADMIN

### Main Functions

#### Charity organization Features:

- Request donations
- Receive Monetary Contributions
- Track donations

- Manage donation requests
- View request history
- Mark donation status
- Edit organization profile
- Register Complaints

#### Donor Features:

- View donation requests
- Respond to request
- View donation status
- Manage donation status
- View donation history
- Edit profile
- Register Complaints

#### Vendor Features:

- List foods close to expiry
- Manage food listings
- Edit vendor details
- Register complaints

#### Guest User Features:

- Explore listings
- Make Monetary donations
- Register complaints

#### Admin Features:

- View and manage users
- View and manage Complaints.

## **2. BACKGROUND**

### **2.1. Existing System**

Presently, popular websites like FoodCloud, OLIO, and akshayapatra.org are utilized for food donations. These websites primarily focus on general charitable donations without the dynamic features of targeted requests, guest user involvement, or food vendor integration.

The proposed food saver website empowers charity organizations to dynamically request specific donations, fostering a targeted and impactful approach to addressing their immediate needs. One distinctive feature of the platform lies in its integration of food vendors, enabling them to list items close to expiry dates. This dual functionality not only promotes responsible food resource management but also aligns with the mission of reducing food waste.

### **2.2. Definition of Problem**

1. Lack of coordination and transparency between donors and organizations.
2. Food wastage is not addressed in existing systems.
3. Limited accessibility and engagement for potential users.

### **2.3. Proposed System**

"ShareSaveSustain" is a comprehensive online platform designed to revolutionize the way we approach food consumption, focusing on both charitable food donations and sustainable food waste management. This innovative system provides a dual-purpose solution by allowing users to participate in food donation initiatives for charitable causes while also engaging in responsible food waste reduction practices. The platform seamlessly connects food donors, including individuals and businesses, with charities and community groups seeking donations. Simultaneously, it facilitates food vendors in listing soon-to-expire items at discounted prices, encouraging responsible consumption and minimizing food waste.

With an emphasis on user-friendly functionalities, "ShareSaveSustain" fosters a sense of community by bringing together various stakeholders, including donors, charities, and food vendors. The system aims to address key challenges in the food industry, such as food scarcity and environmental impact, by promoting a circular economy for food resources. Users can navigate between food donation and waste management features effortlessly, contributing to a sustainable and socially responsible ecosystem. By integrating technology, regulatory compliance, and educational resources, "ShareSaveSustain" seeks to create a dynamic and impactful platform that empowers individuals and businesses to make a positive difference in both local communities and the broader food industry.

Advantages of Proposed System:

- **User-Friendly Interface:** With a user-friendly interface, "ShareSaveSustain" enhances accessibility for a diverse user base. Intuitive navigation and clear functionalities make it easy for users to participate in both food donations and waste reduction efforts.
- **Enhanced Efficiency and Flexibility:** The platform is designed to be highly efficient and flexible, allowing organizations, food vendors and users to log in and interact seamlessly, enhancing overall user experience.
- **24/7 Accessibility:** ShareSaveSustain provides round-the-clock accessibility, allowing users, organizations, vendors to engage with the platform anytime, anywhere globally.
- **Reduced Food Waste:** By facilitating the listing of soon-to-expire food items at discounted prices, the system actively contributes to the reduction of food waste. This aligns with sustainability goals and helps minimize the environmental impact of discarded food.
- **Charitable Contributions:** The platform provides a streamlined process for users to make food donations to charities and community groups. This promotes philanthropy and supports organizations in addressing food scarcity and hunger.
- **Community Engagement:** "ShareSaveSustain" fosters a sense of community by bringing together food donors, charities, and vendors on a common platform. This engagement contributes to a shared commitment to sustainable and responsible food practices.
- **Interactive and Time-Saving:** The platform is highly interactive, saving time for both organizations and users by streamlining processes and reducing paperwork.

- **Circular Economy for Food Resources:** The platform promotes a circular economy for food resources by encouraging responsible consumption, reducing waste, and redirecting surplus food to charitable causes. This contributes to a more sustainable and efficient food system.

### **3. PROJECT OVERVIEW**

#### **3.1. Objective of the project**

The ShareSaveSustain project aims to revolutionize the landscape of charitable giving and sustainable food practices through the creation of a dynamic online platform. Our primary objective is to enhance charitable giving by empowering organizations to actively request specific donations, fostering a targeted and impactful approach. Inclusivity is a key focus, with the platform allowing guest users to contribute financially, broadening the donor base and promoting widespread philanthropy. Uniquely, ShareSaveSustain integrates food vendors into the system, enabling them to list items close to expiry dates, thus addressing both charitable and environmental concerns by reducing food waste. The project seeks to provide donors with a comprehensive and engaging experience, allowing them to visualize the impact of their contributions through graphs and access detailed donation histories. Additionally, the platform aims to create a circular economy for food resources, encouraging responsible consumption practices. User accessibility and experience are prioritized through an intuitive interface, while educational resources raise awareness about sustainable practices. By streamlining logistics, ensuring regulatory compliance, and fostering community collaboration, ShareSaveSustain aims to make a significant and lasting positive impact on food sustainability and social responsibility within the broader food industry.

#### **3.2. Stakeholders**

- Charity Organizations**

The Charity organizations represent a crucial stakeholder group within the ShareSaveSustain platform, actively engaging in dynamic donation requests to address specific needs. After registering they can post their requirements. They will be able to track their requests, update their requests etc.

- Donors (Food Donors and Guest Users)**

Users, comprising both registered users and guest contributors, play a pivotal role in the ecosystem, experiencing a comprehensive platform that allows them to track the impact of their contributions, engage in targeted giving, and actively participate in fostering a sense of shared responsibility. Guest users can view the listings and also make monetary donations.

- Food Vendors

Food vendors, another key stakeholder group, benefit from the platform's unique integration by listing soon-to-expire items. This not only supports sustainable practices but also presents economic opportunities through discounted sales.

- Admin

The admin is in charge of running the entire platform. They log in using a username and password, manage users and rectify complaints. Admins play a key role in ensuring the smooth operation of Esquare.com.

### 3.3. Scope of project

The scope of the ShareSaveSustain project is comprehensive, encompassing various facets of charitable giving, sustainable food practices, and community engagement. The platform aims to provide an inclusive and dynamic space where charity organizations can actively request specific donations, donors can engage comprehensively, and food vendors can contribute to responsible food resource management.

### 3.4. Feasibility Analysis

#### 3.4.1 Feasibility study

Every project is feasible for given unlimited resources and infinitive time.

Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. Here the resources availability and requirements are said to feasible to create the proposed system.

Developing the proposed ShareSaveSustain platform requires a comprehensive allocation of technological, human, financial, and time-related resources. Collaboration and partnerships with charities, donors, and vendors, coupled with community engagement, are crucial. Continuous monitoring, evaluation through analytics tools, and user feedback mechanisms ensure adaptability and improvement. Balancing these resources effectively is pivotal for the successful creation and sustainability of the ShareSaveSustain platform.

### 3.4.2 Technical Feasibility

Technical feasibility assesses whether the current technical resources are sufficient for the new system. If they are not available, can they be upgraded to provide the level of technology necessary for the new system? It checks whether the proposed system can be implemented in the present system without supporting the existing hardware.

The technical feasibility of implementing the ShareSaveSustain platform is promising, given the availability and advancements in web and mobile app development technologies. Leveraging established frameworks and programming languages, such as HTML, CSS, JavaScript, and backend technologies like PHP , provides a solid foundation for creating a user-friendly and dynamic interface. Regular assessments of emerging technologies will allow for future enhancements, keeping the platform technically viable and adaptable in the long run.

### 3.4.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. The resources that are required to implement or install are already available with the Breakdown Assist. The persons of this Assist need no exposure to computer but have to be trained to use this particular software. The project is optimally feasible.

The operational feasibility of the ShareSaveSustain platform is evident in its alignment with the mission to combat hunger and food waste by facilitating efficient donation processes. The platform's user-friendly design and intuitive navigation enhance accessibility for donors, charities, and beneficiaries, fostering widespread adoption. The simplicity of the donation process, coupled with clear communication channels, ensures ease of use for all stakeholders. Operational workflows, including request handling, donation tracking, and user management, are streamlined to optimize efficiency. Overall, the operational feasibility of the ShareSaveSustain platform is grounded in its ability to seamlessly integrate into existing charitable operations while providing a robust, user-centric solution to address food insecurity and waste.

#### 3.4.4 Schedule Feasibility

An evaluation of the time needed for the development of this project. The time schedule required for the development of this project is very important, since more development time effects machine time, costs and delays in the development of the other systems. So, the project should be complete within affixed schedule time as far as this is concerned.

Schedule feasibility study for the design is shown below

Problem identification	5
Requirement analysis	10
Overall design	20
Construction	22
Testing	15

#### 3.4.5 Economic Feasibility

Economic feasibility determines whether the time and money are available to develop the system. It also includes the purchase of new equipment, hardware and software. Since software product must be cost effective in the development, on maintenance and in the use. It is affordable to allocate the required resources.

The project aims to achieve sustainability by attracting donors, charities, and vendors, creating a symbiotic ecosystem. Revenue models, such as transaction fees, partnerships, or sponsorships, are explored to ensure financial sustainability. A cost-benefit analysis examines both tangible and intangible benefits, such as societal impact and reduced food waste, against the incurred costs. Continuous monitoring of financial metrics and periodic reassessments contribute to the economic feasibility of the ShareSaveSustain project, guiding it towards financial success and long-term viability.

## 4. OVERALL PROJECT PLANNING

### 4.1. Development Environment

#### **Hardware Specifications**

- Intel i3 or above
- Memory: at least 4GB
- Display: Color monitor
- Keyboard: Windows Compatible
- Mouse: Windows Compatible

#### **Software Specifications**

##### **Technology used:**

###### **i. Server side**

- Front end : PHP, HTML, CSS
- IDE : Visual Studio Code
- Back end : SQL server
- Operating System : Windows

### 4.2. Constraints

- The set of constraints that we come across this system is as follows
- User Interface is only in English i.e.no other language option is available.
- The system's functionality and accessibility may be limited by the technology available to users.
- The system's impact may be constrained by geographical factors, such as the availability of participating charity organizations and donors in specific regions.
- Admin can login with his assigned username and password.

### **4.3. Deliverables**

List of documents that shall be delivered are User Manual

- System maintenance documentation.
- Application archive with source code.
- Database backup and DDL script.
- Complete source code.

### **4.4. Assumptions and Dependencies**

#### **a) Assumptions**

- All roles are created in the system already but further registration of users on given roles can be done.
- The code should be free of compilation errors/syntax errors.
- The product must have an interface which is simple enough to understand.
- Roles and tasks are predefined and are made known to the administrator.
- End users should have basic knowledge of computer.

#### **b) Dependencies**

- All necessary hardware and software are available for implementing and use of the tool.
- All roles are created in the system already.
- The proposed system should be designed, developed and implemented based on the software requirements specifications document.

#### 4.5. Risks

Some of the risks are follows

- Database crash will cause heavy data loss
- Wrong input will cause discrepancies in data
- Availability of the network.

#### 4.6. Process Model

The process model for developing the project is agile model. The phases are: -

- Requirement analysis
- System study
- Designing
- Coding
- Testing
- Maintenances

#### 4.7. Test Strategy

##### 4.7.1. System Testing

When a system is developed, it is hoped that it performs properly. In practice however some errors always occur. The main purpose of testing and information system is to find the errors and correct them. A successful test is one which finds an error. The main objectives of system testing are:

- To ensure during operation the system will perform as per specifications.
- To make sure that the system meets the requirements during operation.
- To verify that the controls incorporated in the system function as intended.
- To see that when correct inputs are fed to the system the outputs are correct.

- To make sure that during operation incorrect input and output will be deleted.

The scope of a system test should include both manual operations and computerized. Operation system testing is a comprehensive evaluation of the programs, manual procedures, computer operations and controls. System testing is the process of checking if the developed system is working according to the original objectives and requirements. All testing needs to be conducted in accordance with the test conditions specified earlier.

#### 4.7.2 Types of testing

##### **Unit Testing**

Unit Testing will be done to test field validations, navigation, functionality of the programs and its block. These tests are applied on various functions within each program and other critical program blocks.

In the context of the ShareSaveSustain, unit testing would be crucial for ensuring the proper functioning of specific modules and features within the system. Each key element of the system, such as user authentication, donation processing, and request handling, should undergo rigorous unit testing.

For instance, unit tests could focus on the user registration and login functionalities, checking if user credentials are appropriately validated and stored in the database. Donation processing units might be tested to confirm that the system accurately records and updates donation details, including quantity, date, and associated charities. Similarly, unit tests for the request-handling module would verify that the system correctly processes and displays donation requests from charitable organizations.

##### **Module Testing**

Module Testing will be each program done to test the interaction between the various programs within one module. It checks the functionality of with relation to other programs within the same module. It then tests the overall functionality of each module.

For the ShareSaveSustain, module testing would entail examining specific features or sections of the application in isolation. For instance, a module responsible for user management (registration, login, and profile management) would undergo module testing to confirm that user-related functionalities work independently and cohesively. Similarly, the donation processing module would be tested to verify its functionality,

including capturing donation details, updating records, and triggering relevant notifications.

### **Integration Testing**

The major concerns of integration testing are developing an incremental strategy that will limit the complexity of entire actions among components as they are added to the system. Developing a component as they are added to the system, developing an implementation and integration schedules that will make the modules available when needed, and designing test cases that will demonstrate the viability of the evolving system. Though each program works individually they should work after linking them together. This is also referred to as interfacing. Data may be lost across interface and one module can have

adverse effect on another. Subroutines after linking may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing program structure while at the same time conducting tests to uncover errors associated with the interface. In the testing, the programs are constructed and tested in small segments.

Integration testing in the context of the ShareSaveSustain involves evaluating the combined functionality of multiple modules or components to ensure they interact correctly and produce the expected outcomes when integrated. This phase is crucial for identifying potential issues that may arise from the collaboration of different system elements.

For ShareSaveSustain, integration testing would focus on verifying that various modules, such as user management, donation processing, and request handling, work harmoniously when interconnected. The goal is to detect any interface errors, data flow issues, or communication problems between these modules.

### **Validation Testing**

This provides the final assurance that the software meets all the functional, behavioral and performance requirements. The software is completely assembled as a package. Validation succeeds when the software functions in a manner in which user wishes. Validation refers to the process of using software in live environment in order to find errors. During the course of validation, the system failure may occur and sometime the coding has to be changed according to the requirement. Thus, the feedback from the validation phase generally produces changes in the software. Once the application was

made of all logical and interface errors, inputting dummy data ensure that the software developed satisfied all the requirements of the user. The dummy data is known as test cases.

Validation testing in the context of the ShareSaveSustain involves the assessment of the entire system to ensure it meets the specified requirements and functions as intended in the real-world environment. This phase of testing focuses on validating whether the system delivers the expected outputs, provides accurate results, and meets the needs of its users and stakeholders.

For the ShareSaveSustain, validation testing would encompass verifying that features such as donation processing, user registration, and request handling align with the project's goals and user expectations. This testing phase aims to confirm that the system meets the defined acceptance criteria and operates effectively within the targeted environment.

### **Output Testing**

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specific format. Asking the users about the format of output they required, tests the output generated in two ways. One is on screen and another is printed format. The output format on the screen found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as the specified requirement by the user. Hence output testing does not result in any correction in the system.

Key aspects of output testing for the ShareSaveSustain include:

- Data Accuracy: Confirming that the data displayed in the system's outputs, such as donation status, request information, and user profiles, is accurate and up-to-date. This ensures that users can rely on the information presented by the system.
- Calculation Accuracy: Verifying the correctness of any calculations performed by the system, especially those related to financial transactions, quantities, and dates. This helps prevent errors that could impact the integrity of the system.
- Report Generation: Ensuring that any reports or summaries generated by the ShareSaveSystem present relevant and accurate information. This is crucial for

decision-making and monitoring system activities.

### **Acceptance Testing**

Acceptance testing (also known as user acceptance testing) is a type of testing carried out in order to verify if the product is developed as per the standards and specified criteria and meets all the requirements specified by customer. This type of testing is generally carried out by a user/customer where the product is developed externally by another party. Acceptance testing falls under black box testing methodology where the user is not very much interested in internal working/coding of the system, but evaluates the overall functioning of the system and compares it with the requirements specified by them. User acceptance testing is considered to be one of the most important testing by user before the system is finally delivered or handled over to the end user. Acceptance testing is also known as validation testing, final testing, QA testing, factory acceptance testing and application testing etc. And in software engineering, acceptance testing may be carried out at two different levels; one at the system provider level and another at the end user level (hence called user acceptance testing, field acceptance testing or end-user testing). Acceptance test refers to the acceptance of data into the system for processing. The acceptance test contributes to the consistency and smooth working of the system. The system under consideration is tested for users at a time for developing and making changes whenever required.

In the context of the ShareSaveSustain, acceptance testing involves the following key aspects:

- User Acceptance Testing (UAT): In UAT, representatives from the end-users, charities, and other stakeholders interact with the system to validate its functionality and ensure it fulfills their needs. Test scenarios are executed to confirm that the system behaves as expected in real-world usage.
- Functional Acceptance Testing: This phase assesses whether all the functional requirements specified for the ShareSaveSustain have been implemented correctly. It includes testing features related to user registration, donation processing, request handling, and other core functionalities.
- Performance Acceptance Testing: Verifying that the ShareSaveSustain performs adequately under normal and peak load conditions. This testing ensures that the system can handle the expected number of users and transactions without significant degradation in performance.

- Usability Testing: Evaluating the user interface and overall user experience to verify that the system is intuitive, easy to navigate, and meets the expectations of its target audience.

#### 4.8. Testing environment and tools

The hardware specification used for testing:

Operating system	Windows 11
Memory	8 GB
Hard Disk	512 GB

The software specification used for testing:

Front End	PHP, HTML, CSS, Bootstrap
Back End	WAMPP
Operating System	Windows 11

## 5. ITERATION PLANNING

### 5.1. Schedule

SERIAL NO.	TASK	DURATION
1	Problem identification	5 days
2	Requirement Specification	10 days
3	Database Design and Analysis	12 days
4	Design Analysis	9 days
5	Coding	24 days
6	Testing	10 days
	Total	70 days

### 5.2. Risk

- Wrong input
- Software installation issues.
- Database crash will cause heavy data loss

## 6. HIGH LEVEL SYSTEM ANALYSIS

### 6.1. User Characteristics

All users of the system are expected to have basic knowledge of using a computer and basic knowledge in English language.

Users of the system:

- Charity organizations
- Donors
- Guest Users
- Food Vendors
- Admin

### 6.2. Summary of system features/Functional requirements

#### Account Registration

Donors, charity organizations and food vendors can register and create accounts

#### Request donations

Charity organizations can request donations

#### Manage requests

Charity organization can edit or delete existing requests.

#### Respond to request status

Charity organizations can mark whether a donor has donated or not in response to a specific donation request.

#### View request history

Charity organizations can view their request history.

#### View monetary donations

Charity organizations can view details of monetary donations made by guest users.

### [View requests](#)

Donors can view details of requests made by charity organizations.

### [Make donations](#)

Users can express interest to donate to specific requests.

### [Manage donations](#)

Users can delete existing donations.

### [View donation history](#)

Users can view donation history.

### [Monetary donations](#)

Guest users can make monetary donations to specific charity organization.

### [View food listings](#)

Guest users can view details of discounted food products close to expiry date.

### [List food items](#)

Food vendors can list their products close to expiry date.

### [Manage listings](#)

Food vendors can edit or delete existing items.

### [Profile Management](#)

Donors, food vendors and charity organizations can view, edit, and manage their profiles.

### [Register Complaints](#)

Donors, charity organizations, vendors and guest users can register complaints.

### [Manage Complaints](#)

Admin can rectify and change the status of complaints.

### [Manage Users](#)

Admin can view and manage users.

### 6.3. Non-Functional Requirements/Supplementary Specifications

The non-functional requirements which define the system performance are:

#### **Accuracy:**

The level of accuracy in the proposed system will be high. All operations would be done correctly and it ensures that whatever information that comes from the center is accurate.

#### **Reliability:**

The reliability of the proposed system will be high. The reason for the increased reliability of the system is that system uses correct formulas to calculate the results.

#### **Immediate Response:**

The system is highly responsive because it uses well accurate formulas to calculate required results provided the user should enter the valid input data.

#### **Easy to Operate:**

The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

The other non-functional requirements are:

- Security
- Maintainability
- Extensibility
- Reusability
- Resource utilizations

### 6.4. Glossary

Users	Charity organizations, Donors, Food vendors
Guest Users	Monetary donors
Admin	Administrator

## 6.5. Business Rules

The information should be correct and valid.

## 6.6. Use Case

### Register

Users can register their accounts.

### Login

Users can access their accounts using their username and password.

### Make requests

Charity organizations can request for donations.

### Manage Requests

Charity organizations can edit, delete their requests

### View request status

Charity organization can view the list of donors ready for donating and they also have the option to mark whether they have donated or not.

### View monetary donations

Charity organizations can see the monetary donations made by the guest users.

### View requests

Donors can see the list of requests made by the charity organizations.

### Make donations

Donors can express their interest in donation and can also specify the quantity.

### Manage donations

Donors can edit or delete the current interest to donate

### List food items

Vendors can list the food items that are close to expiry date

### Manage listing

Vendors can delete or edit the items in the listing.

### [View food listings](#)

Guest users can view the list the food items close to expiry date and can also search according to the location

### [Make monetary donations](#)

Guest users can make monetary donations to the charity organizations

### [Edit profile](#)

Users can edit the details.

### [Register Complaints](#)

Users can register complaints.

### [View Users](#)

Admin can view all the users

### [Manage Users](#)

Admin can block users.

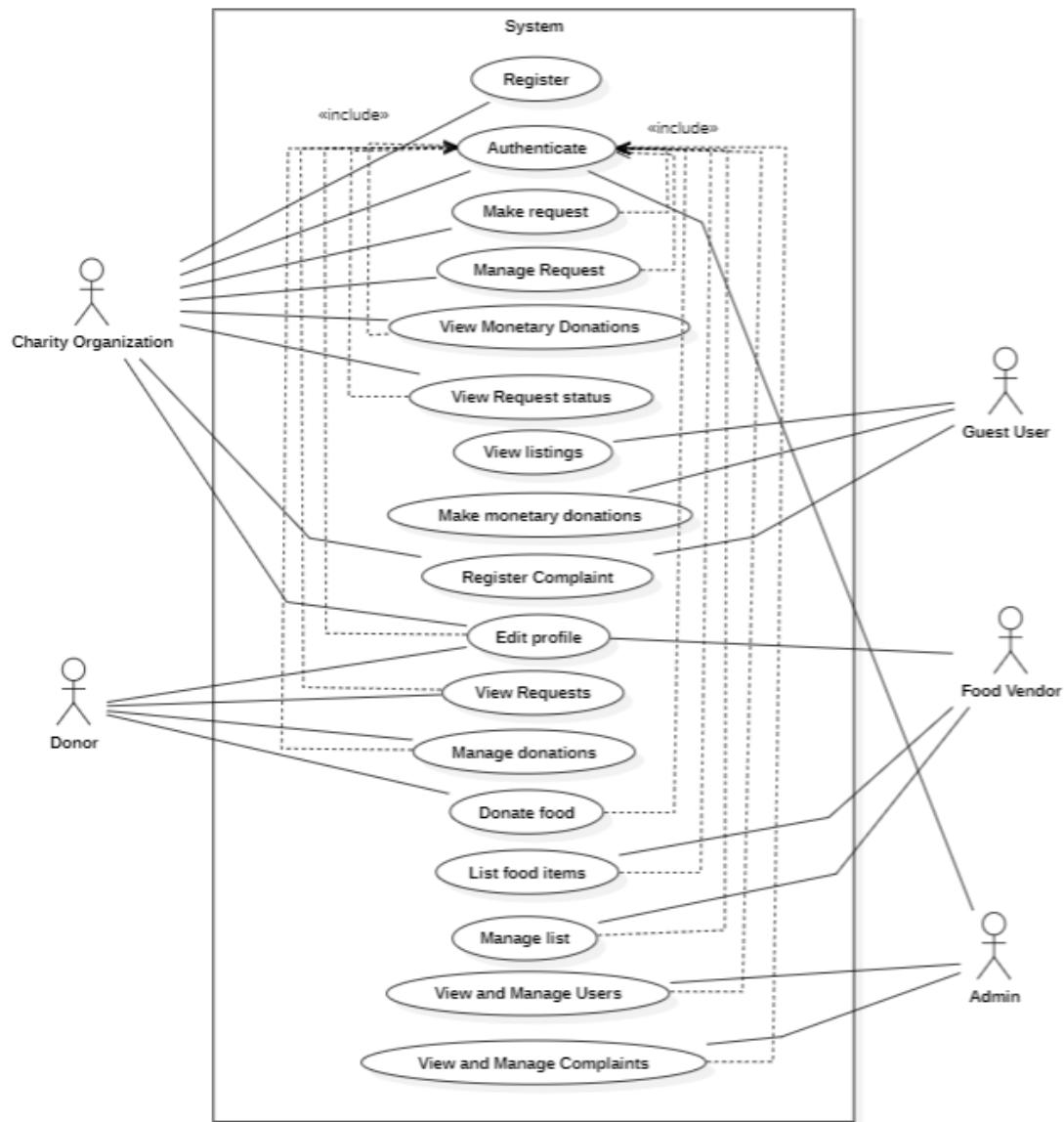
### [View Complaints](#)

Admin can view the complaints from the users.

### [Manage Complaints](#)

Admin can rectify and update the status of the complaints.

## 6.7. Use-Case Diagram



## 7. USE CASE MODEL

### 7.1. Use case text

**Scope:** Food donation management

**Primary Actors:** Charity organizations, Donors, Guest users, Vendors, Admin

#### **Stakeholders and Interests:**

- **Charity organization:** Charity organizations are the ones that make food requests. They can list the details of the request including the quantity needed. Once they made the request, they have an option to either delete or edit the request. If any donor made an interest to donate, they can view it in the request status page. Organization has an option to check whether the donor has donated or not. If the donor has not donated then the quantity is updated automatically. After 12 hours the status of the request becomes inactive. Organization can also view the details of donors who have made monetary donations. They can also view reports.

- **Donors:** Donors can view the list of the requests and can also express an interest to donate by specifying the quantity. They can also edit or delete their current interest. Donors can also generate reports.

- **Food Vendors:** Food vendors can list the food items that are close to expiry date. They can also delete or edit the current items. Items are automatically deleted after the expiry date.

- **Guest users:** They can view the list of food items that are close to expiry date. Users can search based on the location. They can also make monetary donations to charity organizations. A receipt is generated after the donations. They need not make an account to donate.

- **Admin:** Admin can view the list of users registered in the system. Admin can block or unblock users. Admin can also see the complaints and rectify them.

#### **Preconditions:**

Only guest users can use the system without logging in, no other users can use the system without logging in to the system.

#### **Success Guarantee (Post conditions):**

Really easy to use the system.

## **Main Success Scenario:**

### Use Case: Users - Register

1. System redirects to register where users can register.
2. The user enters all the necessary details in the form.
3. User clicks on the ‘Register’ button.
  - a. The system validates the form and if there is no issue the information is stored in the database

### Use Case: Users – Authenticate

1. System redirects to login page
2. Users enters the username and password in the corresponding form.
3. User clicks on Log in button
  - a. The system authenticates and if the username and password is correct the system redirects to respective homepage

### Use Case: Charity Organization- View and manage requests

1. System redirects to organization’s home page.
2. The organization views all the current requests and their details.
3. Organization clicks on ‘Add new request’ button for requesting.
  - a. A popup window appears where organization can provide the details about the request and submit the request
  - b. The organization can also edit or delete from the same page
    - i. System validates the property information and if there’s no issue, the data is updated or deleted from the database.
4. Organization clicks on the ‘Edit’ button.
  - a. A popup window appears where the organization can edit the details
5. Organization clicks on the ‘Delete’ button.
  - a. System prompts for confirmation and validates the information and if there’s no issue, the data is deleted from database.

### Use Case: Charity Organization - Manage Requests status

1. System redirects to organization’s page where they can manage status.
2. The organization views all the donors who are ready to donate and their details.

3. Organization clicks on the ‘Donated’ or ‘Not Donated’ button to update their status.

3.1. The system updates the quantity of request if the organization clicks of ‘Not donated’ button.

#### Use Case: Charity Organization – Edit profile

1. System redirects to organization’s page where they can edit details.
2. The organization views the current details.
3. Organization clicks on the ‘Edit profile’ button.
  - a. System prompts validates the information and if there’s no issue, the data is updated in the database.

#### Use Case: Donor – Make donations

1. System redirects to donors’ page where they can view details of request.
2. The donor clicks on ‘View details’ button of a request.
3. A popup window appears and the donor clicks on ‘Donate’ button
  - a. System validates the information and if there’s no issue, the data is entered into the database.

#### Use Case: Donor- Manage donations

1. System redirects to donors’ page where they can manage current donations.
2. The donor clicks on ‘Delete’ button.
  - a. System validates the information and if there’s no issue, the data is deleted from database

#### Use Case: Donor – Edit profile

1. System redirects to donor’s page where they can edit details.
2. The donor views the current details.
3. Donor clicks on the ‘Edit profile’ button.
  - a. System prompts validates the information and if there’s no issue, the data is updated in the database.

#### Use Case: Vendor – View and manage food listings

1. System redirects to vendors’ home page.
2. The organization views all the current items and their details.
3. Vendor clicks on ‘Add new request’ button for adding.
  - a. A popup window appears where vendor can provide the details about the item and submit the form.

- b. The vendor can also edit or delete from the same page
  - i. System validates the property information and if there's no issue, the data is updated or deleted from the database.
- 4. Vendor clicks on the 'Edit' button.
  - a. A popup window appears where the vendor can edit the details.
- 5. Vendor clicks on the 'Delete' button.
  - a. System prompts for confirmation and validates the information and if there's no issue, the data is deleted from database

#### Use Case: Vendor – Edit profile

1. System redirects to vendor's page where they can edit details.
2. The vendor views the current details.
3. Vendor clicks on the 'Edit profile' button.
  - a. System prompts validates the information and if there's no issue, the data is updated in the database.

#### Use Case: Guest User – Make monetary donations

1. The user clicks on the 'Donate' page and the system redirects to a new page where they can fill the form and select an amount.
2. User clicks on the 'Donate' button.
  - a. System validates the information and if there's no issue, the data is inserted in to the table and the information is sent to the corresponding charity organization.

#### Use Case: Admin – View and manage users

1. System redirects to admin's page where admin can view details.
2. The admin views the current status.
3. Admin clicks on the 'Block' button.
  - a. System prompts for confirmation and blocks the user

#### Use Case: Admin – View and manage complaints

1. System redirects to admin's page where admin can view complaints.
2. The admin views the current status of complaints.
3. Admin clicks on the 'Rectified' button.
  - a. System validates the information and updates in the database.

**Extensions:**User - Register

- 2.a. The new password and password retyped for confirmation isn't same.
  1. System tells the user that the passwords entered doesn't match.
  2. The use case continues at step 2.
- 2.b. The form doesn't contain all the necessary details.
  1. System tells the user that some information is missing.
  2. The use case continues at step 2.
- 2.c. The entered user information is in incorrect format.
  1. System tells the user that some information is incorrect.
  2. The use case continues at step 2.

Charity Organization- Add and manage Request

2. a. The entered information doesn't contain all the necessary details.
  1. System tells the User that some information is missing.
  2. The use case continues at step 2.
- 4a. The entered information doesn't contain all the necessary details.
  1. System tells the charity that some information is missing.
  2. The use case continues at step 4.

Donor- Make donation

- 3.a. The entered property information is in incorrect format.
  1. System tells the donor that some information is incorrect.
  2. The use case continues at step 3.

**Special Requirements:**

1. Text must be visible from 1 meter.
2. We want robust recovery when the system fails.
3. Language internationalization on the text displayed.

**Frequency of Occurrence:**

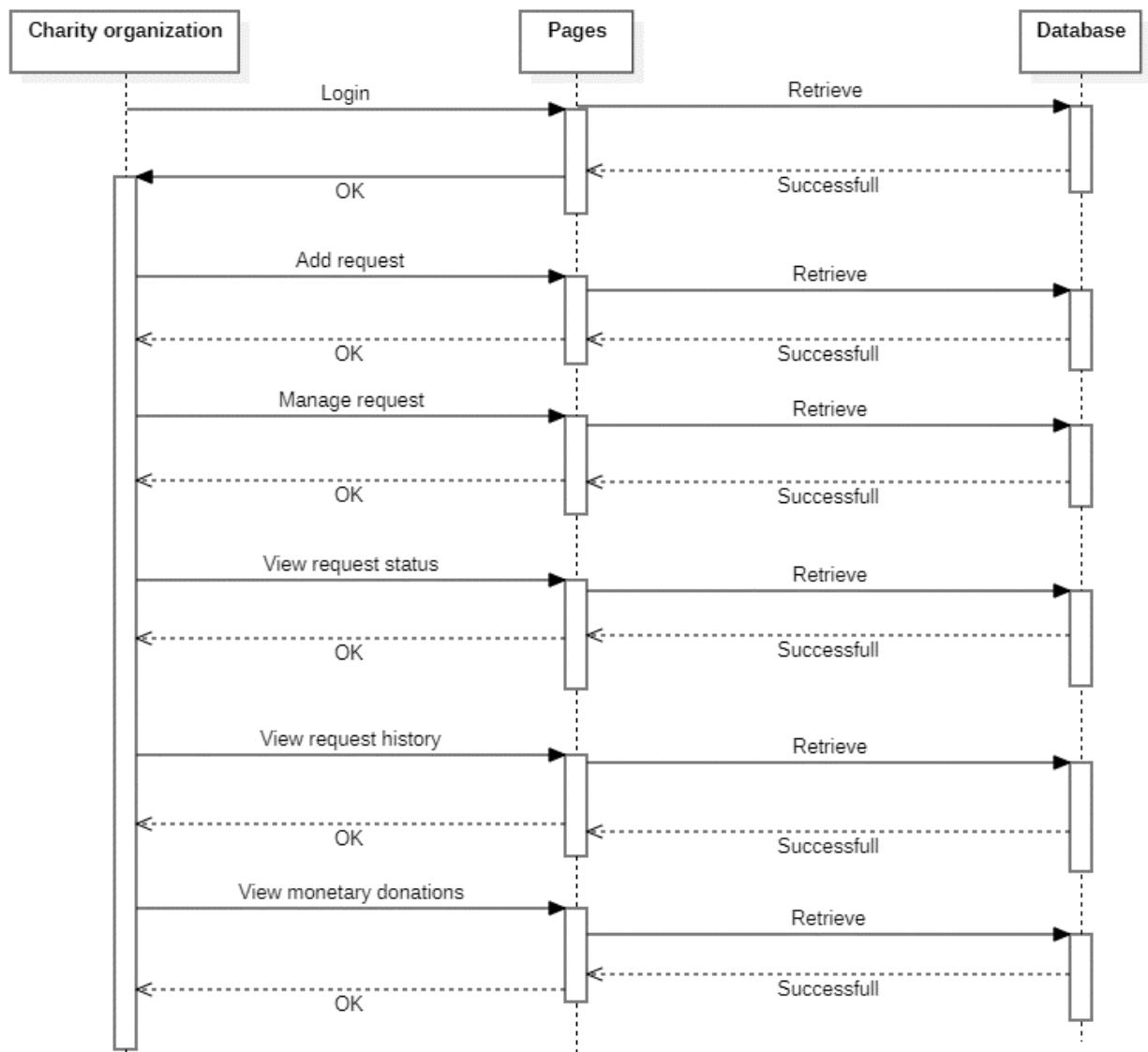
Could be nearly continuous.

**Open Issues:**

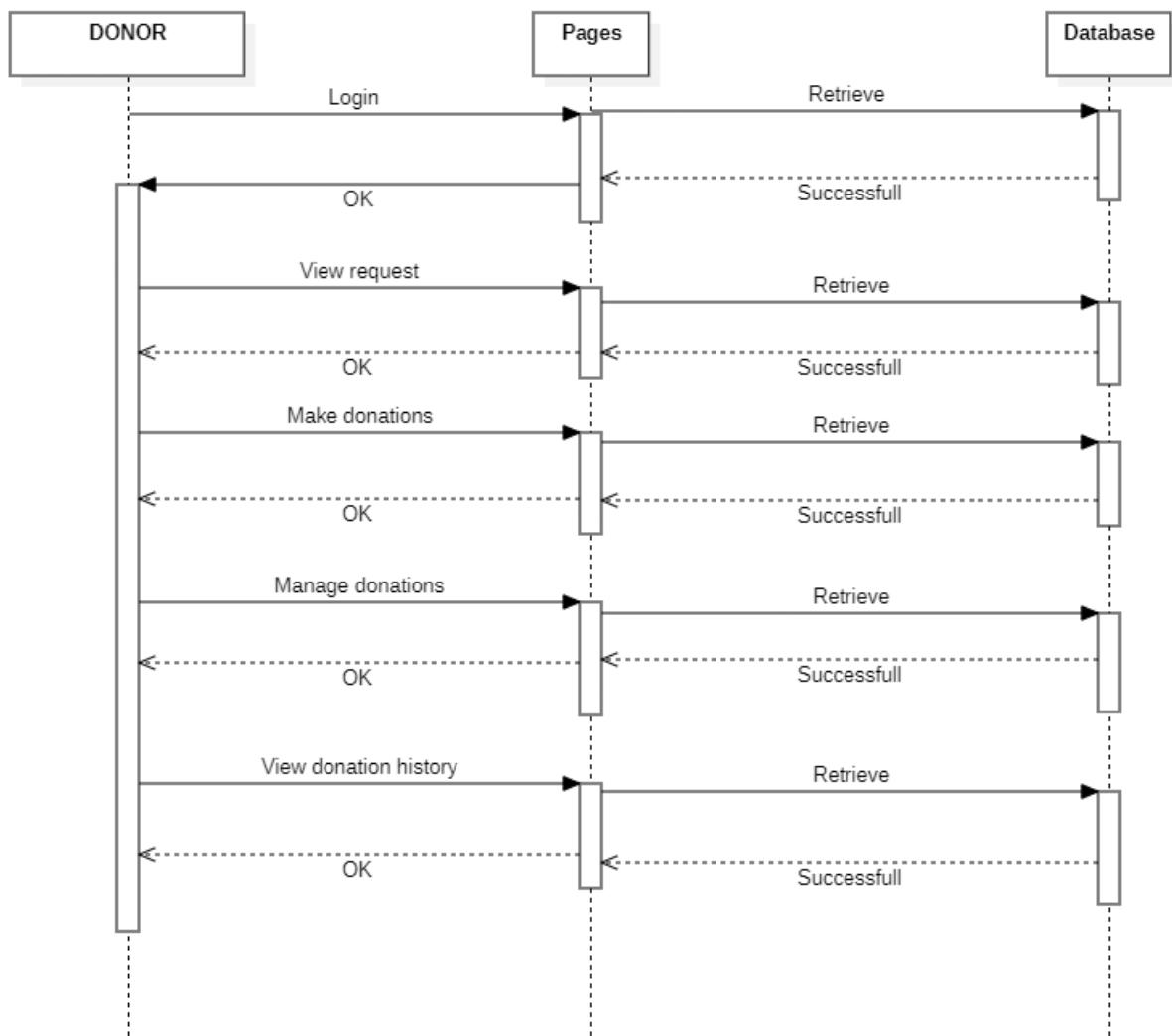
1. Explore the recovery issues

## 7.2.System Sequence Diagram

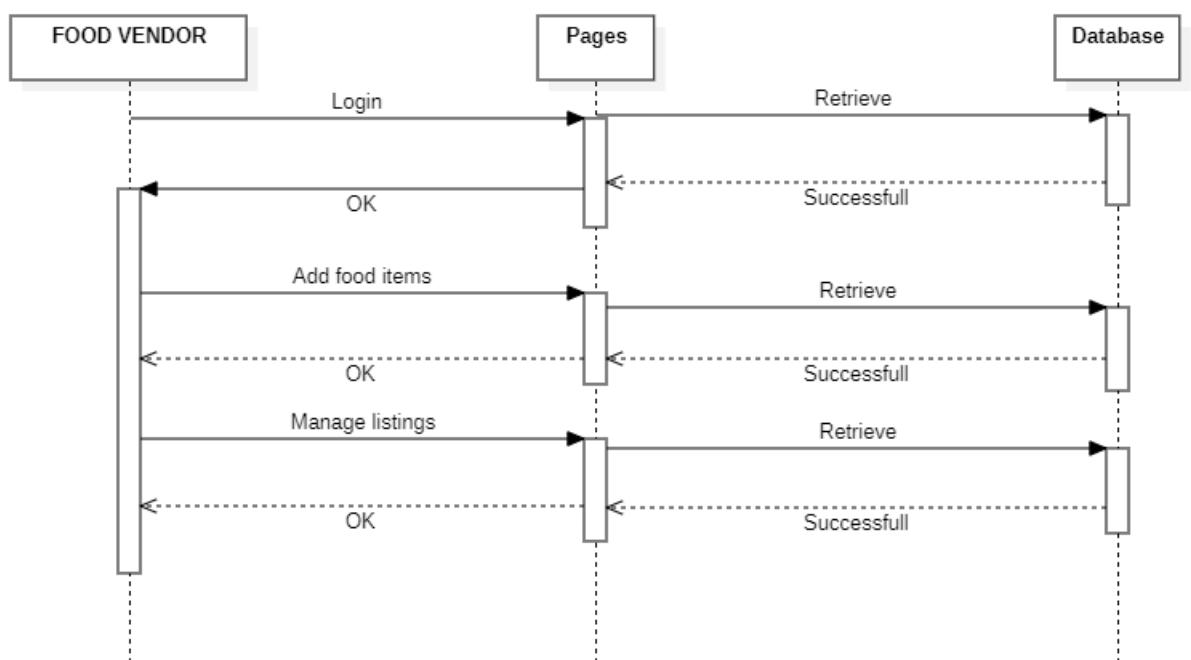
### Charity Organization Side



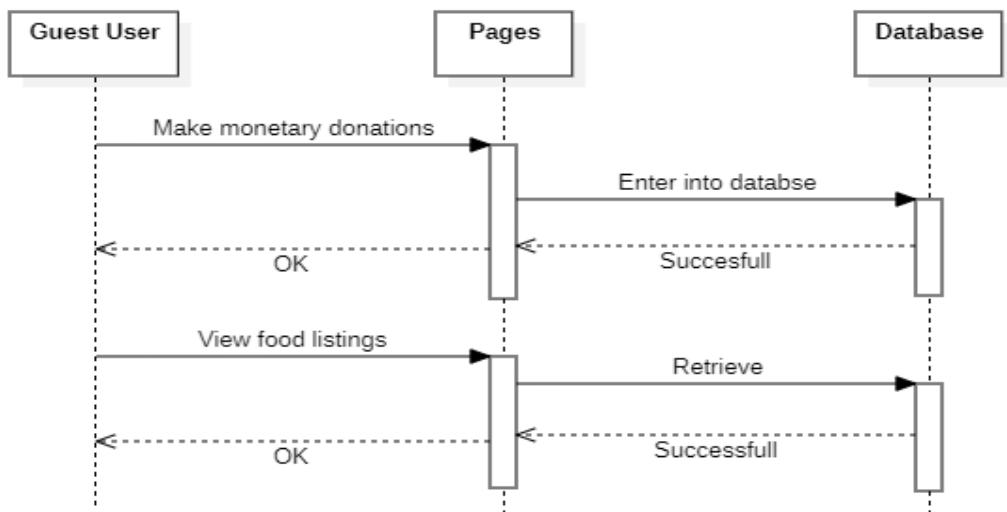
## Donor Side



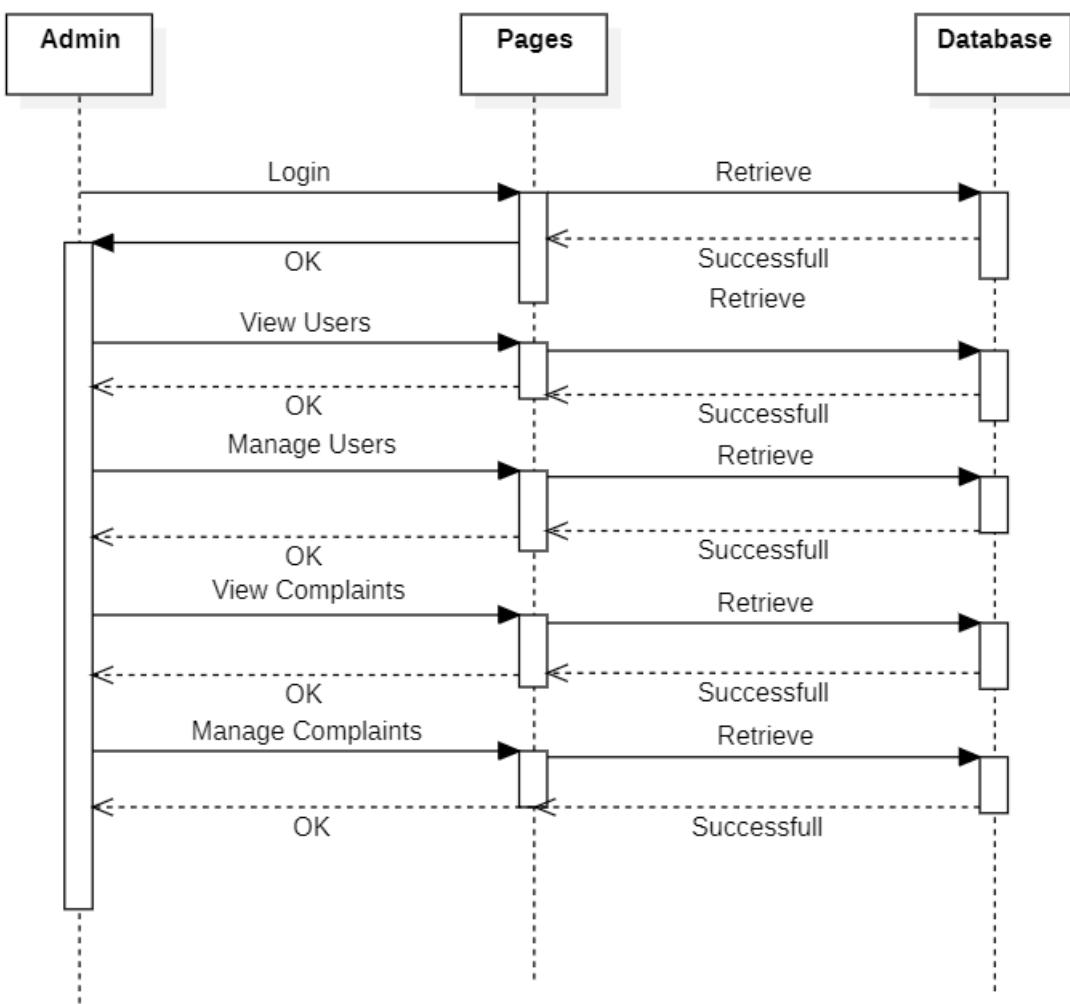
## Vendor Side



### Guest User side



### Admin Side



### 7.3. Operation Contracts

Operation: Registered User (username: string, password: string)

**Cross Reference:** Use case: User login

**Preconditions:** Proper communication between pages

**Conditions:**

- A new login instance was created was created by the user.
- Accepted username and password and stored it in respective attributes.
- After validation, the user gets logged in to the system.

### 7.4. Reports

Donor Reports

**Donation history Report:** Donor can view the donations made, including donation quantity, date etc.

**Total donations Report:** Donor can view the total quantity donated and also view graph- day wise and month wise.

**Total charity organizations supported Report:** Donors can view total number of charity organizations they have supported.

**Highest donation Report:** Donor can view the highest donated quantity.

Charity Organization Reports

**Request history Report:** Charity organization can view the details of the previous requests.

**Monetary donations Report:** Charity organizations can view the details of donors who have made monetary donations.

**Total requests made Report:** Charity organizations can view the total number of requests made.

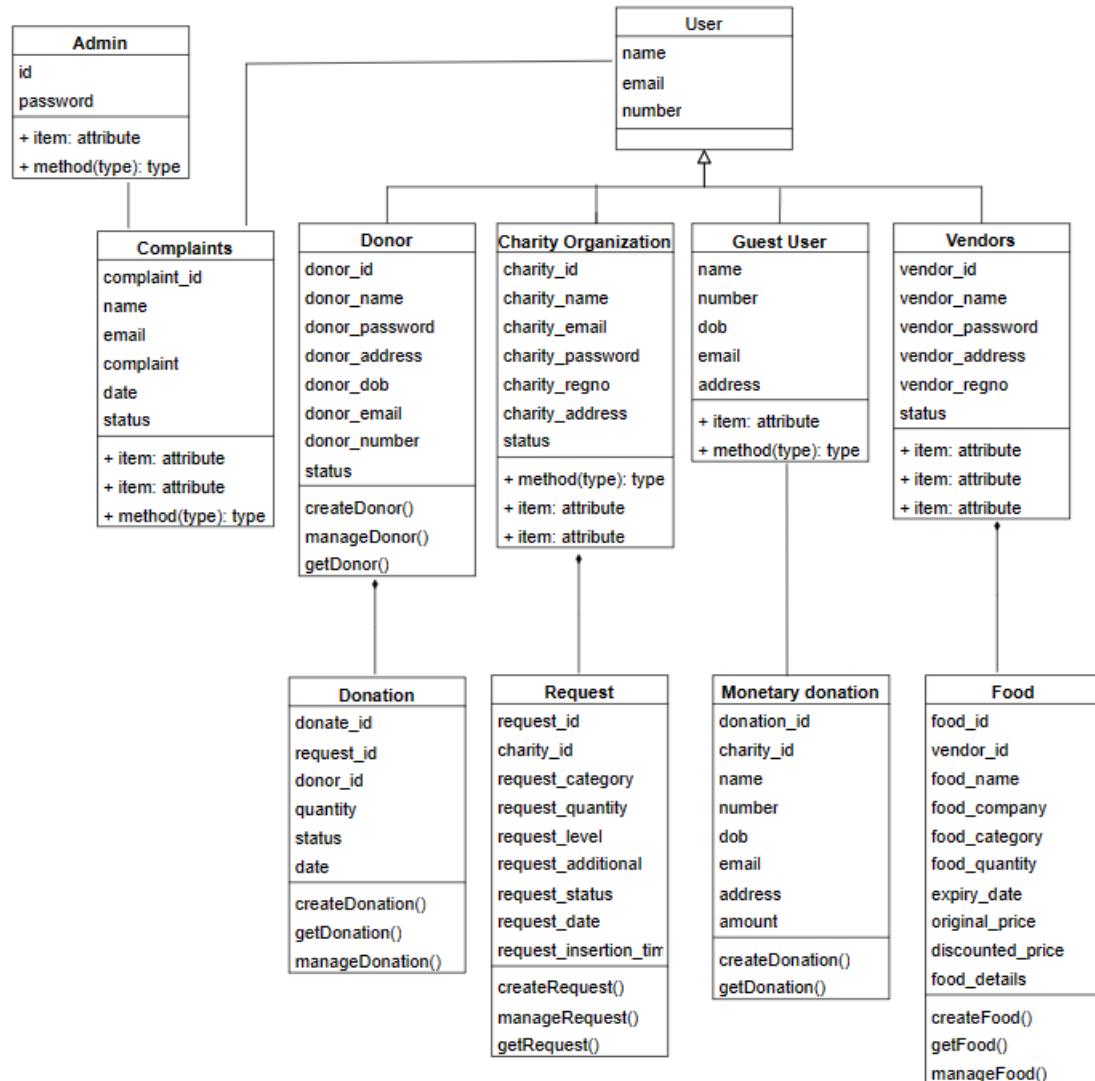
**Total amount received Report:** Charity organizations can view total amount of money received.

Admin Reports

**Total users registered Report:** Admin can view total number of users registered in the system.

## 8. DESIGN MODEL

### 8.1 Class Diagram



## 8.2 UI Design

Home Page: Displays the information about the website and also have the links to registration of users

Donation Page: Guest Users can view the page and make monetary donations through this page.

Find Soon to Expire Foods Page: Guest users can view the details of food items close to expiry date and also search by location

Login Page: Users can login to their account using their username and password.

About Us Page: Guest users can register complaints in this page.

### CHARITY ORGANIZATION

Home page: Displays an overview of total requests made, total numbers of donors and total amount of donations receive

Request Page: Displays the list of current requests. Organization can also add new request, edit or delete existing requests

Request Status Page: Charity organization can view the list donors who have responded to the request and also check whether they have donated or no

Request History Page: Charity organization can view the history of requests made.

Contributions page: Charity organizations can view the details of donors who have made monetary contributions

Edit Profile Page: Charity organizations can edit the details of the organization

Contact Us page: Charity organizations can register complaints

### DONOR:

Home Page: Displays an overview of total donations made, total charity organizations supported etc.

View Request Page: Displays the list of current requests. Donors can express an interest to donate food.

Donation Status Page: Donors can view the status of donations and donor can delete current interest.

Donation history Page: Donor can view the details of previous donations made.

Edit Profile Page: Donors can edit the details.

Contact Us page: Donors can register complaints

**FOOD VENDOR:**

Home Page: Vendors can view the list of current items. Vendors can also add new item and can edit or delete current items.

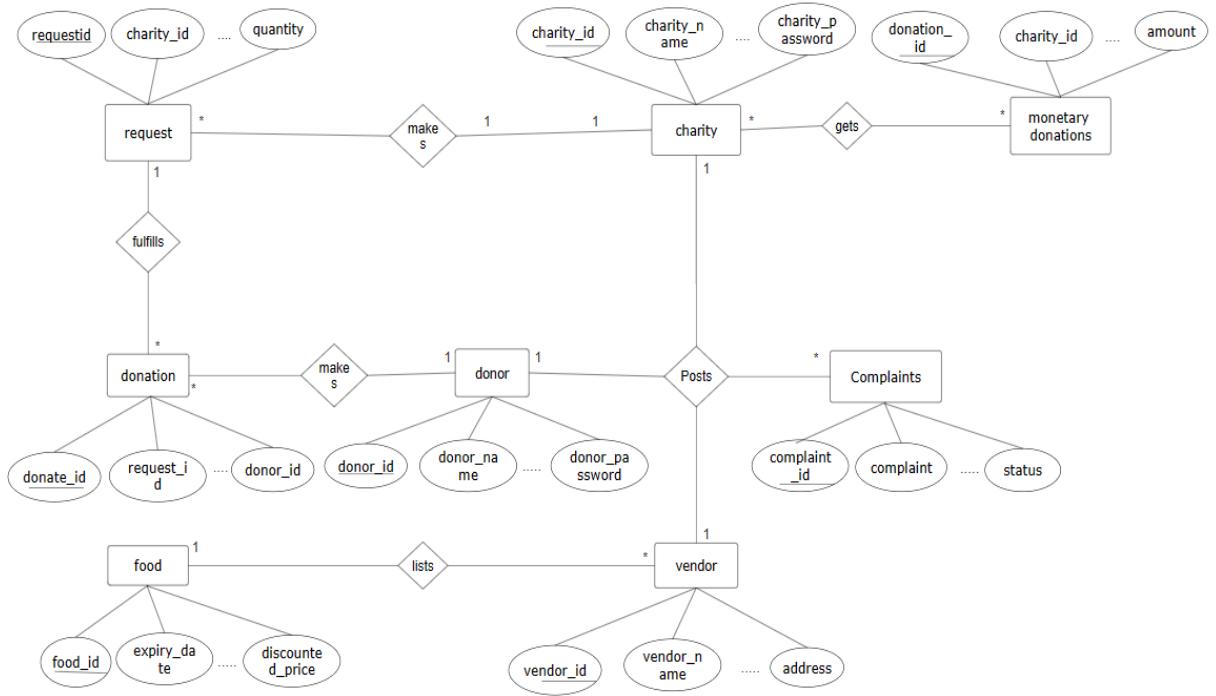
Edit Profile Page: Vendors can edit the details of the shop.

Contact Us page: Donors can register complaints

### 8.3 Theoretical background

ShareSaveSustain- food saver website is developed using one of the widely used front-end tools PHP, HTML, CSS and Bootstrap and at the back-end, we used SQL Server. The Visual Studio Code is used as the Integrated Development Environment (IDE). The Operating System used to develop this application is Windows 11

## 8.4 ER Diagram



## 8.5 Database design

Data Base Name : project

### ADMIN TABLE

S.No	Field name	Data Type	Description	Constraints
1	id	varchar	ID of admin	Primary Key, Not Null
2	password	varchar	Password of admin	Not Null

### CHARITY ORGANIZATION TABLE

S.No	Field name	Data Type	Description	Constraints
1	charity_id	varchar	ID of organization	Primary Key, Not Null
2	charity_name	varchar	Name of organization	Not Null
3	charity_email	varchar	Email Id of organization	Not Null
5	charity_password	varchar	Password of organization	Not Null
6	charity_regno	varchar	Registration no. of organization	Not Null
7	charity_address	varchar	Address of organization	Not Null
8	locality	varchar	Locality of organization	Not Null
9	status	int	Current status	Not Null

### DONOR INFO TABLE

S.No	Field name	Data Type	Description	Constraints
1	donor_id	varchar	ID of donor	Primary Key, Not Null
2	donor_name	text	Name of donor	Not Null
3	donor_password	varchar	Password of donor	Not Null
4	donor_address	varchar	Address of donor	Not Null
5	donor_dob	date	Date of birth of donor	Not Null
6	donor_email	varchar	Email of donor	Not Null

7	donor_number	varchar	Phone number of donor	Not Null
8	status	int	Current status	Not Null
9	locality	varchar	Locality of donor	Not Null

#### VENDOR INFO TABLE

S.No	Field name	Data Type	Description	Constraints
1	vendor_id	varchar	ID of vendor	Primary Key, Not Null
2	vendor_name	varchar	Name of vendor	Not Null
3	vendor_password	varchar	Password of vendor	Not Null
4	vendor_address	varchar	Vendor Address	Not Null
5	vendor_regno	varchar	Register number of vendor	Not Null
6	status	int	Current Status	Not Null

#### REQUEST TABLE

S.No	Field name	Data Type	Description	Constraints
1	request_id	int	ID of request	Primary Key, Not Null, Auto increment
2	charity_id	varchar	ID of organization	Foreign Key, Not Null
3	Request_category	varchar	Category of request	Not Null
4	Request_quantity	int	Quantity of food	Not Null
5	Request_level	varchar	Urgency level of request	Not Null
6	Request_additional	varchar	Additional information about the request	
7	Request_status	int	Status of request	Not Null
8	Request_date	date	Date of request	Not Null
9	Request_insertion_time	timestamp	Timestamp of request	Not Null

## DONATION TABLE

S.No	Field name	Data Type	Description	Constraints
1	donate_id	int	ID of donation	Primary Key, Not Null, Auto increment
2	request_id	int	ID of request	Foreign Key, Not Null
4	donor_id	varchar	ID of donor	Foreign Key,Not Null
5	quantity	date	Donation quantity	Not Null
6	status	int	Status of donation	Not Null
7	date	date	Date of donation	Not Null

## MONETARY DONATION TABLE

S.No	Field name	Data Type	Description	Constraints
1	Donation_id	int	Donation ID	Primary Key, Not Null, Auto increment
2	charity_id	varchar	Organization ID	Foreign Key, Not Null
3	name	varchar	Name of donor	Not Null
4	number	varchar	Number of donor	Not Null
5	dob	date	Date of birth of donor	Not Null
6	email	varchar	Email of donor	Not Null
7	address	varchar	Address of donor	Not Null
8	amount	int	Donated amount	Not Null

### FOOD LISTING TABLE

S.No	Field name	Data Type	Description	Constraints
1	Food_id	int	ID of item	Primary Key, Not Null, Auto increment
2	Vendor_id	varchar	ID of vendor	Foreign Key, Not Null
3	Food_name	varchar	Name of food item	Not Null
4	Food_company	varchar	Company of food item	Not Null
5	Food_category	varchar	Category of food item	Not Null
6	food_quantity	varchar	Quantity of food item	Not Null
7	Expiry_date	date	Expiry date of food item	Not Null
8	Original_price	int	Original price of food item	Not Null
9	Discounted_price	int	Discounted price of food item	Not Null
10	Food_details	varchar	Additional details of food item	

### COMPLAINT TABLE

S.No	Field name	Data Type	Description	Constraints
1	complaint_id	varchar	ID of complaint	Primary Key, Not Null
2	name	varchar	Name of user	Not Null
3	email	varchar	Email Id of user	Not Null
5	complaint	varchar	Complaint	Not Null
6	date	date	Date of complaint	Not Null
7	status	int	Status of complaint	Not Null
8	user_id	varchar	ID of users	Not Null

## 9. TESTING

### 9.1 Test cases

#### **Test Case 1: Invalid User Login**

An unregistered or unauthorized login attempt must be blocked.

**Precondition:** Unauthorized users do not have valid credentials to log in.

**Assumption:** Only authorized users have access to valid credentials for logging in.

#### **Test Steps:**

1. Go to Login Page
2. Enter incorrect credentials
3. Submit the credentials.

**Expected Result:** A login attempt with invalid or wrong input results in an unsuccessful login message.

The screenshot shows a red error message box with the text "Incorrect username or password" and an "OK" button. Below the message box is a photograph of two people preparing food in trays. To the right is a "Donor Login" form with fields for "Username" and "Password" and a "Login" button.

ShareSaveSustain

HOME DONATE LOGIN FIND SOON-TO-EXPIRE FOODS ABOUT US

Incorrect username or password

OK

Donor Login

Username:

Password:

Login

## **Test Case 2: Check results on not entering matching passwords by User at Signup page**

All the attempts to register profile must be blocked if the passwords entered by the users don't match.

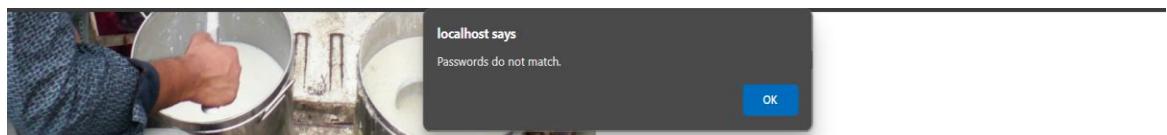
**Precondition:** User accidentally enters mismatched passwords.

**Assumption:** The Signup page is designed to require users to confirm their password to avoid errors during the registration process.

### **Test Steps:**

1. Go to the signup page and fill the details.
2. Enter mismatching password and confirm password.
3. Submit the details.

**Expected Result:** The registration form should display an error message indicating that passwords do not match.



**Donor Sign-Up**

<b>Username:</b> gayathri_123	<b>Address:</b> aluva
<b>Create Password:</b> *****	<b>Date of Birth:</b> 18-10-2001
<b>Confirm Password:</b> *****	<b>Email:</b> gayathri181001@gmail.com
<b>Name:</b> GAYATHRI DINESH	<b>Phone Number:</b> 07736306975
<input checked="" type="checkbox"/> I accept the terms and conditions.	
<b>Submit</b>	

### **Test Case 3: Check results on not storing all the required fields by User at Signup page**

All the attempts to register user must be blocked if all the required fields aren't filled by the user

**Precondition:** User accidentally forgets to fill a field.

**Assumption:** The Signup page is designed to require users to enter all necessary information to avoid errors during the registration process.

#### **Test Steps:**

1. Go to signup page.
2. Fill all the fields except for the name.
3. Click the submit button.

**Expected Result:** An attempt by the user to register without filling all or many of the required fields, results in an unsuccessful error message.

**Donor Sign-Up**

<b>Username:</b> gayathri_123	<b>Address:</b> aluva
<b>Create Password:</b> .....	<b>Date of Birth:</b> 18 - 10 - 2001
<b>Confirm Password:</b> .....	<b>Email:</b> gayathri181001@gmail.com
<b>Name:</b> <input type="text"/>	<b>Phone Number:</b> 07736306975

 Please fill out this field.

I accept the terms and conditions.

**Submit**

#### **Test Case 4: Check for an existing username during registration.**

All the attempts to send messages must be blocked if all the required fields aren't filled by the user.

**Precondition:** The system already has existing user accounts with at least one known username.

**Assumption:** The signup page is designed to prevent the creation of user accounts with duplicate usernames.

#### **Test Steps:**

1. Go to the signup page.
2. Enter an existing username and other details.
3. Submit the form.

**Expected Result:** The signup form should display an error message indicating that the entered username already exists.

**Donor Sign-Up**

Username already in use

**OK**

<b>Username:</b> <input type="text"/>	<b>Address:</b> <input type="text"/>
<b>Create Password:</b> <input type="password"/>	<b>Date of Birth:</b> <input type="text"/> dd-mm-yyyy <span style="float: right;">(i)</span>
<b>Confirm Password:</b> <input type="password"/>	<b>Email:</b> <input type="text"/>
<b>Name:</b> <input type="text"/>	<b>Phone Number:</b> <input type="text"/>

I accept the terms and conditions.

**Submit**

### **Test Case 5: Check results on validating details entered by User at Signup page**

All the attempts to register the user must be blocked if any of the fields are filled in an incorrect format.

**Precondition:** User accidentally fills details in incorrect format.

**Assumption:** The Signup page is designed to require users to enter all correct information to avoid errors during the registration process.

#### **Test Steps:**

- i. Go to signup page.
- ii. Fill all the fields.
- iii. Click the submit button by entering incorrect email address format.

**Expected Result:** The signup form should display an error message indicating the error.

The screenshot shows a web browser window with a light gray background. In the center, there is a form for user registration. At the top right of the form, a dark gray modal dialog box is displayed. The dialog box contains the text "localhost says" in white, followed by "Please enter a valid email address." in a smaller font. At the bottom right of the dialog box is a blue "OK" button. Below the dialog box, the form fields are visible. The "Username" field contains "gayathri\_123". The "Address" field contains "aluva". The "Create Password" field contains ".....". The "Date of Birth" field contains "18 - 10 - 2001". The "Confirm Password" field contains ".....". The "Email" field contains "gayathri181001@gmail". The "Name" field contains "Gayathri". The "Phone Number" field contains "7736306975". At the bottom of the form, there is a checkbox labeled "I accept the terms and conditions." followed by a "Submit" button.

## **Test Case 6: Check whether the user has selected the donation amount**

All the attempts to donate money must be blocked if the user forgets to select the donation amount.

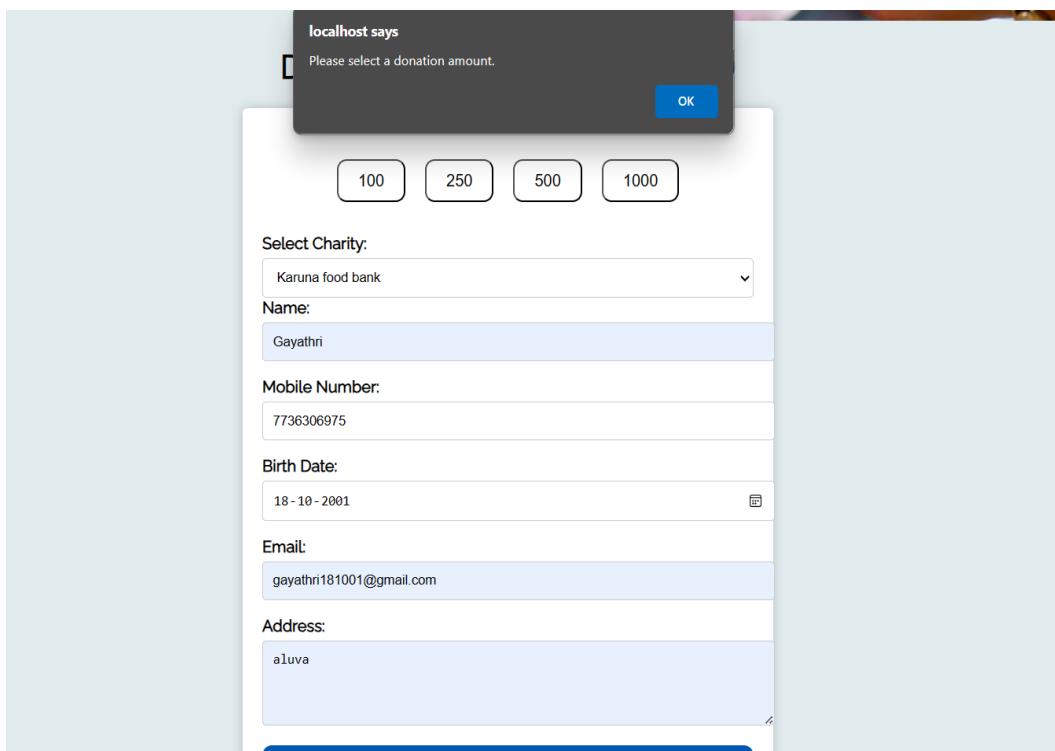
**Precondition:** User forgets to select the donation amount.

**Assumption:** The donation page requires the user to select donation amount before submitting.

### **Test Steps:**

1. Go to the donate page.
2. Fill the necessary details.
3. Submit the form without selecting the amount.

**Expected Result:** An attempt by the user to donate amount without selecting the amount, results in an unsuccessful error message.



## **Test Case 7: Charity organization forgets to enter quantity in food request**

All the attempts to request for food must be blocked if the charity organization forgets to enter the quantity.

**Precondition:** Charity organization forgets to enter the quantity.

**Assumption:** The food request form requires the charity organization to specify the quantity of food items.

### **Test Steps:**

1. Log in to the system as charity organization using the correct user's credentials.
2. Go to the Requests page and click add new request.
3. Submit the form without entering the quantity.

**Expected Result:** An attempt by the charity organization to request food without entering the quantity, results in an unsuccessful error message.

ShareSaveSustain

HOME Contact Us LOG OUT

The screenshot shows a web application interface for managing requests. On the left, there is a sidebar with links: REQUESTS, REQUEST STATUS, REQUEST HISTORY, CONTRIBUTIONS, and EDIT DETAILS. The main area has a header 'Add New Request'. The form contains fields for Request ID (26), Request Date (2023-10-13), Category/Type (Lunch), and Additional Details (only vegetarian foods). The Status is set to Inactive. A dropdown menu for Urgency is open, showing options: High (selected) and Low. A tooltip error message 'Please fill out this field.' points to the Quantity input field, which is currently empty. Below the input fields is an Additional Information text area containing 'Food for senior citizens.'. At the bottom right is a 'Submit' button.

### **Test Case 8: Check whether the user enters the desired quantity for donation**

All the attempts to donate food quantity exceeding the request quantity must be blocked if donor attempts to donate more than the request quantity.

**Precondition:** User accidentally enters the quantity more than the request quantity.

**Assumption:** Users can donate only the requested quantity.

#### **Test Steps:**

1. Log in to the system as donor using the correct user's credentials.
2. Go to the View request page.
3. Express interest to donate and submit the form by entering exceeding amount than the request amount.

**Expected Result:** An attempt by the user to donate food more than the specified quantity, results in an unsuccessful error message.

The screenshot shows the ShareSaveSustain application interface. On the left, there is a sidebar with options: VIEW REQUESTS, DONATION STATUS, DONATION HISTORY, and EDIT DETAILS. The main content area displays a "Request Details" card for a charity organization named "Karuna food bank". The card includes fields for Name of Charity Organization, Location, Category/Type, Quantity, and Urgency Level. Below this, there is an "Additional Information" section with contact details and additional instructions. A modal dialog box is overlaid on the screen, displaying the error message: "localhost says Error: Donation quantity cannot exceed the available quantity." with an "OK" button. At the top right of the main content area, there are links for HOME, Contact Us, and LOG OUT.

Quantity	Urgency Level	More Details
30	high	<a href="#">Details</a>

### **Test Case 9: Check results on validating details entered by user at Edit page**

All the attempts to edit details must be blocked if any of the fields are filled in an incorrect format.

**Precondition:** User accidentally enters a data with wrong format.

**Assumption:** The edit page requires the users to enter correct details.

#### **Test Steps:**

1. Log in to the system as Donor using the correct user's credentials.
2. Go to the Edit details page.
3. Submit the form by entering incorrect email.

**Expected Result:** An attempt by the user to edit their profile by entering wrong format, results in an unsuccessful error message.

ShareSaveSustain

HOME Contact Us LOG OUT

EDIT DETAILS

The screenshot shows the ShareSaveSustain application interface. On the left, there is a sidebar with options: VIEW REQUESTS, DONATION STATUS, DONATION HISTORY, and EDIT DETAILS. The EDIT DETAILS option is highlighted. The main area has a header 'EDIT DETAILS' and contains three input fields: Name (Gayathri), Email (gayathri), and Number (987456123). A validation error message is displayed above the Email field: 'Please include an '@' in the email address. 'gayathri' is missing an '@.''. There is also a small placeholder 'aliva' below the Email field. At the bottom is a large 'EDIT' button.

Name:	Gayathri
Email:	gayathri
Number:	987456123

EDIT

## **Test Case 10: Check results on not storing all the required fields by food vendor at listing page**

All the attempts to add new food item must be blocked if all the required fields aren't filled by the vendor.

**Precondition:** Vendor accidentally forgets to enter a detail.

**Assumption:** The listing is designed to require vendors to enter all necessary information to avoid errors.

### **Test Steps:**

1. Log in to the system as vendor using the correct user's credentials.
2. Click on the add new item.
3. Submit the form by not entering a field.

**Expected Result:** An attempt by the vendor to add new item by not entering necessary details, results in an unsuccessful error message.

The screenshot shows a web application interface for managing food items. At the top, there's a navigation bar with links for HOME, Contact Us, Edit details, and LOG OUT. Below the navigation, there's a table with columns for Image, Product Name, Company Name, Category, Price, Discounted price, Additional Details, Edit, and Delete. One row in the table is visible, showing an image of almond kernels, the product name 'Almond kernels', the company name 'Nutraj', and the category 'Dry fruits'. To the right of the table, a modal window titled 'Add New Item' is open. It contains fields for Product Name ('sunflower oil'), Company Name (empty), Category (empty), Expiry Date ('23-01-2024'), Original Price ('500'), Discounted Price ('350'), Choose Image ('Choose File product2.jpg'), and Additional Information (empty). A validation error message 'Please fill out this field.' is displayed above the Category field. The overall theme of the application is dark grey with some light blue highlights for input fields.

## 9.2 Test Report

In all the test cases, as we got the expected result, we can easily come to the conclusion that the testing process was successful and hence proved that our application is efficient enough to store and process data. Almost about 99.9% of test cases were passed successfully.

### 9..3. Sample Code used for testing

#### **Test Case 1: Invalid user login (Donor)**

```
if(isset($_POST['submit'])){

session_start();

$username = $_POST['donor_username'];

$password = $_POST['donor_password'];

$sql = "SELECT * FROM donor WHERE donor_id = '$username' AND
donor_password = '$password'";

$result = $conn->query($sql);

if ($result->num_rows == 1) {

$row = $result->fetch_assoc();

$_SESSION['user_id'] = $row['donor_id'];

header("Location: donorwelcome.php");

} else {

$_SESSION['error_message'] = "Incorrect username or password";

header("Location: login.php");

}

$conn->close();

}
```

**Test Case 2: Check results on not entering new matching passwords by User at UpdateProfile page**

```
<script>

    function validateForm() {

        var password = document.getElementById("password").value;

        var confirm_password = document.getElementById("confirm_password").value;

        if (password !== confirm_password) {

            alert("Passwords do not match.");

            return false;

        }

    }

</script>
```

**Test Case 3: Check results on not storing all the required fields by User at Signup page**

```
<form name="myform" id="myform" action="insert1.php" method="POST" onsubmit="return validateForm()">

    <div class="container2">
        <div class="column">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" required>
            <label for="password">Create Password:</label>
            <input type="password" id="password" name="password" required>
            <label for="confirm_password">Confirm Password:</label>
            <input type="password" id="confirm_password" name="confirm_password" required>
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>
        </div>
        <div class="column">
            <label for="address">Address:</label>
            <input type="text" id="address" name="address" required>
            <label for="date_of_birth">Date of Birth:</label>
            <input type="date" id="date_of_birth" name="date_of_birth" required>
        </div>
    </div>
</form>
```

```

<label for="email">Email:</label>
<input type="email" id="email" name="email" required>
<label for="phone">Phone Number:</label>
<input type="tel" id="phone" name="phone" required>
<label class="accept-conditions">
<input type="checkbox" id="accept_conditions" name="accept_conditions" required>
I accept the terms and conditions.
</label>
</div>
</div>
<input type="submit" class="submit-button" name="submit"/>
</form>

```

#### **Test Case 4: Check for an existing username during registration**

```

$check_sql = "SELECT donor_id FROM donor WHERE donor_id = '$dusername'";
$check_result = $conn->query($check_sql);
if ($check_result->num_rows > 0) {
    $_SESSION['error_message'] = "Username already in use";
    header("Location:donor.php");
    exit();
}

```

#### **Test Case 5: : Check results on validating details entered by User at Signup page**

```

<script>
function validateForm() {
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;
    var confirm_password = document.getElementById("confirm_password").value;
    var name = document.getElementById("name").value;
    var address = document.getElementById("address").value;
    var date_of_birth = document.getElementById("date_of_birth").value;
    var email = document.getElementById("email").value;
    var phone = document.getElementById("phone").value;
    var accept_conditions = document.getElementById("accept_conditions").checked;
    var usernamePattern = /^[a-zA-Z]+[a-zA-Z\d]+$/;
    if (!usernamePattern.test(username)) {

```

```

        alert("Username must contain both characters and numbers.");
        return false;
    }
    if (password.length < 8) {
        alert("Password must be at least 8 characters long.");
        return false;
    }
    if (password !== confirm_password) {
        alert("Passwords do not match.");
        return false;
    }
    var emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
    if (!emailPattern.test(email)) {
        alert("Please enter a valid email address.");
        return false;
    }
    var phonePattern = /^\d{10}$/;
    if (!phonePattern.test(phone)) {
        alert("Please enter a valid 10-digit phone number.");
        return false;
    }
    if (!accept_conditions) {
        alert("Please accept the terms and conditions.");
        return false;
    }
    return true;
}
</script>
```

#### **Test Case 6: Check whether the user has selected the donation amount**

```

form class="donation-form" method="post" action="insert7.php" onsubmit="return
validateForm()">
    <div class="form-group">
        <div class="donation-amounts">
            <button class="donation-amount" onclick="selectButton(this)">100</button>
            <button class="donation-amount" onclick="selectButton(this)">250</button>
```

```

<button class="donation-amount" onclick="selectButton(this)">500</button>
<button class="donation-amount" onclick="selectButton(this)">1000</button>
</div>
<script>
function selectButton(button) {
var buttons = document.querySelectorAll('.donation-amount');
buttons.forEach(function (btn) {
btn.classList.remove('selected');
});
button.classList.add('selected');
var selectedText = button.textContent;
console.log("Selected Amount: " + selectedText);
document.getElementById('selectedAmount').value = selectedText;
}
function validateForm() {
var selectedAmount = document.getElementById('selectedAmount').value;
if (selectedAmount === "") {
alert('Please select a donation amount.');
return false;
}
}
</script>

```

### **Test Case 7: Charity organization forgets to enter quantity in food request**

```

<h2>Add New Request</h2>
<form id="requestForm" action="insert4.php" method="POST">
<label for="category">Category/Type:</label>
<input type="text" id="category" name="category" required>
<label for="quantity">Quantity:</label>
<input type="number" id="quantity" name="quantity" required>
<label for="urgency">Urgency Level:</label>
<select id="urgency" name="urgency">
<option value="none" label="none">----</option>
<option value="High" label="high">High</option>
<option value="Medium" label="medium">Medium</option>

```

```

<option value="Low" label="low">Low</option>
</select>
<label for="additionalInfo">Additional Information:</label>
<textarea id="additionalInfo" name="additionalInfo" rows="4"></textarea>
<input type="hidden" id="urgency_label" name="urgency_label" value="">
<input type="submit" name="submit" class="add-button"/>
</form>

```

**Test Case 8: Check whether the user enters the desired quantity for donation**

```

<script>
    function validateForm() {
        var quantity = document.getElementById("quantity").innerText;
        var donationQuantity = document.getElementById("donationQuantity").value;
        quantity = parseInt(quantity);
        donationQuantity = parseInt(donationQuantity);
        if (donationQuantity > quantity) {
            alert("Error: Donation quantity cannot exceed the available quantity.");
            return false;
        }
        return true;
    }
</script>

```

**Test Case 9: Check results on validating details entered by user at Edit page**

```

<script>
    function validateForm() {
        var emailInput = document.getElementById("email");
        var numberInput = document.getElementById("number");
        // Email validation
        var emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
        if (!emailPattern.test(emailInput.value)) {
            document.getElementById("email-error").innerText = "Please enter a valid email address.";
            return false;
        }
    }
</script>

```

```

} else {

    document.getElementById("email-error").innerText = "";

}

var numberPattern = /^d{10}$/;

if (!numberPattern.test(numberInput.value)) {

    document.getElementById("number-error").innerText = "Please enter a valid 10-digit
phone number./";

    return false;

} else {

    document.getElementById("number-error").innerText = "";

}

return true;

}

</script>

```

**Test Case 10: Check results on not storing all the required fields by food vendor at listing page**

```

<h2>Add New Item</h2>
<form id="requestForm" method="post" action="insert5.php">
<label for="product">Product Name:</label>
<input type="text" id="product" name="product" required>
<label for="company">Company Name:</label>
<input type="text" id="company" name="company" required>
<label for="category">Category:</label>
<input type="text" id="category" name="category" required>
<label for="quantity">Quantity:</label>
<input type="text" id="quantity" name="quantity" required>
<label for="expiry">Expiry Date:</label>
<input type="date" id="expiry" name="expiry" required>
<label for="price">Original Price:</label>
<input type="text" id="price" name="price" required>
<label for="dprice">Discounted Price:</label>
<input type="text" id="dprice" name="dprice" required>

```

```
<br>
<label for="image">Choose Image:</label>
<input type="file" name="image" id="image" accept="image/*">
<label for="additionalInfo">Additional Information:</label>
<textarea id="additionalInfo" name="additionalInfo" rows="4"></textarea>
<button type="submit" class="add-button" name="submit">Submit</button>

</form>
```

## 10.TRANSITION

### 10.1System Implementation

Implementation is the process of having the system personal checks out and put equipment's to use, train the users to use the new system and construct any file that are needed to see it. The final and important phases in the system lifecycle are the implementation of new system. The file conversion is the most time consuming and expensive activity in the implementation stage. System implementation refers to the step necessary to install a new system to put into the operation. The implementation has different meaning, ranging from the conversion of basic application to complete replacement of computer system. Implementation includes all these activities that take place to convert from old system to new system. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale adopted is found out initially. The system is tested properly and at the same time the users are trained in new procedure. Proper implementation is essential to provide a reliable system to meet organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but it will prevent improper installation. The implementation involves the following things: -

1. Careful planning.
2. Investigation of the system and constraints
3. Design the methods to achieve the changeover.
4. Train the staff in the changed phase.
5. Evaluation of change over method.

After converting as a package, it has been delivered to the customers where it is implemented and tailored to meet the specific requirements.

## 10.2 System Maintenance

Like housework, dirty clothes and weeds, system work never seems to an end; users almost always want changes or encounter problems. Thus, the system maintenance part of the system process deserves special attention. It is during system maintenance that the analyst:

- a. Resolves necessary changes
- b. Correct errors.
- c. Enhance or modifies the system.
- d. Assign staff to perform maintenance activities.
- e. Provides for scheduled maintenance.

Most system spends the bulk of their time in the maintenance phase, with constant enhancements and repairs. Studies show that more money is spent in this forth phase than in all of the others combined. Writing system is that require as little maintenance as possible is one of the primary goal as well as one of the benefits of today's modern methodology of software development. Maintenance is divided into three categories.

1. Corrective maintenance.
2. Adaptive maintenance.
3. Preventive maintenance.

### 10.2.1 Corrective maintenance

It has to do with the removal of residual errors present in the product when it is delivered as well as errors introduced into the software during its maintenance accounts for about 20% of the maintenance cost.

### 10.2.2 Adaptive maintenance

It involves adjusting the application to changes in the environment, that is a new release of the hardware or the operating system or a new database system. It also accounts for nearly 20% of the maintenance cost.

### 10.2.3 Preventive maintenance

It involves changing the software to improve some qualities. It accounts for over 50% of maintenance costs. Here changes are due to the functions offered by the application, and new functions, improve the performance of application etc.

Maintenance is not such a difficult task. The above three maintenance tasks can be easily carried out under this system.

## 11. ANNEXURE

### 11.1 References

#### Websites

[1] YouTube

[2] Random sites

<https://W3schools.com/>

<http://www.stackoverflow.com/questions>

<http://tutorialspoint.com/>

<https://www.geekforgeeks.com>

### 11.2 Annexure I: User Interview Questionnaires

1. How would you approach the system?
2. What about usability of this system?
3. What are normal project requirements?

### 11.3 CONCLUSION

The Food Saver project successfully addresses the crucial need to streamline the process of food donation and distribution. The primary goal of the project was to create an efficient platform that connects donors, charity organizations, and food vendors to minimize food wastage and ensure that surplus food reaches those in need. Throughout the development process, key features were implemented to enhance user experience and functionality. Donors can easily register, view food requests from charity organizations, and respond by donating surplus food. Charity organizations can request food, manage donations, and view responses from donors. Food vendors have the capability to list food items close to expiry dates.

The project employs a robust and secure architecture, utilizing a relational database to store and manage data efficiently. The web-based application ensures accessibility from various devices, facilitating widespread use. Noteworthy features include real-time updates on food availability, detailed product information, and a secure authentication system. The system encourages transparency and accountability, ensuring that the donation process is seamless and trustworthy.

The software undergoes rigorous testing, employing validation techniques to eliminate errors and ensure seamless operation. The successful design and implementation of the system demonstrate its capability to provide a reliable and user-friendly platform for real estate dealings.

Embarking on this project has not only contributed to the enhancement of our technical skills but has also instilled confidence in our abilities as future IT professionals. We extend our gratitude to all those who supported us in the successful completion of this project.

## 11.4 SAMPLE CODE

### 11.4.1 Screenshots

#### HOME PAGE

ShareSaveSustain

HOME DONATE LOGIN FIND SOON-TO-EXPIRE FOODS ABOUT US



# SHARE SAVE SUSTAIN

Inspiring Impactful Change



#### WHO WE ARE

At Share Save Sustain, we are driven by a simple yet powerful mission: to combat hunger and food waste simultaneously. We believe that every meal matters, and we are dedicated to making a positive impact in our communities and beyond. Our initiative started with a heartfelt commitment to share the abundance of food resources with those in need while minimizing the staggering levels of food wastage we witness daily. With a team of passionate volunteers, donors, and partners, we have created a platform that connects surplus food from restaurants, grocery stores, and individuals with local food banks and shelters. Our innovative approach not only feeds the hungry but also addresses the urgent environmental issue of food waste. Join us in our journey to nourish the less fortunate, reduce food waste, and create a world where no one goes to bed hungry. Together, we can make a meaningful difference—one meal at a time.



Share Save Sustain

Join Us in the Fight Against Hunger.

#### Contact Us

Email: sharesavesustain@gmail.com

Phone: +91 987654321

Address: 123 Main Street, City, Country

## MONETARY DONATION PAGE

ShareSaveSustain

HOME DONATE LOGIN FIND SOON-TO-EXPIRE FOODS ABOUT US



Transform Lives with Your Generosity: Make a Donation Today

ISKCON  
DWARKA

DONATE NOW(AMOUNT IN Rs)

Select Charity:

Akshaya Patra

Name:

Gayatri

Mobile Number:

987456321

Birth Date:

03-01-2001



Email:

gayathri@gmail.com

Address:

Ernakulam

Share Save Sustain

Join Us in the Fight Against Hunger.

Contact Us

Email: sharesavesustain@gmail.com

Phone: +91 987654321

Address: 123 Main Street, City, Country

## DONATION RECEIPT

Thank You for Your Donation!

**ShareSaveSustain**

123 Main Street, City, Country

Donation ID: 12

Donation Amount: 250

Charity Id: akshayapatra123

Name: Gayatri

Mobile Number: 987456321

Birth Date: 2001-01-03

Email: gayathri@gmail.com

Address: Ernakulam

[Back to Donate](#)**FOOD LISTING PAGE**

## Rescue Delicious Fare: Discover Foods Nearing Expiry

Image	Name	Company Name	Category	Quantity	Place	More Details
	sunflower oil	Fortune	Oil	1L	Kadungallor	<a href="#">Details</a>
	Almond	Almond Kernels	Dry fruits	250g	Ernakulam	<a href="#">Details</a>

## Share Save Sustain

Join Us in the Fight Against Hunger.

## Contact Us

Email: sharesavesustain@gmail.com

Phone: +91 987654321

Address: 123 Main Street, City, Country

## LOGIN PAGE

ShareSaveSustain

HOME DONATE LOGIN FIND SOON-TO-EXPIRE FOODS ABOUT US



### Donor Login

Username:

Password:

**Login**

### Charity Organization Login

Username:

Password:

**Login**



### Food Vendor Login

Username:

Password:

**Login**

### Share Save Sustain

Join Us in the Fight Against Hunger.

#### Contact Us

Email: sharesavesustain@gmail.com

Phone: +91 987654321

Address: 123 Main Street, City, Country

## DONOR REGISTRATION PAGE

ShareSaveSustain

HOME DONATE LOGIN FIND SOON-TO-EXPIRE FOODS ABOUT US



### Join Us in Making a Difference: Sign Up as a Donor Today

Your generosity can change lives. By becoming a donor, you play a vital role in supporting those in need. Sign up today to make a meaningful contribution and help us in our mission to alleviate hunger and provide assistance to those facing food insecurity. Together, we can make a significant difference in our community. Join our community of caring individuals and start making an impact today.

### Donor Sign-Up

<b>Username:</b> <input type="text" value="bf492FsR"/>	<b>Address:</b> <input type="text" value="Kalamassery"/>
<b>Create Password:</b> <input type="password" value="*****"/>	<b>Date of Birth:</b> <input type="text" value="18-10-2001"/>
<b>Confirm Password:</b> <input type="password" value="*****"/>	<b>Email:</b> <input type="text" value="gayathri@gmail.com"/>
<b>Name:</b> <input type="text" value="Gayathri"/>	<b>Phone Number:</b> <input type="text" value="987456321"/>
<input type="checkbox"/> I accept the terms and conditions.	
<input type="button" value="Submit"/>	

### Share Save Sustain

Join Us in the Fight Against Hunger.

#### Contact Us

Email: sharesavesustain@gmail.com

Phone: +91 987654321

Address: 123 Main Street, City, Country

## CHARITY ORGANIZATION REGISTRATION PAGE

ShareSaveSustain

HOME DONATE LOGIN FIND SOON-TO-EXPIRE FOODS ABOUT US



### Join Our Cause: Register Your Charity Organization Today!

Your charity organization plays a crucial role in making the world a better place. By registering with us, you gain access to a platform that connects you with donors, volunteers, and resources to support your mission. Join our community of like-minded organizations dedicated to creating positive change. Together, we can amplify the impact of your charitable efforts and contribute to a brighter future for those in need.

#### Organization Sign-Up

**Organization Name:**

Akshaya Patra

**Email:**

akshayapatra@gmail.com

**Username:**

akshayapatra

**Password:**

\*\*\*\*\*

**Confirm Password:**

\*\*\*\*\*

**Registration Number:**

43128762

**Address:**

ernakulam

**Submit**

#### Share Save Sustain

Join Us in the Fight Against Hunger.

#### Contact Us

Email: sharesavesustain@gmail.com

Phone: +91 987654321

Address: 123 Main Street, City, Country

## FOOD VENDOR REGISTRATION PAGE

ShareSaveSustain

HOME DONATE LOGIN FIND SOON-TO-EXPIRE FOODS ABOUT US



### Join the Fight Against Food Waste: List Your Excess Inventory Today!

Your partnership can make a significant difference in the battle against food waste. By signing up as a food vendor, you're contributing to a sustainable future where excess food finds its way to those in need rather than going to waste. Help us bridge the gap between surplus and scarcity in our community. Listing food items close to their expiry date not only reduces waste but also supports local food initiatives. Together, we can create a world where good food is shared and enjoyed by all!

#### Vendor Sign-Up

**Vendor ID:**  
Reliance

**Name:**  
Reliance Supermarket

**Password:**  
.....

**Confirm Password:**  
.....

**Address:**  
Ernakulam

**Registration Number:**  
AE8742DD

**Submit**

#### Share Save Sustain

Join Us in the Fight Against Hunger.

#### Contact Us

Email: sharesavesustain@gmail.com

Phone: +91 987654321

Address: 123 Main Street, City, Country

## ABOUT US PAGE

ShareSaveSustain

HOME DONATE LOGIN FIND SOON-TO-EXPIRE FOODS ABOUT US



## ABOUT US

Welcome to ShareSaveSustain, a platform dedicated to making a positive impact on the world by addressing the critical issue of food waste and hunger. At ShareSaveSustain, we believe in the power of community and collaboration to create meaningful change. Our mission is to bridge the gap between excess food and those in need, fostering a sustainable and compassionate ecosystem. Through our innovative approach, we connect donors, charitable organizations, and volunteers, creating a network that shares, saves, and sustains resources. By leveraging technology and the collective efforts of individuals, we strive to reduce food waste, support local communities, and make a lasting difference in the fight against hunger. Join us in our commitment to building a better and more sustainable future for all. Together, we can Share, Save, and Sustain.

Feel free to drop us a line below

Your Name:

Your Email:

Your Message:

**Submit**

### Share Save Sustain

Join Us in the Fight Against Hunger.

#### Contact Us

Email: sharesavesustain@gmail.com

Phone: +91 987654321

Address: 123 Main Street, City, Country

## DONOR HOME PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

WELCOME GAYATHRI



### Total Donations

20

### Number of Charities Supported

1

### Largest Single Donation

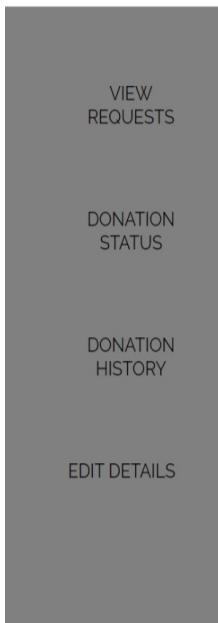
20

## VIEW REQUEST PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

### DONATION REQUESTS



Name of Charity Organization	Address	Category/Type	Quantity	Urgency Level	More Details
Akshaya Patra	ernakulam	Breakfast	50	high	<a href="#">Details</a>
Karuna food bank	ernakulam	Lunch	11	medium	<a href="#">Details</a>

## DONATION REQUESTS

Name of Charity Organization	Quantity	Urgency Level	More Details
Akshaya Patra	50	high	<a href="#">Details</a>
Karuna food bank	11	medium	<a href="#">Details</a>

## Request Details

Name of Charity Organization: Akshaya Patra  
 Location: ernakulam  
 Category/Type: Breakfast  
 Quantity: 50  
 Urgency Level: high

## Additional Information

Contact Information: akshayapatra@gmail.com  
 Additional Instructions: Only vegetarian foods

Express Interest

Quantity to Donate:

[Submit](#)

VIEW  
REQUESTS

DONATION  
STATUS

DONATION  
HISTORY

EDIT DETAILS

DONATION STATUS PAGE

## DONATION STATUS

Donation id	Charity name	Quantity	Delete
58	Akshaya Patra	20	<a href="#">Delete</a>

VIEW  
REQUESTS

DONATION  
STATUS

DONATION  
HISTORY

EDIT DETAILS

## DONATION HISTORY PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

### DONATION HISTORY



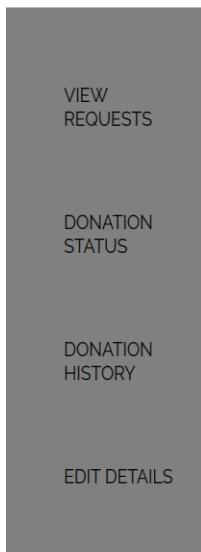
Name of Charity Organization	Location	Request Date	Quantity Donated
Akshaya Patra	ernakulam	03-03-2024	20

## DONOR EDIT PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

### EDIT DETAILS



<b>Name:</b>	<input type="text" value="GAYATHRI"/>
<b>Email:</b>	<input type="text" value="gayathri181001@gmail.com"/>
<b>Address:</b>	<input type="text" value="aluva"/>
<b>Number:</b>	<input type="text" value="7736306975"/>
<input type="button" value="EDIT"/>	

## CHARITY ORGANIZATION HOME PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

WELCOME Karuna food bank



Total Requests

3

Number of donors

2

Total donor food contributions

33

Total donation amount received

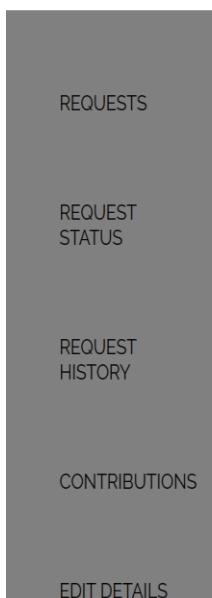
Rs.750

## REQUEST PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

Add New Request



Request ID	Request Date	Category/Type	Quantity	Urgency Level	Additional Details	Status	Edit	Delete
26	2023-10-13	dinner	32	medium	only vegetarian foods	Inactive		
31	2024-01-12	Lunch	5	high	Food for senior citizens.	Active		

[Add New Request](#)

REQUESTS

REQUEST STATUS

REQUEST HISTORY

CONTRIBUTIONS

EDIT DETAILS

**Add New Request**

Category/Type:

Breakfast

Quantity:

50

Urgency Level:

high

Additional Information:

Only Vegetarian foods

[Submit](#)**REQUEST STATUS PAGE****REQUEST STATUS**

REQUESTS

REQUEST STATUS

REQUEST HISTORY

CONTRIBUTIONS

EDIT DETAILS

Request id	Donor name	Donor Address	Quantity	Donated	Not Donated
31	GAYATHRI	aluva	5	<a href="#">Y</a>	<a href="#">X</a>

## REQUEST HISTORY PAGE

ShareSaveSustain

[HOME](#) [Contact Us](#) [LOG OUT](#)

### REQUEST HISTORY



Request id	Request Date	Donor name	Donor number	Quantity
26	2023-10-20	Gayathri	987456123	8
31	2024-01-12	GAYATHRI	7736306975	20

## MONETARY DONATIONS

ShareSaveSustain

[HOME](#) [Contact Us](#) [LOG OUT](#)

### CONTRIBUTIONS



Donor name	Donor number	Amount
Gayathri	7736306975	250
Varun	0994695114	500

## CHARITY ORGANIZATION EDIT PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

### EDIT DETAILS



Name:	Karuna food bank
Email:	karuna@gmail.com
Address:	ernakulam
<b>EDIT</b>	

## FOOD VENDOR HOME PAGE

ShareSaveSustain

HOME Contact Us Edit details LOG OUT

WELCOME abc1

Add New Item

Image	Product Name	Company Name	Category	Quantity	Expiry date	original price	Discounted price	Additional Details	Edit	Delete
	Almond kernels	Nutraj	Dry fruits	500g	2024-01-26	500	300			

ShareSaveSustain

No data found

HOME Contact Us Edit details LOG OUT

Add New Item

Product Name: Almond

Company Name: Almond Kernels

Category: Dry fruits

Quantity: 250g

Expiry Date: 06 - 03 - 2024

Original Price: 200

Discounted Price: 150

Choose Image: Choose File product1.jpg

Additional Information:

Add New Item

## FOOD VENDOR EDIT PAGE

ShareSaveSustain

HOME Contact Us Edit details LOG OUT

Edit details

Name: Margin free

Address: Kadungallor

EDIT

## COMPLAINT PAGE

ShareSaveSustain

[HOME](#) [Contact Us](#) [LOG OUT](#)

Register your complaint

Your Message:

**Submit**

**Share Save Sustain**

Join Us in the Fight Against Hunger.

**Contact Us**

Email: sharesavesustain@gmail.com

Phone: +91 987654321

Address: 123 Main Street, City, Country

## ADMIN LOGIN PAGE

Login to ShareSaveSustain

Username:

Password:

**Login**

## ADMIN WELCOME PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

WELCOME ADMIN



Total Donors

1

Total Charity Organizations

1

Total Vendors

1

Total Complaints

1

## VIEW DONORS PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

WELCOME ADMIN



Name of User	Location	Contact Number	Status	Action
Gayathri	Aluva	987456123	Active	BLOCK
Ram	Kochi	987456124	Active	BLOCK

## VIEW ORGANIZATIONS PAGE

ShareSaveSustain

[HOME](#) [Contact Us](#) [LOG OUT](#)

WELCOME ADMIN



Name of Charity Organization	Location	Email	Status	Action
Akshaya Patra	Ernakulam	akshayapatra@gmail.com	Active	<a href="#">BLOCK</a>
Karuna Food Bank	Ernakulam	karuna@gmail.com	Active	<a href="#">BLOCK</a>

## VIEW VENDORS PAGE

ShareSaveSustain

[HOME](#) [Contact Us](#) [LOG OUT](#)

WELCOME ADMIN



Name of Vendor	Location	Contact Number	Status	Action
Margin Free	Ernakulam	987456123	Active	<a href="#">BLOCK</a>

## VIEW VENDORS PAGE

ShareSaveSustain

HOME Contact Us LOG OUT

WELCOME ADMIN



Complaint ID	Name	Date	Status	Details
1	Gayathri	06-03-2024	Active	<a href="#">More details</a>
2	Dhruv	07-03-2024	Active	<a href="#">More details</a>

#### 11.4.2 Code

##### Request page of organization

```
<?php

$conn = new mysqli("localhost", "root", "", "project");

$sql = "SELECT request_id, request_category, request_quantity, request_level,
request_additional,request_status,request_date FROM request WHERE charity_id =
'$userid' AND (request_status = 1 OR (request_status = 0 AND request_quantity > 0))
ORDER BY request_date DESC";

$result = $conn->query($sql);

if ($result->num_rows > 0) {

?>

<table>

<thead>

<tr>

<th>Request ID</th>
<th>Request Date</th>
<th>Category/Type</th>
<th>Quantity</th>
<th>Urgency Level</th>
<th>Additional Details</th>
<th>Status</th>
<th>Edit</th>
<th>Delete</th>

</tr>
</thead>
<tbody>

<?php

while ($row = $result->fetch_assoc()) {

echo "<tr>";
$category=$row["request_category"];
$quantity=$row["request_quantity"];
$level=$row["request_level"];
$add= $row["request_additional"];
$id=$row["request_id"];
$requestStatus = $row["request_status"];
```

```

echo "<td>" . $row["request_id"] . "</td>";
$formattedDate = date('d-m-Y', strtotime($row["request_date"]));
echo "<td>" . $formattedDate. "</td>";
echo "<td>" . $row["request_category"] . "</td>";
echo "<td>" . $row["request_quantity"] . "</td>";
echo "<td>" . $row["request_level"] . "</td>";
echo "<td>" . $row["request_additional"] . "</td>";
$labelText = ($requestStatus == 1) ? 'Active' : 'Inactive';
$labelBackgroundColor = ($requestStatus == 1) ? '#4CAF50' : '#FF5733';
if ($requestStatus == 0 && $quantity > 0) {
    $labelText = 'Inactive';
    $labelBackgroundColor = '#FF5733';
}
echo '<td><span style="display: inline-block; padding: 4px 8px; border-radius: 4px; background-color: ' . $labelBackgroundColor . '; color: white; text-align: center;">' .
$labelText . '</span></td>';
echo '<td><span class="edit-icon" onclick="openEditPopup(' . $id . ', \'' . $category .
'\', \'' . $quantity . ', \'' . $level . '\', \'' . $add . '\', \'' . $requestStatus .
'\')">\#9998;</span></td>';
echo '<td><span class="delete-icon"><a href="deleterequest.php?rid=' . $id .
'">\#10006;</span></td>';
echo "</tr>";
}
?>
</tbody>
</table>
<?php
} else {
    echo "<center>No data found</center>";
}
$conn->close();
?>
<div class="overlay" id="overlayAdd" onclick="closePopup()"></div>
<div class="popup" id="popupAdd">
    <span class="close-btn" onclick="closePopup()">&times;</span>
    <h2>Add New Request</h2>

```

```

<form id="requestForm" action="insert4.php" method="POST">
    <label for="category">Category/Type:</label>
    <input type="text" id="category" name="category" required>
    <label for="quantity">Quantity:</label>
    <input type="number" id="quantity" name="quantity" required>
    <label for="urgency">Urgency Level:</label>
    <select id="urgency" name="urgency">
        <option value="none" label="none">----</option>
        <option value="High" label="high">High</option>
        <option value="Medium" label="medium">Medium</option>
        <option value="Low" label="low">Low</option>
    </select>
    <label for="additionalInfo">Additional Information:</label>
    <textarea id="additionalInfo" name="additionalInfo" rows="4"></textarea>
    <input type="hidden" id="urgency_label" name="urgency_label" value="">
    <input type="submit" name="submit" class="add-button"/>
</form>
</div>
<script>
document.getElementById('urgency').addEventListener('change', function() {
    var selectedOption = this.options[this.selectedIndex].getAttribute('label');
    document.getElementById('urgency_label').value = selectedOption;
});
</script>
<div class="overlay" id="overlayEdit" onclick="closeEditPopup()"></div>
<div class="popup" id="popupEdit">
    <span class="close-btn" onclick="closeEditPopup()">&times;</span>
    <h2>Edit Request</h2>
    <form id="editRequestForm" action="requestedit.php" method="POST">
        <label for="editCategory">Category/Type:</label>
        <input type="text" id="editCategory" name="category" required>
        <label for="editQuantity">Quantity:</label>
        <input type="number" id="editQuantity" name="quantity" required>
        <label for="editUrgency">Urgency Level:</label>
        <input type="text" id="editUrgency" name="editUrgency" required>
        <label for="editAdditionalInfo">Additional Information:</label>

```

```

<textarea id="editAdditionalInfo" name="additionalInfo" rows="4"></textarea>
<label for="requestStatus">Request Status:</label>
<p>To make it active/inactive set the status to 1/0 </p>
<input type="number" id="requestStatus" name="requestStatus" required>
    <input type="hidden" id="request_id" name="request_id">
    <input type="hidden" id="urgency_label2" name="urgency_label2" value="">
    <input type="submit" name="submit" class="add-button">
</form>
</div>
<script>
function openEditPopup(requestId, category, quantity, urgency, additionalInfo, requestStatus)
{
    document.getElementById('overlayEdit').style.display = 'block';
    document.getElementById('popupEdit').style.display = 'block';
    document.getElementById('request_id').value = requestId;
    document.getElementById('editCategory').value = category;
    document.getElementById('editQuantity').value = quantity;
    document.getElementById('editUrgency').value = urgency;
    document.getElementById('editAdditionalInfo').value = additionalInfo;
    document.getElementById('requestStatus').value = requestStatus;
}
function closeEditPopup() {
    document.getElementById('overlayEdit').style.display = 'none';
    document.getElementById('popupEdit').style.display = 'none';
}
</script>
<script>
function openPopup() {
    document.getElementById('overlayAdd').style.display = 'block';
    document.getElementById('popupAdd').style.display = 'block';
}
function closePopup() {
    document.getElementById('overlayAdd').style.display = 'none';
    document.getElementById('popupAdd').style.display = 'none';
    document.getElementById('popupEdit').style.display = 'none';
}
</script>
insert4.php
<?php
session_start();

```

```

if(isset($_POST['submit'])) {
    $charityid = $_SESSION['userid'];
    $category=$_POST['category'];
    $quantity = $_POST['quantity'];
    $urgency_label = $_POST['urgency_label'];
    $add = $_POST['additionalInfo'];
    $conn = new mysqli("localhost", "root", "", "project");
    $sql = "INSERT INTO request (charity_id, request_category, request_quantity,
        request_level, request_additional,request_status,request_date,request_insertion_time)
        VALUES ('$charityid', '$category', $quantity, '$urgency_label',
        '$add',1,CURDATE(),NOW())";
    if ($conn->query($sql) === TRUE) {
        header("Location: request.php?success=1");
        exit();
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
    $conn->close();
}
?>

Request status page

<?php

$conn = new mysqli("localhost", "root", "", "project");
$sql = "SELECT
            dn.request_id AS request_id,
            d.donor_name AS donor_name,
            d.donor_address AS donor_address,
            dn.quantity AS quantity,
            dn.status AS status,
            dn.donate_id AS donate_id
        FROM
            donate dn
        LEFT JOIN
            donor d ON d.donor_id = dn.donor_id
        WHERE
            dn.charity_id = '$user_id' AND dn.status=0" ;

```

```

$result = $conn->query($sql);
if ($result->num_rows > 0) {
?>
<table>
<thead>
<tr>
<th>Request id</th>
<th>Donor name</th>
<th>Donor Address</th>
<th>Quantity </th>
<th>Donated</th>
<th>Not Donated</th>
</tr>
</thead>
<tbody>
<?php
while ($row = $result->fetch_assoc()) {
    $requestId = $row["request_id"];
    $donateId = $row["donate_id"];
    $quantity = $row["quantity"];
    $linkText = "Y";
    $linkURL = "donated.php?param=". urlencode($donateId) . "";
    $linkText2 = "X";
    $linkURL2 = "deletedonation2.php?param=". urlencode($donateId) . "";
    echo "<tr>";
    echo "<td>" . $row["request_id"] . "</td>";
    echo "<td>" . $row["donor_name"] . "</td>";
    echo "<td>" . $row["donor_address"] . "</td>";
    echo "<td>" . $row["quantity"] . "</td>";
    echo "<td>'<a href='donated.php?did=' . $donateId . '">' . $linkText . '</a>';
    echo "<td>'<a href='deletedonation2.php?did=' . $donateId . '">' . $linkText2 . '</a>';
    echo "</td>";
    echo "</tr>";
}
} else {
echo "No data found.";

```

```

        }
        $conn->close();
    ?>
    </tbody>
    </table>
donated.php
<?php
if (isset($_GET['did'])) {
    $donateId = $_GET['did'];
    $conn = new mysqli("localhost", "root", "", "project");
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $updateDonateSql = "UPDATE donate SET status = 1 WHERE donate_id = ?";
    $updateDonateStmt = $conn->prepare($updateDonateSql);
    $updateDonateStmt->bind_param("i", $donateId);
    if ($updateDonateStmt->execute()) {
        echo "Donation status updated successfully";
    } else {
        echo "Error updating donation status: " . $conn->error;
    }
    $updateDonateStmt->close();
    $conn->close();
    header("Location: requeststatus.php");
} else {
    echo "Invalid request";
}
?>
deletedonation.php
<?php
if (isset($_GET['did'])) {
    $donateId = $_GET['did'];
    $conn = new mysqli("localhost", "root", "", "project");
    $sql="SELECT quantity,charity_id FROM donate WHERE donate_id='$donateId'";
    $result = $conn->query($sql);
    if ($result->num_rows == 1) {

```

```

$row = $result->fetch_assoc();
$quantity= $row['quantity'];
$charityId=$row['charity_id'];
}

$updateRequestSql = "UPDATE request SET request_quantity = request_quantity + ?,
request_status = CASE WHEN request_quantity > 0 THEN 1 ELSE request_status
END WHERE charity_id = ? ";
if ($updateRequestStmt = $conn->prepare($updateRequestSql)) {
    $updateRequestStmt->bind_param("is", $quantity, $charityId);
    if (!$updateRequestStmt->execute()) {
        die("Request update failed: " . $updateRequestStmt->error);
    }
    $updateRequestStmt->close();
} else {
    die("Request prepare statement failed: " . $conn->error);
}
$donateSql = "DELETE FROM donate WHERE donate_id=? ";
if ($donateStmt = $conn->prepare($donateSql)) {
    $donateStmt->bind_param("s", $donateId);
    if (!$donateStmt->execute()) {
        die("Donation deletion failed: " . $donateStmt->error);
    }
    $donateStmt->close();
} else {
    die("Donation prepare statement failed: " . $conn->error);
}
$conn->close();
header("Location: requeststatus.php");
}
else
{
echo "Parameters not set.";
}
?>

```

### View Request Page of Donors

```
<?php  
$conn = new mysqli("localhost", "root", "", "project");  
$sql = "SELECT  
        charity.charity_name AS charity_name,  
        charity.charity_address AS charity_address,  
        request.charity_id AS charity_id,  
        request.request_id AS request_id,  
        request.request_category AS request_category,  
        request.request_quantity AS request_quantity,  
        request.request_level AS request_level,  
        charity.charity_email AS charity_email,  
        request.request_id AS request_id,  
        request.request_additional AS request_additional  
    FROM charity  
    INNER JOIN request ON charity.charity_id = request.charity_id  
    WHERE request.request_status = 1";  
$result = $conn->query($sql);  
if ($result->num_rows > 0) {  
?>  
    <table>  
        <thead>  
            <tr>  
                <th>Name of Charity Organization</th>  
                <th>Address</th>  
                <th>Category/Type</th>  
                <th>Quantity</th>  
                <th>Urgency Level</th>  
                <th>More Details</th>  
            </tr>  
        </thead>  
        <tbody>  
            <?php  
            while ($row = $result->fetch_assoc()) {  
                $charityId = $row["charity_id"];  
                $requestId = $row["request_id"];  
            }  
        </tbody>  
    </table>
```

```

echo "<tr>";
echo "<td>" . $row["charity_name"] . "</td>";
echo "<td>" . $row["charity_address"] . "</td>";
echo "<td>" . $row["request_category"] . "</td>";
echo "<td>" . $row["request_quantity"] . "</td>";
echo "<td>" . $row["request_level"] . "</td>";
echo '<td><span class="details-link" onclick="openDetails(' .
$row["charity_name"] . '\', \' . $row["charity_address"] . '\', \' .
$row["request_category"] . '\', \' . $row["request_quantity"] . '\', \' .
$row["request_level"] . '\', \' . $row["charity_email"] . '\', \' . $row["request_additional"] .
'\', \' . $charityId. '\', \' . $requestId. '\', \' . $user_id. '\')">Details</span></td>';
echo "</tr>";
}
?>
</tbody>
</table>
<div id="details-popup" class="details-popup">
<span class="close-btn" onclick="closeDetails()">&times;</span>
<h2>Request Details</h2>
<form method="post" action="process_donation.php" onsubmit="return
validateForm()">
<p class="text1">Name of Charity Organization: <span id="orgName"></span></p>
<p class="text1">Location: <span id="address"></span></p>
<p class="text1">Category/Type: <span id="category"></span></p>
<p class="text1">Quantity: <span id="quantity"></span></p>
<p class="text1">Urgency Level: <span id="urgency"></span></p>
<hr>
<h2>Additional Information</h2>
<p class="text1">Contact Information: <span id="contactInfo"></span></p>
<p class="text1">Additional Instructions: <span id="instructions"></span></p>
<br>
<label>
    <input type="checkbox" id="expressInterest"> Express Interest
</label>
<br>
<label for="donationQuantity">Quantity to Donate:</label>

```

```

<br>
<input type="number" id="donationQuantity"
name="donationQuantity"placeholder="Enter quantity">
<br>
<input type="hidden" id="donorId" name="donorId" >
<input type="hidden" id="requestId" name="requestId">
<input type="hidden" id="charityId" name="charityId">
<button class="submit-button" name="submit">Submit</button>
</div>
<script>
    function openDetails(charityName, address, category, quantity, urgency, contactInfo,
instructions,charityid,requestid,userid) {
        document.getElementById("orgName").textContent = charityName;
        document.getElementById("address").textContent = address;
        document.getElementById("category").textContent = category;
        document.getElementById("quantity").textContent = quantity;
        document.getElementById("urgency").textContent = urgency;
        document.getElementById("contactInfo").textContent = contactInfo;
        document.getElementById("instructions").textContent = instructions;
        document.getElementById("requestId").value = requestid;
        document.getElementById("donorId").value = userid;
        document.getElementById("charityId").value = charityid;
        var popup = document.getElementById("details-popup");
        popup.style.display = "block";
    }
    function closeDetails() {
        var popup = document.getElementById("details-popup");
        popup.style.display = "none";
    }
    function validateForm() {
        // Get the values of quantity and donationQuantity
        var quantity = document.getElementById("quantity").innerText;
        var donationQuantity = document.getElementById("donationQuantity").value;
        quantity = parseInt(quantity);
        donationQuantity = parseInt(donationQuantity);
        if (donationQuantity > quantity) {

```

```

        alert("Error: Donation quantity cannot exceed the available quantity.");
        return false;
    }
    return true;
}
</script>
<?php
} else {
    echo "No data found.";
}

$conn->close();
?>
process_donation.php
<?php
session_start();
if (isset($_POST['submit'])) {
    $charityId = $_POST['charityId'];
    $donorId = $_POST['donorId'];

    $requestId = $_POST['requestId'];
    $donationQuantity = $_POST['donationQuantity'];
    $conn = new mysqli("localhost", "root", "", "project");
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $donateSql = "INSERT INTO donate (charity_id, donor_id, quantity,request_id,date)
                 VALUES (?, ?, ?,?,CURDATE())";
    $updateRequestSql = "UPDATE request SET request_quantity = request_quantity - ?,
                                request_status = CASE WHEN request_quantity = 0 THEN 0 ELSE request_status
                                END WHERE request_id = ? AND request_status = 1";
    if ($donateStmt = $conn->prepare($donateSql)) {
        $donateStmt->bind_param("ssii", $charityId, $donorId, $donationQuantity,$requestId);
        if (!$donateStmt->execute()) {
            die("Donation insertion failed: " . $donateStmt->error);
        }
        $donateStmt->close();
    }
}

```

```

} else {
    die("Donation prepare statement failed: " . $conn->error);
}

if ($updateRequestStmt = $conn->prepare($updateRequestSql)) {
    $updateRequestStmt->bind_param("ii", $donationQuantity, $requestId);
    if (!$updateRequestStmt->execute()) {
        die("Request update failed: " . $updateRequestStmt->error);
    }
    $updateRequestStmt->close();
} else {
    die("Request prepare statement failed: " . $conn->error);
}
$conn->close();
header("Location: viewrequest.php?success=1");
}

?>

Donation Status Page

<?php

if (isset($_GET['success']) && $_GET['success'] == 1 ) {
    echo '<div class="center-container">
<div class="success-messaage">
<p>Deleted successfully</p>
<a href="donationstatus.php" class="ok-button">OK</a>
</div> </div>';
}

$conn = new mysqli("localhost", "root", "", "project");
$sql = "SELECT
        dn.date AS request_date,
        dn.quantity AS quantity,
        dn.status AS status,
        dn.donate_id AS donate_id,
        c.charity_name AS charity_name,
        c.charity_id AS charity_id,
        dn.donor_id AS donor_id
    FROM
        donate dn
"

```

```

JOIN
    charity c ON c.charity_id = dn.charity_id
WHERE
    dn.donor_id = '$user_id' AND status=0" ;
$result = $conn->query($sql);
if ($result->num_rows > 0) {
?>
<table>
<thead>
<tr>
    <th>Donation id</th>
    <th>Charity name</th>
    <th>Quantity </th>
    <th>Delete </th>
</tr>
</thead>
<tbody>
<?php
while ($row = $result->fetch_assoc()) {
    $donateId = $row["donate_id"];
    $charityId = $row["charity_id"];
    $quantity = $row["quantity"];
$linkText = "Delete";
echo "<tr>";
echo "<td>" . $row["donate_id"] . "</td>";
echo "<td>" . $row["charity_name"] . "</td>";
echo "<td>" . $row["quantity"] . "</td>";
echo '<td><a href="deletedonation.php?did=' . $donateId . '">' . $linkText .
'</a></td>';
echo "</tr>";
}
} else {
echo "No data found.";
}
$conn->close();
?>

```

```

        </tbody>
    </table>
deletedonation.php
<?php
if (isset($_GET['did'])) {
    $donateId = $_GET['did'];
    $conn = new mysqli("localhost", "root", "", "project");
    $sql="SELECT quantity,charity_id FROM donate WHERE donate_id='$donateId'";
    $result = $conn->query($sql);
    if ($result->num_rows == 1) {
        $row = $result->fetch_assoc();
        $quantity= $row['quantity'];
        $charityId=$row['charity_id'];
    }
    $updateRequestSql = "UPDATE request SET request_quantity = request_quantity + ?,
    request_status = 1 WHERE charity_id = ? ";
    if ($updateRequestStmt = $conn->prepare($updateRequestSql)) {
        $updateRequestStmt->bind_param("is", $quantity, $charityId);
        if (!$updateRequestStmt->execute()) {
            die("Request update failed: " . $updateRequestStmt->error);
        }
        $updateRequestStmt->close();
    } else {
        die("Request prepare statement failed: " . $conn->error);
    }
    $donateSql = "DELETE FROM donate WHERE donate_id=? ";
    if ($donateStmt = $conn->prepare($donateSql)) {
        $donateStmt->bind_param("s", $donateId);
        if (!$donateStmt->execute()) {
            die("Donation deletion failed: " . $donateStmt->error);
        }
        $donateStmt->close();
    } else {
        die("Donation prepare statement failed: " . $conn->error);
    }
    $conn->close();
}

```

```

        header("Location: donationstatus.php?success=1");
    }
else
{
    echo "Parameters not set.";
}
?>

FOOD LISTING PAGE OF VENDOR

<?php

$conn = new mysqli("localhost", "root", "", "project");

$sql = "SELECT vendor_name FROM vendor WHERE vendor_id = '$vendorid'";

$result = $conn->query($sql);

if ($result->num_rows == 1) {

    $row = $result->fetch_assoc();

    $vendorname= $row['vendor_name'];

}

$conn->close();

?>

<h2>WELCOME <?php echo $vendorname; ?></h2>

<div class="add-button-container">

    <button class="add-button" onclick="openPopup()">Add New Item</button>

</div>

<?php

global $Root;

$Root= $_SERVER['DOCUMENT_ROOT'];

$conn = new mysqli("localhost", "root", "", "project");

$sql = "SELECT

    food_id,food_name,food_company,food_category,food_quantity,expiry_date,original_p
rice,discounted_price,food_details,image FROM food WHERE vendor_id =
'$vendorid';

$result = $conn->query($sql);

if ($result->num_rows > 0) {

?>

<table>

    <thead>

        <tr>

```

```

<th>Image</th>
<th>Product Name</th>
<th>Company Name</th>
<th>Category</th>
<th>Quantity</th>
<th>Expiry date</th>
<th>original price</th>
<th>Discounted price</th>
<th>Additional Details</th>
<th>Edit</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<?php
while ($row = $result->fetch_assoc()) {
    $id=$row["food_id"] ;
    $name=$row["food_name"] ;
    $company=$row["food_company"];
    $category=$row["food_category"];
    $quantity=$row["food_quantity"] ;
    $date= $row["expiry_date"];
    $oprice= $row["original_price"];
    $dprice=$row["discounted_price"] ;
    $details= $row["food_details"] ;
    $Image = $row["image"];
    echo "<tr>";
    echo '<td><img src="" . $Image. "" alt="Product Image" style="max-width: 100px; max-height: 100px;"></td>';
    echo "<td>" . $row["food_name"] . "</td>";
    echo "<td>" . $row["food_company"] . "</td>";
    echo "<td>" . $row["food_category"] . "</td>";
    echo "<td>" . $row["food_quantity"] . "</td>";
    echo "<td>" . $row["expiry_date"] . "</td>";
    echo "<td>" . $row["original_price"] . "</td>";
    echo "<td>" . $row["discounted_price"] . "</td>";
}

```

```

        echo "<td>" . $row["food_details"] . "</td>";
        echo '<td><span class="edit-icon" onclick="openEditPopup(' . $id . ', "' . $name . '",',
        $company . ', "' . $category . '", "' . $quantity . '", "' . $date . '", "' . $oprice . '",',
        $dprice . ', "' . $details . '")">&#9998;</span></td>';
        echo '<td><span class="delete-icon"><a href="deletefood.php?fid=' . $id . '',
        '">&#10006;</span></td>';
        echo "</tr>";
    }
?
</tbody>
</table>
<?php
} else {
    echo "No data found";
}
$conn->close();
?>
<div class="overlay" id="overlayAdd" onclick="closePopup()"></div>
<div class="popup" id="popupAdd">
    <span class="close-btn" onclick="closePopup()">&times;</span>
    <h2>Add New Item</h2>
    <form id="requestForm" method="post" action="insert5.php"
    enctype="multipart/form-data">
        <label for="product">Product Name:</label>
        <input type="text" id="product" name="product" required>
        <label for="company">Company Name:</label>
        <input type="text" id="company" name="company" required>
        <label for="category">Category:</label>
        <input type="text" id="category" name="category" required>
        <label for="quantity">Quantity:</label>
        <input type="text" id="quantity" name="quantity" required>
        <label for="expiry">Expiry Date:</label>
        <input type="date" id="expiry" name="expiry" required>
        <label for="price">Original Price:</label>
        <input type="text" id="price" name="price" required>
        <label for="dprice">Discounted Price:</label>

```

```

<input type="text" id="dprice" name="dprice" required>
<br>
<label for="image">Choose Image:</label>
<input type="file" name="image" id="image" accept="image/*">
<label for="additionalInfo">Additional Information:</label>
<textarea id="additionalInfo" name="additionalInfo" rows="4"></textarea>
<button type="submit" class="add-button" name="submit">Submit</button>
</form>
</div>
<div class="overlay" id="overlayEdit" onclick="closeEditPopup()"></div>
<div class="popup" id="popupEdit">
<span class="close-btn" onclick="closeEditPopup()">&times;</span>
<h2>Edit Request</h2>
<form id="editRequestForm" method="post" action="foodedit.php"
      enctype="multipart/form-data">
<label for="editname">Product Name:</label>
<input type="text" id="editname" name="editname" required>
<label for="editcompany">Company Name:</label>
<input type="text" id="editcompany" name="editcompany" required>
<label for="editcategory">Category:</label>
<input type="text" id="editcategory" name="editcategory" required>
<label for="editquantity">Quantity:</label>
<input type="text" id="editquantity" name="editquantity" required>
<label for="editdate">Expiry Date:</label>
<input type="text" id="editdate" name="editdate" required>
<label for="editoprice">Original Price:</label>
<input type="text" id="editoprice" name="editoprice" required>
<label for="editdprice">Discounted Price:</label>
<input type="text" id="editdprice" name="editdprice" required>
<br>
<label for="editAdditionalInfo">Additional Information:</label>
<textarea id="editAdditionalInfo" name="editAdditionalInfo" rows="4"></textarea>
<input type="hidden" id="food_id" name="food_id" >
<button type="submit" class="add-button" name="submit">Submit</button>
</form>
</div>

```

```

<script>

    function openEditPopup(foodId,
        name,company,category,quantity,date,oprice,dprice,details) {
        document.getElementById('overlayEdit').style.display = 'block';
        document.getElementById('popupEdit').style.display = 'block';
        document.getElementById('food_id').value = foodId;
        document.getElementById('editname').value = name;
        document.getElementById('editcompany').value = company;
        document.getElementById('editcategory').value = category;
        document.getElementById('editquantity').value = quantity;
        document.getElementById('editdate').value = date;
        document.getElementById('editoprice').value = oprice;
        document.getElementById('editdprice').value = dprice;
        document.getElementById('editAdditionalInfo').value = details;
    }

    function closeEditPopup() {
        document.getElementById('overlayEdit').style.display = 'none';
        document.getElementById('popupEdit').style.display = 'none';
    }

    function openPopup() {
        document.getElementById('overlayAdd').style.display = 'block';
        document.getElementById('popupAdd').style.display = 'block';
    }

    function closePopup() {
        document.getElementById('overlayAdd').style.display = 'none';
        document.getElementById('popupAdd').style.display = 'none';
        document.getElementById('popupEdit').style.display = 'none';
    }

</script>
insert5.php
<?php
session_start();
$vendorid = $_SESSION['vendorid'];
if (isset($_POST['submit'])) {
    $product = $_POST['product'];
    $comapny = $_POST['company'];

```

```

$category = $_POST['category'];
$quantity = $_POST['quantity'];
$expiry = $_POST['expiry'];
$price = $_POST['price'];
$dprice = $_POST['dprice'];
$additionalInfo = $_POST['additionalInfo'];
$targetDir = "uploads/";
$targetFile = $targetDir . basename($_FILES["image"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($targetFile, PATHINFO_EXTENSION));
if ($uploadOk == 1) {
    if (move_uploaded_file($_FILES["image"]["tmp_name"], $targetFile)) {
        $conn = new mysqli("localhost", "root", "", "project");
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }
        $sql = "INSERT INTO food (vendor_id, food_name, food_company, food_category,
food_quantity, expiry_date, original_price, discounted_price, food_details,image)
VALUES ('$vendorid', '$product', '$comapny', '$category', '$quantity', '$expiry',
$price, $dprice, '$additionalInfo','$targetFile')";
        if ($conn->query($sql) === TRUE) {
            echo "New record created successfully";
            header("Location: vendorwelcome.php");
        } else {
            echo "Error: " . $sql . "<br>" . $conn->error;
        }
    }
}
$conn->close();
}
?>

```