

# Ram Logic | Web Application for Voiland Food Pantry

---

## *Project Report*



### SPONSORS

*Washington State University*  
*Frank Innovation Zone*

### PREPARED FOR

Maynard Siev

### PREPARED BY

Cristobal Escobar  
Alan Sun



# I. PROJECT DESCRIPTION

---

## I.1 Introduction

The Voiland Food Pantry & Wellness Center serves as a critical resource for Washington State University students, providing access to food, hygiene products, and wellness support. With increasing demand and a growing variety of inventory, maintaining accurate records of pantry usage, tracking inventory levels, and collecting meaningful data has become a key operational challenge. Manual data entry processes are prone to errors and can make it difficult to quickly generate reports, limiting the pantry's ability to analyze trends and make informed decisions regarding procurement and volunteer scheduling.

The Data Tracker project is intended to address these challenges by designing and implementing an integrated system that collects, stores, and visualizes pantry data in a reliable and efficient manner. The system will include a database that records client check-ins, item distribution, and volunteer activity. Hardware peripherals such as a barcode scanner or card reader will be integrated to streamline check-in and inventory management processes, reducing manual input and improving accuracy.

Accurate and timely data collection is vital for the pantry's mission. Usage statistics can inform decisions about stocking patterns, predict surges in demand (e.g., during midterms or finals), and provide justification for continued funding and expansion. By presenting data in a centralized, easy-to-use interface through a WordPress-based web application, the system aims to empower pantry staff and volunteers to monitor operations in real time and generate reports for stakeholders.

This project is positioned within the domain of information systems for community services, combining database design, hardware integration, and web development to improve service delivery. The database team, newly assembled for this project, is responsible for developing the system from the ground up, beginning with the creation of an Entity-Relationship diagram to define key data relationships. The resulting platform will not only support the day-to-day operation of the pantry but also contribute to its long-term sustainability and impact on student well-being.

## I.2 Background and Related Work

The pantry's current system is based on paper lists and spreadsheets, as well as disjointed digital methods, to organize pantry inventory and volunteer sign-ups. These methods are susceptible to human error, not very scalable, and lack analytics for data-driven decision-making. Food pantry programs usually require dedicated software systems that handle either inventory management or volunteer coordination but integrated solutions that manage both functions comprehensively remain scarce. Our front-end system will be developed on WordPress because it supports established web technologies and allows us to build a customized solution that meets the pantry's requirements.

The system to be constructed will operate together with the pantry's newly acquired and function-tested hardware components comprising a card reader and barcode scanner. These devices will support both inventory tracking and client registration processes for the pantry.

## I.3 Project Overview

The core Food Pantry Data Tracker project is underway, with key specifications defined and an initial Entity-Relationship diagram created for the database. The team has selected WordPress as the platform for the web application and confirmed that hardware peripherals (barcode scanner and card reader) are available and functional. Key development areas include the following:

1. Database implementation and integration with WordPress
2. Front-end interface design with client-focused default view and volunteer sign-in option
3. Hardware integration for automated check-in and inventory management
4. Real-time data visualization and reporting capabilities
5. User interface refinements for accessibility and ease of use
6. Development of documentation, tutorials, and user guidance

By completing these tasks, this project aims to improve the accuracy, efficiency, and reliability of pantry operations, enabling staff to make data-driven decisions and better serve the student community.

## I.4 Client and Stakeholder Identification

The project sponsor is the Voiland Food Pantry & Wellness Center represented by Maynard Siev. A data management system consisting of both software and hardware

components is needed by the pantry to manage food distribution operations alongside volunteer scheduling and client support functions.

The pantry is a part of the larger operation of nonprofit and community needs resource management. Food banks, resource pantries, wellness centers, and similar entities typically have similar needs to track their services, volunteers, and client interactions. By having a solution that meets the specified use cases of barcode scanning, role-based interfaces, and database-powered reporting, this system can be applied to larger or similar efforts within the community.

Pantry customers, volunteers, and administrators are stakeholders that depend on the resource availability and can help guide its use. The use cases of inventory management, volunteer availability, client-facing availability, and recent hardware interface should provide a platform that is scalable, reliable, and easy to use in a broader context of nonprofits and similar use cases.

## II. TEAM MEMBERS & BIOS

---

Include an entry in a narrative form for each of your team members. The goal is to demonstrate the team's skills and project coverage. This is not just a pasted in resume, but a summary of your involvement in the project, and your technical interests. Feel free to lift from your Team Inventory to include:

- Name
- Degree plan
- Project role - which aspects you're responsible for
- General areas of experience and technical interests

Example:

Aaron Crandall is a computer science student interested in artificial intelligence, satellite development, and clock making. His prior projects have included smart homes, radio controlled dirigibles, and programming clocks. Aaron's skills include C/C++, Python, RabbitMQ, Genetic Algorithms, and delinting. For this project, his responsibilities include developing the Gamma Module, leading user experience feedback, and delivering sandwiches.

## III. REQUIREMENTS AND SPECIFICATIONS

---

### III.1 Introduction

This section outlines the data specifications for the Voiland Food Pantry's backend database system which manages inventory levels along with student and volunteer information.

A local, lightweight Flask front-end has been developed in order to test database functionality and ensure proper data validation and constraint enforcement before being exposed to the partner team's WordPress-based website.

The database design should preserve data organization and accuracy while positioning it for easy future import into the website environment.

The resulting platform will include a database to record client check-ins and item distribution, with integration of hardware peripherals (a Cougar Card reader and barcode scanner) to streamline check-in and inventory management. Accurate data collection is vital, as usage statistics will inform decisions on stocking patterns, predict demand, and provide justification for continued funding and expansion of the pantry's mission. The web application will be accessible via a WordPress domain, with a centralized, easy-to-use interface for staff and volunteers.

## III.2 System Requirements Specification

### III.2.1 Use Cases

Student Check-in (Client):

Pre-condition	Application running, on the clientele-facing page.
Post-condition	Pantry usage recorded (StudentId, VisitDate, TotalItems).
Basic Path	User scans Cougar Card. <b>OR</b> User uses the backup digital form sign-in.
Related Requirements	Student must be in the Students table. Requires hardware integration.

Volunteer/Admin Sign-in:

Pre-condition	Application running, on the home page.
Post-condition	User directed to their respective role-based interface.
Basic Path	User selects a separate sign-in option for Volunteers/Admins.

Related Requirements	Separate data models must exist for different user types.
----------------------	---

View Inventory:

Pre-condition	Application connected to the database.
Post-condition	Display of all items, including Name, Category, Quantity, and Expiration Date.
Basic Path	User navigates to the Inventory view.
Related Requirements	Must retrieve data from the Items table.

Add Inventory Item:

Pre-condition	Application connected to the database.
Post-condition	New item recorded in the Items table.
Basic Path	User navigates to Add Item page. Enters Name, Category, Quantity, Expiration Date, and optional Barcode. Clicks "ADD ITEM".
Related Requirements	All fields except Barcode are required.

Edit Inventory Item:

Pre-condition	Inventory item exists and is displayed.
Post-condition	Item details (Name, Category, Quantity, Exp Date) updated in the database.
Basic Path	User clicks 'Edit' for an item. Enters new parameters and

	clicks 'Save'.
Related Requirements	Must be able to retrieve the item by its ItemId.

#### Delete Inventory Item:

Pre-condition	Inventory item exists and is displayed.
Post-condition	Item removed from the Items table.
Basic Path	User clicks 'Delete' for an item. Confirms the deletion.
Related Requirements	User must have Admin/Volunteer privileges.

#### Manage Volunteer Shifts:

Pre-condition	Application running, connected to the database.
Post-condition	Shifts are recorded in the Shifts table.
Basic Path	Admin/Volunteer inputs a Shift Date, Start Time, and End Time.
Related Requirements	Must support time and date data types.

#### Generate Expiration Report:

Pre-condition	Application connected to the database.
Post-condition	A report or list of items nearing their ExpDate is displayed.
Basic Path	Admin/Staff requests the Expiration Report.

Related Requirements	Must support reporting functionality.
----------------------	---------------------------------------

Record Items Taken:

Pre-condition	Student check-in/usage has been recorded (Usageld).
Post-condition	ItemsTaken table is populated, linking Usageld to ItemId and QuantityTaken.
Basic Path	Volunteer/Staff records the items and quantities distributed during the visit.
Related Requirements	Must deduct quantity from the Items table.

## III.2.2 Functional Requirements

### III.2.2.1 Inventory Item Management

Description	The system should store and manage information about all types of food pantry items, including name, category, quantity, expiration dates, and barcode.
Source	Requested from our client Maynard
Priority	0

### III.2.2.2 Auto-Generated Identifiers

Description	Each table in our database should have a unique auto-increment primary key to maintain data integrity and simplify relationships between tables.
Source	Suggested by our team and approved by our client.
Priority	0

#### III.2.2.3 Student Visit Logging

Description	The system should record the student's food pantry visits, it should store student ID, name, major and total numbers of items taken each visit
Source	Requested by our client Maynard.
Priority	0

#### III.2.2.4 Volunteers Hours Tracking

Description	The database should allow Volunteers and Shifts tables to be linked through the VolunteerHours and record the total hours worked by a volunteer with the corresponding shift.
Source	Requested by our client Maynard.
Priority	1

#### III.2.2.5 Referential Integrity and Validation

Description	The system should enforce data consistency using primary and foreign key constraints. Input validation needs to ensure proper data types.
Source	Suggested by our team and approved by our client.
Priority	0

#### III.2.2.6 Data Export for Website Integration

Description	The database should support exporting pantry data (items, students, volunteers, visits) into a MySQL- or CSV-compatible format for use by the WordPress website team.
Source	Suggested by our team and approved by our client.
Priority	0

#### III.2.2.7 Flask Testing Interface

Description	A simple Flask web application should connect to the local MySQL database to perform CRUD operations for testing and demonstration purposes before integration with WordPress.
Source	Suggested by our instructor and approved by our client.

Priority	0
----------	---

### III.2.2 Non-Functional Requirements

#### Compatibility

- The database must be available and integrated with the front-end application being developed on a WordPress domain.
- The system must integrate with the client's existing, function-tested hardware (Cougar Card reader and barcode scanner).

#### Usability

- The default state of the application must be the clientele-facing page.
- The design must be a user-friendly system that motivates students to sign in and supports simple data entry.

#### Performance

- The simplicity of implementation should be a top concern, factoring in budget constraints.

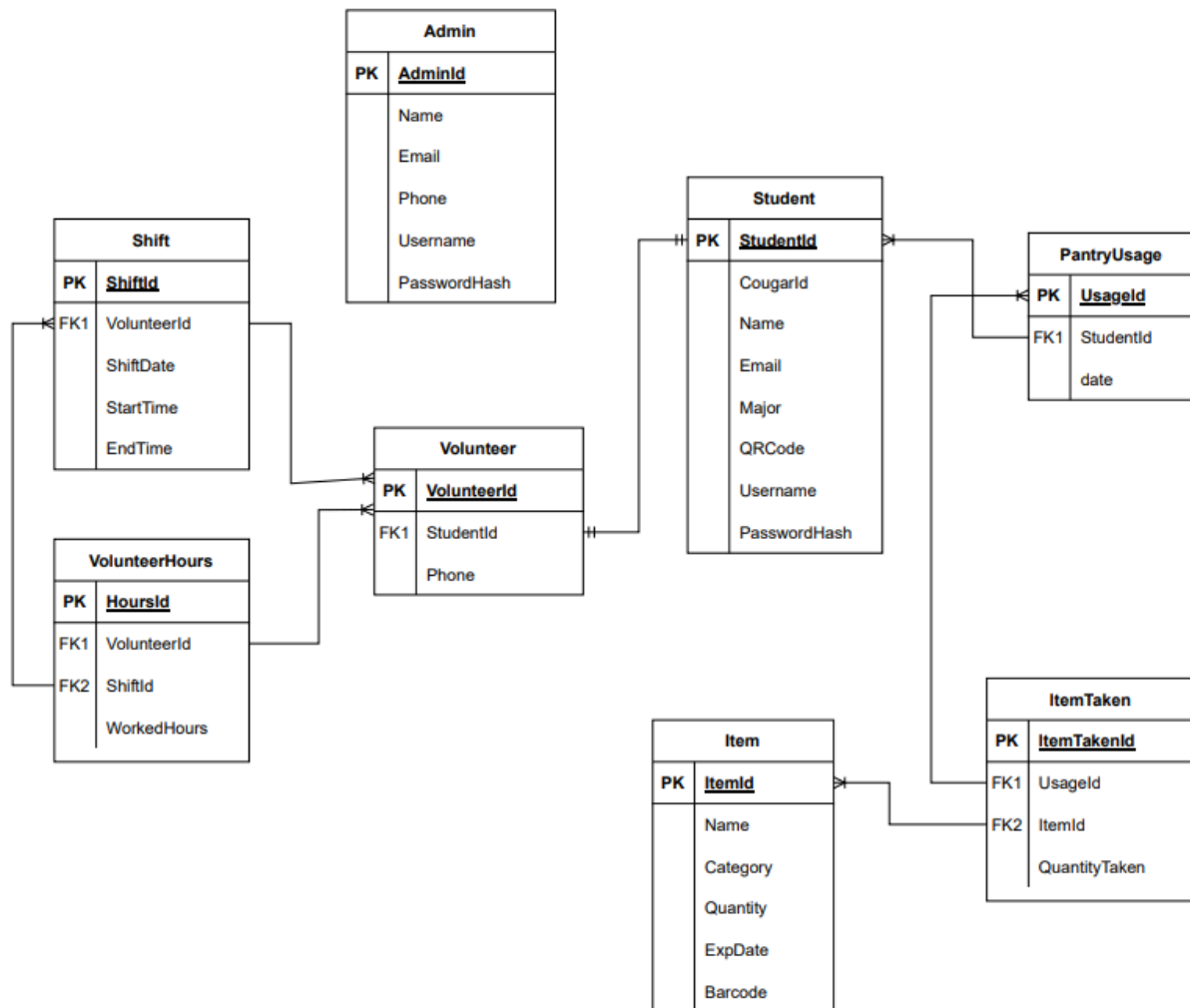
#### Scalability

- The database must be flexible and easy to expand to multiple departments if desired.
- The system should be applicable to larger or similar efforts within the context of non-profits and community needs resource management.

#### Maintainability

- The system must be able to support a workflow for syncing progress with the website team.

### III.2.3 Entity and Relationship Requirements



The Entity–Relationship (ER) diagram defines the logical structure of the Voiland Food Pantry database and serves as the foundation for the implemented schema. The core entities implemented were Items, Students, Volunteers, PantryUsage, Shifts, and VolunteerHours. The relationships between these entities were added to allow for functionality for all of the major system operations. These entities were chosen because they correspond to physical items and operations discussed and identified in the meeting with the client. Additional entities can be easily added to the database as it is developed and tested further, such as tables for tracking donors and user authentication, for example. The finalized ERD will likely continue to be modified and extended as future testing and front end integration is done with WordPress and other components.

### III.3 System Evolution

This section documents the current assumptions and identified risks that will influence the project's development path.

#### Assumptions

- **Hardware Functionality:** The client-provided hardware (Cougar Card reader and barcode scanner) is functional, tested, and will be IT-approved shortly for integration.
- **Platform Integration:** The database solution can be successfully integrated and made available to the website team's **WordPress** front-end application.
- **Scope Division:** The division of work between the database team (responsible for the back-end) and the website team (responsible for the front-end) is clearly established.
- **Entity Relationships:** The final Entity-Relationship (ER) diagram accurately captures the required data relationships (inventory, usage, volunteer hours, and shift scheduling).

#### Anticipated Changes and Risks

Risk/Unfinished Work	Implication/Mitigation	Source
<b>Database Integration</b>	Finalizing the ER diagram, beginning SQL schema creation, and defining data exchange protocols are required to integrate with the 421 (website) team.	Incomplete Issues
<b>Coordination</b>	Improving coordination and establishing a workflow with the website team is necessary for smooth integration.	Needs Improvement
<b>Data Specification</b>	The team still needs detailed specifications	Needs Improvement

	from the client on exact reporting requirements and data fields for usage, inventory, and volunteer tracking.	
<b>Access Control</b>	Precise access permissions and restrictions for each user type need to be defined and documented.	Needs Improvement
<b>Expiry Alerts</b>	Logic for generating reports on soon-to-expire items needs to be implemented (e.g., automated expiry alerts or reporting scripts).	Needs Improvement

# GLOSSARY

---

## **Admin**

An Administrator user with full access and privileges to manage the entire system, including user roles, inventory, and reporting.

## **Barcode Scanner**

A hardware peripheral used to read item barcodes for streamlined inventory tracking and distribution recording.

## **Cougar Card Reader**

A hardware peripheral used to scan Washington State University student ID cards to facilitate client check-in and usage tracking.

## **Database Team**

The team responsible for designing, implementing, and maintaining the project's back-end data structure and logic (Ram Logic).

## **ER Diagram**

Entity-Relationship Diagram; a visual model of the database structure defining the relationships between data entities (tables).

## **Inventory**

The collection of food and hygiene products currently available and tracked by the Voiland Food Pantry.

## **Voiland Food Pantry**

The client organization and subject of the Data Tracker project (Voiland Food Pantry & Wellness Center).

## **WordPress**

The content management system (CMS) selected as the platform for the front-end web application.

## REFERENCES

---

**R.1** Client Meeting Agenda and Minutes (9/3/2025, 9/10/2025, 10/2/2025).

**R.2** Project Report: \*Ram Logic

**R.3** Initial Database Schema Definition (*Fiz Project/App/DB Files/schema.sql*).

**R.4** Sprint Report 5 (Future work and coordination items).

**R.5** Entity-Relationship Diagram Draft (*ER\_FoodPantry\_DiagramDraft2.drawio.pdf*).