

## **Project Proposal: Community Food Sharing Platform**

Tshwetso, Mokgatlhe, and 221411

Open Window, School of Fundamentals

DV200

Tsungai Katsuro

September 2024

## Table of Contents

List of Figures.....	3
Heading in Title Case.....	4
References.....	5

Use Google Docs' Headings to structure your assignment and refresh this ToC to update it automatically. Remember to set the line spacing back to double after refreshing the ToC. DELETE RED TEXT PRIOR TO SUBMISSION.

### List of Figures

**Figure 1** *Example of eidetic photomontage of a cultural harbour project in Odens* ..... **5**

**Use tabs to separate the figure number, title, and page number. If the document does not contain any images you may delete this page. DELETE RED TEXT PRIOR TO SUBMISSION.**

# Project Proposal: Community Food Sharing Platform

## Problem Statement

### Identifying the Issue:

Food waste is a pervasive global problem. Each year, substantial quantities of edible food are discarded by households, restaurants, and supermarkets. Concurrently, many individuals and families grapple with food insecurity, lacking reliable access to sufficient nutritious food. This disparity underscores a critical societal issue where available resources are underutilised, and needs are unmet.

### Significance and Solution:

The proposed Community Food Sharing Platform aims to mitigate food waste and alleviate hunger by connecting those with surplus food to those in need. By developing an accessible web application where individuals and businesses can list surplus food items for others to claim, we create a community-driven solution that promotes sustainability and social responsibility.

---

## Target Audience

### Primary Users:

1. **Donors:** Households, restaurants, grocery stores, and farmers willing to share surplus food.
2. **Recipients:** Individuals and families experiencing food insecurity.
3. **Volunteers:** Community members who can assist in food distribution or platform moderation.

**Benefits to Users:**

- Reduce food waste and contribute positively to the community.
  - Gain access to fresh, edible food, enhancing food security.
  - Engage in meaningful community service, fostering social cohesion.
- 

**Technology Stack**

**Chosen Stack: LAMP (Linux, Apache, MySQL, PHP)**

**Justification:**

- **Proven Reliability:** The LAMP stack is a time-tested, robust platform for web application development.
- **Cost-Effectiveness:** Open-source components reduce licensing costs and allow for budget-friendly development.
- **Wide Support and Resources:** Extensive community support facilitates troubleshooting and development.
- **Compatibility:** Ideal for developing dynamic web applications with strong database integration.

**Component Utilization:**

- **Linux:** The operating system that provides a stable and secure environment for the application.
- **Apache:** Serves as the web server to handle HTTP requests and deliver content to users.
- **MySQL:** Manages the relational database for storing user data, food listings, requests, and transaction history.

- **PHP:** The server-side scripting language to build the application's logic, handle form submissions, and interact with the database.

## Application Features

### 1. User Registration and Authentication:

- **Functionality:** Users can create accounts as donors, recipients, or volunteers.
- **CRUD Operations:** Create accounts, read user profiles, update personal information, and delete accounts if necessary.

### 2. Food Listing Creation:

- **Functionality:** Donors can post available food items with descriptions and images.
- **CRUD Operations:** Create new listings, read existing listings, update listing details, and delete listings when items are no longer available.

### 3. Search and Filter Functionality:

- **Functionality:** Recipients can search for available food based on location, type, or expiration date.
- **CRUD Operations:** Read operations to view and filter food listings.

### 4. Request and Claim Items:

- **Functionality:** Recipients can request or claim food items, and donors can manage these requests.
- **CRUD Operations:** Create requests, read request statuses, update request details, and delete requests if canceled.

### 5. Messaging System:

- **Functionality:** Enables communication between donors and recipients to coordinate pickups.

- **CRUD Operations:** Create messages, read conversation threads, and delete messages if needed.

#### 6. Rating and Feedback:

- **Functionality:** Users can rate their experiences and provide feedback to promote trust.
  - **CRUD Operations:** Create feedback entries, read ratings, update feedback if necessary, and delete inappropriate comments.
- 

## Database Design

### Tables and Relationships:

#### 1. Users Table:

- Fields: `user_id` (PK), `name`, `email`, `password_hash`, `address`, `user_type` (donor, recipient, volunteer), `rating`.

#### 2. Food\_Listings Table:

- Fields: `listing_id` (PK), `donor_id` (FK to Users), `item_description`, `quantity`, `expiration_date`, `pickup_location`, `image_path`, `status`.

#### 3. Requests Table:

- Fields: `request_id` (PK), `listing_id` (FK to Food\_Listings), `recipient_id` (FK to Users), `request_date`, `status`.

#### 4. Messages Table:

- Fields: `message_id` (PK), `sender_id` (FK to Users), `receiver_id` (FK to Users), `content`, `timestamp`.

#### 5. Feedback Table:

- Fields: `feedback_id` (PK), `transaction_id` (FK to Requests), `giver_id` (FK to Users), `receiver_id` (FK to Users), `rating`, `comments`.

### Normalization and Data Modeling Considerations:

- **Normalization:** Apply normalization up to the third normal form (3NF) to eliminate data redundancy and ensure data integrity.
  - **Foreign Keys and Constraints:** Establish relationships between tables using foreign keys to maintain referential integrity.
  - **Indexing:** Implement indexing on frequently searched fields like `user_id` and `listing_id` to improve query performance.
- 

## User Interface and Experience

### Intended UI/UX:

- **Responsive Design:** Utilize HTML5 and CSS3 to ensure the application is accessible on various devices, including desktops, tablets, and smartphones.
- **Intuitive Navigation:** Organize content logically with clear menus, buttons, and icons to guide users seamlessly through the application.
- **Visual Appeal:** Employ a clean and modern design aesthetic with a focus on usability and accessibility.

### Catering to Target Audience:

- **Accessibility Features:** Include options for larger text sizes, high-contrast themes, and screen reader compatibility.
- **Localization:** Provide multi-language support to cater to diverse user demographics.



- **User-Friendly Forms:** Simplify data entry with clear labels, placeholders, and validation messages to reduce user errors.
- 

## Security Considerations

### Potential Risks:

- **Unauthorized Access:** Risk of unauthorized users gaining access to sensitive data.
- **SQL Injection Attacks:** Malicious input could compromise the database.
- **Data Privacy:** Exposure of personal information such as addresses and contact details.
- **Cross-Site Scripting (XSS):** Injection of malicious scripts into web pages viewed by other users.

### Mitigation Strategies:

- **Input Validation and Sanitization:**
  - Validate all user inputs on both client-side (JavaScript) and server-side (PHP).
  - Use prepared statements with parameterized queries to prevent SQL injection.
- **Authentication and Authorization:**
  - Implement secure password hashing using algorithms like bcrypt.
  - Enforce role-based access control to restrict functionalities based on user roles.
- **Secure Communication:**
  - Use HTTPS to encrypt data transmitted between the client and server.
  - Obtain and configure SSL/TLS certificates on the server.
- **Session Management:**

- Utilize secure session handling practices, including session timeouts and regeneration of session IDs upon login.
  - **Data Protection:**
    - Store minimal personal data necessary for application functionality.
    - Comply with data protection regulations such as GDPR if applicable.
  - **Regular Security Audits:**
    - Conduct periodic code reviews and vulnerability assessments.
    - Keep the software stack updated with the latest security patches.
- 

## Project Timeline

### Phase 1: Planning and Requirement Analysis (Weeks 1-2)

- **Tasks:**
  - Define detailed project requirements.
  - Identify key features and functionalities.
  - Establish project scope and objectives.
- **Deliverables:**
  - Project requirements document.
  - Initial project plan and schedule.

### Phase 2: Design (Weeks 3-4)

- **Tasks:**
  - Design the database schema using ER diagrams.
  - Create wireframes and mockups for the user interface.
  - Plan the application architecture and flow.
- **Deliverables:**

- Finalized database design.
- UI/UX design prototypes.
- System architecture document.

### **Phase 3: Development (Weeks 5-8)**

- **Tasks:**

- Set up the development environment on a Linux server.
- Develop the front-end using HTML, CSS, and JavaScript.
- Implement server-side logic with PHP.
- Integrate the application with the MySQL database.
- Configure the Apache web server to serve the application.

- **Deliverables:**

- Functional modules for user registration, food listings, and messaging.
- Integrated and operational application components.

### **Phase 4: Testing (Weeks 7-8)**

- **Tasks:**

- Conduct unit testing on individual components.
- Perform integration testing to ensure modules work together seamlessly.
- Execute user acceptance testing with a group of beta users.
- Identify and fix bugs or performance issues.

- **Deliverables:**

- Test plans and test cases.
- Bug reports and resolutions.
- Tested and refined application ready for deployment.

### **Phase 5: Deployment (Week 7-8)**

- **Tasks:**

- Deploy the application on a production Linux server.
- Optimize server settings for performance and security.
- Set up backup and recovery procedures.
- Prepare deployment documentation.
- **Deliverables:**
  - Live application accessible to users.
  - Deployment and maintenance documentation.

### **Milestones and Expected Deliverables:**

- **Week 2:** Approval of project requirements and plan.
  - **Week 4:** Completion of design phase with approved UI/UX prototypes.
  - **Week 6:** Completion of development phase with all features implemented.
  - **Week 7-8:** Completion of testing phase with all issues resolved.
  - **Week 8:** Successful deployment of the application.
- 

## **Challenges and Risks**

### **Potential Challenges:**

- **Technical Integration:** Ensuring seamless interaction between PHP, MySQL, and Apache.
- **Security Vulnerabilities:** Protecting against common web application attacks.
- **User Engagement:** Attracting and retaining users on the platform.

### **Strategies to Overcome Challenges:**

- **Technical Expertise:**
  - Allocate time to familiarize myself with the latest PHP frameworks and best practices.

- Utilize MVC (Model-View-Controller) frameworks like Laravel for better code organization and efficiency.
  - **Enhanced Security Measures:**
    - Implement security best practices from the onset of development.
    - Regularly update all components of the LAMP stack to their latest stable versions.
  - **Marketing and Community Outreach:**
    - Partner with local organizations and community leaders to promote the platform.
    - Use social media and local events to raise awareness and encourage participation.
- 

## Conclusion

NeighbourGoods. The Community Food Sharing Platform seeks to create a meaningful impact by addressing the dual challenges of food waste and hunger. Leveraging the reliable and robust LAMP stack, the application will provide a secure, user-friendly platform that connects donors with surplus food to recipients in need. By fostering community engagement and promoting sustainable practices, this project has the potential to make a lasting positive contribution to society.

---

### Expected Impact:

- **Reduction in Food Waste:** Decrease the amount of edible food being discarded.
- **Improved Food Security:** Provide reliable access to food for individuals and families facing hunger.

- **Community Empowerment:** Strengthen community bonds through shared resources and mutual support.

**Project Significance:**

This platform not only addresses immediate needs but also encourages a culture of sharing and environmental stewardship. By facilitating efficient redistribution of resources, the project contributes to social equity and sustainability, aligning with broader goals of community development and welfare.

## References