Food Donation system proposal

Contents

1. Problem Statement

Problem Definition: Food waste is a major global issue, with millions of tons of food being thrown away each year, while millions of people go hungry. In many communities, there is a disconnect between food donors (e.g., restaurants, grocery stores, and individuals) and organisations like shelters and food banks that need food to support those in need. This imbalance results in unnecessary waste and fails to address food insecurity effectively.

How the Application Addresses the Problem? This web application will serve as a platform where businesses and individuals can list available food donations, while food banks and shelters can easily browse, claim, and collect these resources. By bridging the gap between food donors and recipients, the platform will enable quicker and more efficient transfers of surplus food, helping to reduce waste and fight hunger in local communities.

2. Target Audience

Primary Users:

- **Donors**: Restaurants, grocery stores, caterers, and individuals with surplus food.
- **Recipients**: Food banks, shelters, and community organizations that distribute food to those in need.
- **Admin**: Individuals who help manage food logistics.

Benefits:

- **Donors**: Will have a simple platform to list waste food and contribute to the community, reducing waste and potentially receiving tax benefits.
- **Recipients**: Will have access to available food donations, allowing them to claim food that matches their needs.
- **Admin**: Will be able to manage food collections, deliveries, and other logistical needs efficiently, ensuring a seamless distribution process.

3. Technology Stack

Chosen Stack: LAMP

Justification:

- The **LAMP stack** is a widely-used and reliable solution for building web applications. It offers a mature ecosystem, is cost-effective, and provides high scalability.

Overview of Components:

- **Linux**: The operating system that will host the web application, providing stability and security.
- **Apache**: The web server that will handle requests and serve web pages to users.
- **MySQL**: The database system for storing user profiles, food donations, and transaction histories.
- **PHP**: The server-side scripting language for processing user inputs, interacting with the database, and generating dynamic web content.

4. Application Features

Core Features:

1. **Food Donation Listing**:
    - Donors will create listings of available food, including details like quantity, type, expiry date, and pickup location.
    - **CRUD Operations**:
        - **Create**: Donors submit new food listings.
        - **Read**: Recipients can view available listings by category or location.
        - **Update**: Donors can edit listing details (e.g., if food quantities change).
        - **Delete**: Listings are removed once the food is claimed or no longer available.
2. **Claiming Donations**:
    - Recipients will be able to claim available food donations, and they will coordinate pickup times with donors.
    - **CRUD Operations**:
        - **Create**: Recipients claim a donation by creating a request.
        - **Read**: Donors can see which organisations have claimed their listings.
        - **Update**: Recipients can modify pickup details (e.g., time, method).

        - **Delete**: Remove requests after food is collected.

5. Database Design

Key Tables:

1. **Users**: Stores information about donors, recipients (UserID, Name, Role, ContactInfo, etc.).
2. **Donations**: Tracks food donations (DonationID, DonorID, FoodType, Quantity, ExpiryDate, Location, Status).
3. **Claims**: Links recipients with donations (ClaimID, DonationID, RecipientID, PickupDate, PickupStatus).
4. **Volunteer:** VolunteerID, AssignedDonationID, Status (e.g., In-Transit, Completed), PickupLocation, DropoffLocation, AssignmentTime

Normalisation:

- The database will be normalised to third normal form (3NF) to eliminate redundancy and ensure data integrity. For instance, the relationship between donations and claims is a one-to-many relationship, with each donation potentially being claimed by multiple recipients.

6. User Interface and Experience

Intended UI:

- The application will feature a clean, intuitive interface with a user-friendly dashboard for each user type (donors, recipients, volunteers). Key features include:
    - **Donor Dashboard**: Easily create and manage food donation listings.
    - **Recipient Dashboard**: Browse and claim donations, with filters for location, food type, or expiration date.

User Experience (UX):

- **Simplicity**: The application will be designed with ease of use in mind, especially for non-technical users like small food banks and local restaurants.
- **Accessibility**: It will be optimised for mobile use to ensure that users can access the platform on the go, which is critical for quick food donations and claims.

7. Security Considerations

Potential Risks:

1. **Data Breaches**: The system will handle sensitive data, such as user contact information and donation details.
2. **Unauthorised Access**: Ensuring that only verified users (e.g., food banks, volunteers) can access certain features.
3. **Input Validation**: The risk of injection attacks through improper input handling.

Mitigation Strategies:

- **Encryption**: All sensitive data (e.g., user credentials, contact information) will be encrypted using SSL/TLS.
- **Authentication and Authorization**: The system will use secure login mechanisms (e.g., two-factor authentication) and role-based access control to restrict permissions based on user roles.
- **Input Validation**: Form data will be sanitized and validated on both the client and server side to prevent SQL injection and cross-site scripting (XSS) attacks.

8. Project Timeline

Planning (1 week):

- Define project requirements, finalise database schema, and create wireframes for the user interface.

Development (6 weeks):

- **Weeks 1-2**: Develop core features (food donations, user management).
- **Weeks 3-4**: Implement claiming process and volunteer management.
- **Weeks 5-6**: Design the user interface and integrate security features.

Testing (1 week)

- Need to refractor and improve upon code so that it is easier to understand and more efficient.

9. Challenges and Risks

Potential Challenges:

- **Data Accuracy**: Ensuring that food availability and pickup times are accurately maintained.
- **Scalability**: As more users join, database performance and system responsiveness may slow down.

Mitigation Strategies:

- **Data Accuracy**: Implement automated reminders for users to confirm donation statuses.
- **Scalability**: Use database indexing and load balancing to optimise performance as the user base grows.

10. Conclusion

The food donation web application will serve as a powerful tool to reduce food waste and address hunger in local communities. By connecting donors, recipients, and volunteers on a single platform, the app will facilitate efficient food distribution while minimising waste. The project's impact will be immediate and long-lasting.