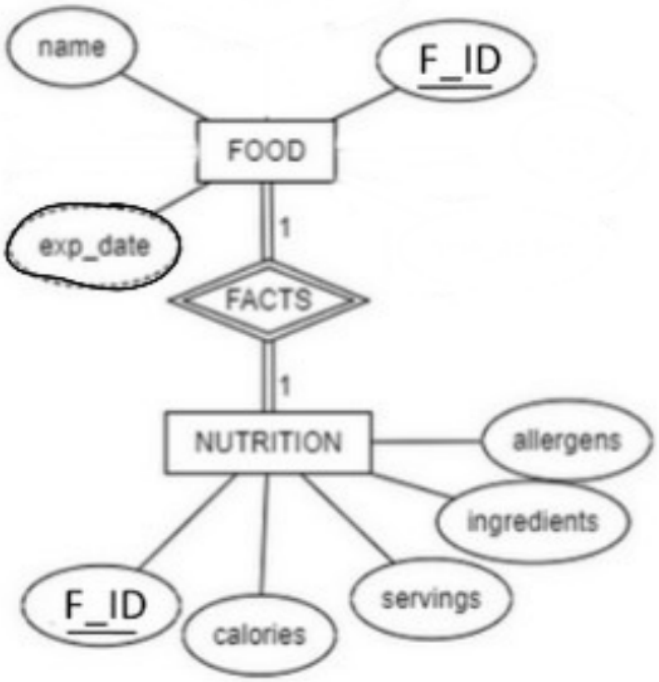
Group g Members :
Claire DeVlieger
Yannis Fu
Everett Yan

Problem Statement

Our project idea is a food expiration tracking web application. The software will allow users to log the food that they have into a ‘Pantry’ and check at any time which items are getting close to the expiration date. This will be tracked depending on the name and expiration date of the foods. Name, expiration date, and quantity information will be entered manually by the user into the input field. Information about stored items in the ‘Pantry’ will be available to view and sort based on the criteria that the user selects. For example, the user will be able to see a list of the foods organized based on their expiration date or ordered by when they were added.

The front-end aspect of our application displays a website where the user can interact with their stored foods. It opens on the ‘Home’ page which displays the title screen and a welcome message, as well as including a navigation bar where other pages can be reached. The ‘About’ page displays general information about the purpose and usage of Pantry. The ‘Foods’ page is where the user can interact with the database and make changes to the contents of their storage. This includes adding, removing, and sorting the contents of the Pantry. Any foods expiring within a short amount of time will be displayed on the ‘Home’ page, serving as a reminder to make use of those items quickly. This app will allow users to make better use of their food within a timely manner and reduce the amount of waste due to food expiring unexpectedly.

Conceptual Database Design (Revised)



Note: the original design met the specifications of Phase I, but during implementation had to be simplified due to time constraints.

Logical and Physical Database Design

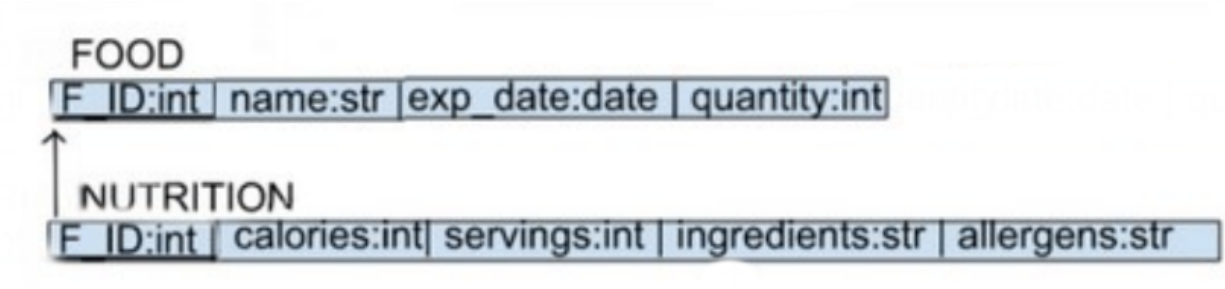


Table	Attribute	Type	Constraint
FOOD	F_ID	integer	Primary key
FOOD	name	string	NOT NULL
FOOD	exp_date	date	NOT NULL
FOOD	quantity	integer	
NUTRITION	F_ID	integer	Foreign key
NUTRITION	calories	integer	
NUTRITION	servings	integer	
NUTRITION	ingredients	string	
NUTRITION	allergens	string	

Note: the original designs met the specifications of Phase II, but during implementation had to be simplified due to time constraints.

Application Program Design (Revised)

Function 1: Create_Food

//This function has the user manually input information that will be used to create a FOOD tuple.
//It accesses the "FOOD" table.
Input: FOOD name, FOOD expiration date, FOOD quantity
Steps:

1. Creates a new food ID that is automatically incremented so that there are no duplicates.
2. Have the user fill in attributes about food, including the name, expiration date, and quantity.
3. Check if user inputted attribute values are valid. Reject if the inputted values are not in the domain or are invalid/NULL.
4. Insert the itinerary into the FOOD table.

Function 2: Food_and_Nutrition_List

//This function lists which FOOD items are closest to expiration, providing a list sorted in order of expiration date displayed on the main menu.
//It accesses the "FOOD" and "NUTRITION" tables.
Input: the current date.
Steps:

1. The function searches the FOOD table and determines the one with the closest expiration date value, then ranks each FOOD by comparing to this date.
2. This outputs the information about the FOOD's name, expiration date, and quantity, as well as the associated NUTRITION information.
3. If any of the date values are within a certain range, they are marked as nearing the expiration date.
4. Information about the FOODs are retrieved and displayed on the food menu in order of their ranking.

Function 3: Delete_Food

//This function allows users to delete food that they have previously added. This is typically done when the user has eaten/used/disposed of the food.
//It accesses the "FOOD" and "NUTRITION" tables.
Input: F_ID
Steps:

1. First check if the inputted F_ID exists. If not, reject the operation.
2. If the F_ID does exist, then the delete operation is executed. The FOOD tuple that corresponds to the F_ID input is removed from the FOOD table.

3. Additionally, the delete cascades as all other tuples that were related to the FOOD tuple. This means any NUTRITION tuples that shared a foreign key in the FOOD tuple are also deleted, as F_ID cannot be NULL in those tuples.
4. The corresponding FOOD tuple, now deleted, is also no longer displayed.

Function 4: Sort_Food
//This function allows a user to sort the foods in their Pantry based on different criteria.
// It accesses the "FOOD" table.
Input: name, expiration date, or amount
Steps:

1. Receives user input for the criteria to sort by - expiration date, name, or amount of food.
2. Returns tuples organized in order of what was selected.
3. Displays the associated tuples.

Function 5: Get_Nutrition
//This function gets nutrition information from a public food database.
// It accesses the "FOOD" and "NUTRITION" tables.
Input: FOOD name
Steps:

1. Checks the information of the FOOD.
2. References the database to find the nutrition details.
3. Inputs the found information into the NUTRITION table, linking to the FOOD's ID using a foreign key approach.

Note: the original functions met the specifications of Phase II, but during implementation had to be simplified due to time constraints.

Installation

Intended operating system:

- Windows

Steps to install on another computer:

1. Copy the HTTPS URL from the git repository, https://github.com/ezybg7/pantrywebsite
2. Open your system's default command prompt.
3. Clone the repository using 'git clone' (if git is not installed use https://gitforwindows.org/)
4. Navigate into the repository using 'cd pantrywebsite'.
5. Navigate into the client folder with 'cd client'.
6. Install NPM using 'npm install -g npm'.
7. Install react using 'npm install react-scripts --save'.
8. Install Next.js using 'npm install react react-dom next'.
9. Install Axios using 'npm install axios'.
10. Open the website using 'npm run dev'.
11. This should open a browser window with the web application running. If it does not, copy the URL displayed in the command prompt window into a browser tab.
12. To activate the SQL server, enter the server folder using 'cd..' and 'cd server'.
13. Run the commands 'npm install cors dotenv express mysql2 nodemon' and 'npm run dev'.
14. The web application should now work properly.

Dependencies

- next
- react
- react-dom
- cors
- Mysql2
- Axios
- express

User Manual

Welcome to Pantry! We have included a user manual for your convenience here,

When you first start up the web application you will be taken to the pantry welcome Homepage. From this page users can navigate to the ‘About’ or ‘Food’ tabs, located on the upper navigation bar. After entering one of these pages, the home screen is accessible using the ‘Home’ tab.

Figure 1: Pantry Homepage

The ‘About’ page provides background info and statistics about the worldwide crisis of food waste, as well as a basic description of the functions of Pantry.

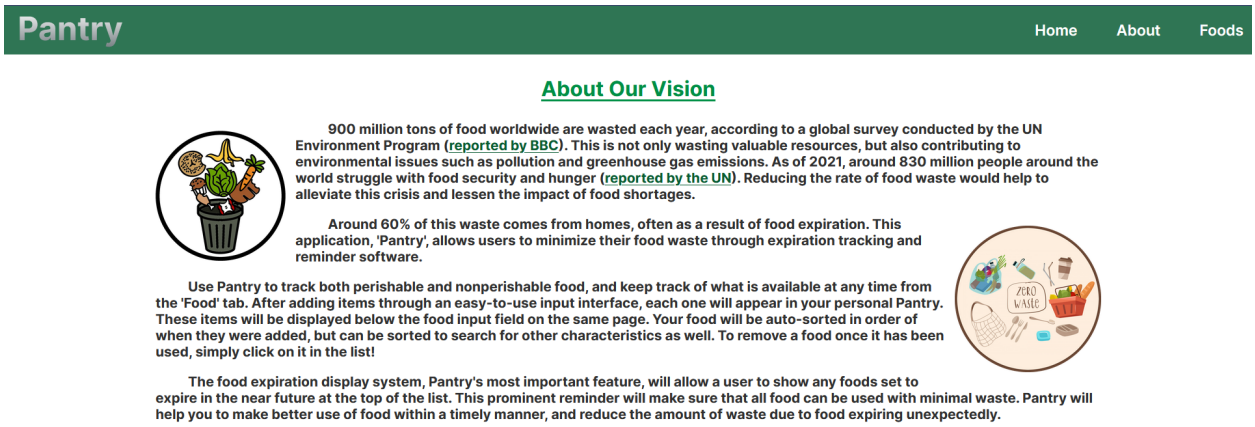


Figure 2: Pantry About page

The ‘Food’ page is the most important section of Pantry.
This page allows foods to be added to the Pantry database by entering information about the food name, expiration date, and quantity. Once the ‘Add food’ button is clicked, the new items should appear in order of when they were added.
The dropdown menu next to the button allows sorting based on different categories. Orgainzation is available for expiration date, food name, or amount of food.
To remove a food once it has been used, simply click on it in the list and it will be deleted!

Figure 3: Pantry Food page

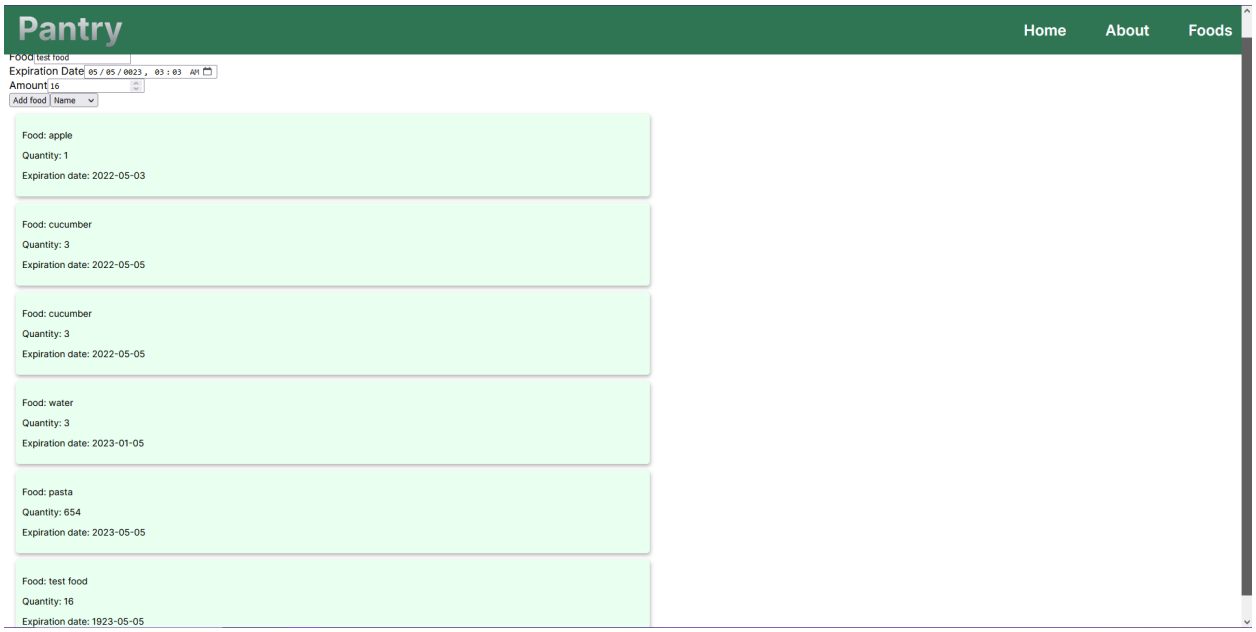


Figure 4: Example food added