Senior Capstone Project Requirements

Project Name

Household Menu Planner and Ingredient Tracker (HoMePIT)

Team Members

- Erik Anderson
- Jerome Chestnut
- Maxwell Fugette

Abstract

Menu planning is an inescapable and oft time-consuming part of human existence. Some individuals attempt to avoid it by subsisting on fast and highly processed food, while others try their best to manage it using whatever tools are available to organize and categorize this part of their lives. With healthy eating being the preferred option, it is imperative for it to be made as simple and convenient as possible. We believe that many of the tasks associated with meal planning, such as the tracking of household ingredients and foodstuffs, planning what meals to eat on what days, and the composition of shopping lists are often too tedious and time-consuming for the average person, which will inevitably lead to unhealthy dietary choices. All of this can be simplified with the appropriate application of technology. HoMePIT is our attempt at making this simplification.

HoMePIT, as its expansion suggests, will assist primarily in the tasks associated with planning a household's menu and tracking its supply, usage, and replacement of ingredients and other foodstuffs. It will be designed to streamline the culinary experience of a household by making ingredient inventorying, grocery shopping, meal planning, and nutrition tracking easier to manage on an everyday basis. Using Google authentication, users will have no difficulty signing into this service to begin using it. Users will be able to add ingredients, recipes, information relating to ingredients and recipes, and more to HoMePIT. Simultaneously, HoMePIT will assist users in tracking food expiration dates, nutrition, and daily, weekly, and monthly menus, allow users to more easily compose daily menus from their input recipes and ingredients on hand, and produce shopping lists both automatically and covering user-defined time periods - depending entirely on what a user wants. With this, HoMePIT is intended to not just be an application, but rather as essential to a kitchen as the food itself.

Tools and Technology

- Platform: Firefox Browser, Chrome Browser
- Operating System (OS): Windows 10/11, Android
- Integrated Development Environment (IDE): Repl.it (online), VSCode (desktop)
- Programming Languages: Javascript, PostgreSQL, HTML, CSS
- 3rd Party Tools/Libraries: Repl.it, GitHub, Supabase, React
- Server Software: Supabase, Node.js
- Communication Software: Discord, SMS

Requirements List

1. Sign-in and Sign-out

1.1. When first opening the website, a user will be presented with a prompt to sign into the website by connecting their Google account.

- 1.1.1. On click, the user will be taken to the Google account authentication page
- 1.1.2. After signing in, users will be presented with the Pantry page and Navigation Bar

2. Navigation Bar

2.1. There will be a bar displayed with prompts for changing the currently viewed page

- 2.1.1. Clicking a prompt will change the currently viewed page
  - 2.1.1.1. There will be a prompt for the Pantry page
  - 2.1.1.2. There will be a prompt for the Recipe Book page
  - 2.1.1.3. There will be a prompt for the Meal Planner page
  - 2.1.1.4. There will be a prompt for the Shopping List page
  - 2.1.1.5. There will be a prompt for the Settings page
- 2.1.2. The current page will be visually indicated
- 2.1.3. A page that has updates or notifications will be visually indicated

3. Pantry

3.1. This page will contain a listing of ingredients determined by the user

- 3.1.1. The list will initially be empty
- 3.1.2. There will be a button to add a new ingredient
  - 3.1.2.1. On click, a modal for defining an ingredient will appear, with:
    - 3.1.2.1.1. A label for ingredient name
      - 3.1.2.1.1.1. This will be marked as required with an *
    - 3.1.2.1.2. An input box connected to the ingredient name label
      - 3.1.2.1.2.1. There will be no placeholder value
      - 3.1.2.1.2.2. It will accept characters, numbers, and punctuation
    - 3.1.2.1.3. A label for ingredient quantity
    - 3.1.2.1.4. An input box connected to the ingredient quantity label
      - 3.1.2.1.4.1. There will be no placeholder value
      - 3.1.2.1.4.2. It will accept numbers

3.1.2.1.5. A dropdown menu connected to the ingredient quantity input box

3.1.2.1.5.1. There will be no placeholder value

3.1.2.1.5.2. It will contain imperial masses

3.1.2.1.5.2.1. oz, lb

3.1.2.1.5.3. It will contain imperial volumes

3.1.2.1.5.3.1. gal, qt, pt, cup, ½ cup, ⅓ cup, ¼ cup, fl oz, tbsp, tsp, ½ tsp, ¼ tsp, ⅛ tsp

3.1.2.1.5.4. It will contain metric mass

3.1.2.1.5.4.1. kg, g

3.1.2.1.5.5. It will contain metric volumes

3.1.2.1.5.5.1. dl, cl, ml

3.1.2.1.6. A label for ingredient threshold quantity

3.1.2.1.7. An input box connected to the ingredient threshold quantity label

3.1.2.1.7.1. There will be no placeholder value

3.1.2.1.7.2. It will accept numbers

3.1.2.1.8. A dropdown menu connected to the ingredient threshold quantity input box

3.1.2.1.8.1. There will be no placeholder value

3.1.2.1.8.2. It will contain imperial masses

3.1.2.1.8.2.1. oz, lb

3.1.2.1.8.3. It will contain imperial volumes

3.1.2.1.8.3.1. gal, qt, pt, cup, ½ cup, ⅓ cup, ¼ cup, fl oz, tbsp, tsp, ½ tsp, ¼ tsp, ⅛ tsp

3.1.2.1.8.4. It will contain metric mass

3.1.2.1.8.4.1. kg, g

3.1.2.1.8.5. It will contain metric volumes

3.1.2.1.8.5.1. dl, cl, ml

3.1.2.1.9. A label for expiration date

3.1.2.1.10. An input box connected to the expiration date label

3.1.2.1.10.1. There will be no placeholder value

3.1.2.1.10.2. It will accept numbers, in the format of DD-MM-YYYY and DD-MM

3.1.2.1.11. A label for serving size

3.1.2.1.12. An input box connected to the serving size label

3.1.2.1.12.1. There will be no placeholder value

3.1.2.1.12.2. It will accept numbers

3.1.2.1.13. A dropdown menu connected to the serving size input box

3.1.2.1.13.1. There will be no placeholder value

3.1.2.1.13.2. It will contain imperial masses

3.1.2.1.13.2.1. oz, lb

3.1.2.1.13.3. It will contain imperial volumes

3.1.2.1.13.3.1. gal, qt, pt, cup, ½ cup, ⅓ cup, ¼ cup, fl oz, tbsp, tsp, ½ tsp, ¼ tsp, ⅛ tsp

3.1.2.1.13.4. It will contain metric mass

3.1.2.1.13.4.1. kg, g

3.1.2.1.13.5. It will contain metric volumes

3.1.2.1.13.5.1. dl, cl, ml

3.1.2.1.14. A label for calories per serving

3.1.2.1.15. An input box connected to the calories per serving label

3.1.2.1.15.1. There will be no placeholder value

3.1.2.1.15.2. It will accept numbers

3.1.2.1.16. A label for protein per serving

3.1.2.1.17. An input box connected to the protein per serving label

3.1.2.1.17.1. There will be no placeholder value

3.1.2.1.17.2. It will accept numbers

3.1.2.1.18. A label for fat per serving

3.1.2.1.19. An input box connected to the fat per serving label

3.1.2.1.19.1. There will be no placeholder value

3.1.2.1.19.2. It will accept numbers

3.1.2.1.20. A label for carbs per serving

3.1.2.1.21. An input box connected to the carbs per serving label

3.1.2.1.21.1. There will be no placeholder value

3.1.2.1.21.2. It will accept numbers

3.1.2.1.22. A label for purchase serving count

3.1.2.1.23. An input box connected to the purchase serving count label

3.1.2.1.23.1. There will be no placeholder value

3.1.2.1.23.2. It will accept numbers

3.1.2.1.24. A label for purchase cost

3.1.2.1.25. An input box connected to the purchase cost label

3.1.2.1.25.1. There will be no placeholder value

3.1.2.1.25.2. It will accept numbers

3.1.2.1.26. A label for purchase location

3.1.2.1.27. An input box connected to the purchase location label

3.1.2.1.27.1. There will be no placeholder value

3.1.2.1.27.2. It will accept characters, numbers, and punctuation

3.1.2.1.27.3. Multiple locations are separated by commas

3.1.2.1.28. A label for user-defined tags

3.1.2.1.29. An input box connected to the user-defined tags label

3.1.2.1.29.1. There will be no placeholder value

3.1.2.1.29.2. It will accept characters, numbers, and punctuation

3.1.2.1.29.3. Multiple tags are separated by commas

3.1.2.1.30. A label for substitutions

3.1.2.1.31. A list connected to the substitutions label

3.1.2.1.31.1. There will be a button to add a new substitution

3.1.2.1.31.1.1. On click, the Pantry (3) will appear

3.1.2.1.31.1.1.1. All described functionality will be maintained

3.1.2.1.31.1.1.2. If ingredients are selected and ‘add’ is pressed, the ingredients will be added to the substitution list.

3.1.2.1.31.1.1.2.1. When ingredients are added to the substitution list for one ingredient, that ingredient will be automatically added to the substitution list of its substitutions

3.1.3. Each ingredient on the list will display their name, quantity, and expiration date

3.1.3.1. The ingredient name can be clicked

3.1.3.1.1. On click, the modal described by 3.1.2.1 will appear

3.1.3.1.1.1. Existing data will show as placeholder data

3.1.3.2. The ingredient quantity can be clicked

3.1.3.2.1. On click, a modal for changing quantity will appear, with:

3.1.3.2.1.1. A label for ingredient quantity

3.1.3.2.1.2. An input box connected to the ingredient quantity label

3.1.3.2.1.2.1. The current quantity will be the placeholder value

3.1.3.2.1.2.2. It will accept numbers

3.1.3.2.1.3. A dropdown menu connected to the ingredient quantity input box

3.1.3.2.1.3.1. The current quantity unit will be shown

3.1.3.2.1.3.2. It will contain imperial masses

3.1.3.2.1.3.2.1. oz, lb

3.1.3.2.1.3.3. It will contain imperial volumes

3.1.3.2.1.3.3.1. gal, qt, pt, cup, ½ cup, ⅓ cup, ¼ cup, fl oz, tbsp, tsp, ½ tsp, ¼ tsp, ⅛ tsp

3.1.3.2.1.3.4. It will contain metric mass

3.1.3.2.1.3.4.1. kg, g

3.1.3.2.1.3.5. It will contain metric volumes

3.1.3.2.1.3.5.1. dl, cl, ml

3.1.3.2.1.4. A label for ingredient threshold quantity

3.1.3.2.1.5. An input box connected to the ingredient threshold quantity label

3.1.3.2.1.5.1. There will be no placeholder value

3.1.3.2.1.5.2. It will accept numbers

3.1.3.2.1.6. A dropdown menu connected to the ingredient threshold quantity input box

3.1.3.2.1.6.1. There will be no placeholder value

3.1.3.2.1.6.2. It will contain imperial masses

3.1.3.2.1.6.2.1. oz, lb

3.1.3.2.1.6.3. It will contain imperial volumes

3.1.3.2.1.6.3.1. gal, qt, pt, cup, ½ cup, ⅓ cup, ¼ cup, fl oz, tbsp, tsp, ½ tsp, ¼ tsp, ⅛ tsp

3.1.3.2.1.6.4. It will contain metric mass

3.1.3.2.1.6.4.1. kg, g

3.1.3.2.1.6.5. It will contain metric volumes

3.1.3.2.1.6.5.1. dl, cl, ml

3.1.3.3. If the ingredient quantity is below the ingredient threshold quantity, there will be a visual indication nearby the shown quantity.

3.1.4. Each ingredient on the list will have a “remove” button

3.1.4.1. On click, the given item will be removed from the pantry

3.1.5. The list will be sorted alphabetically, low-to-high by default

3.1.6. There will be a button to change the list’s sorting

3.1.6.1. On click, a modal for changing sorting will appear, with:

3.1.6.1.1. A label for metric to sort by

3.1.6.1.2. A dropdown list connected to the sorting metric label

3.1.6.1.2.1. The dropdown list will have a placeholder value of ‘alphabetical’

3.1.6.1.2.1.1. If another option was selected earlier, that option will be the placeholder

3.1.6.1.2.2. The list will contain the following options:

3.1.6.1.2.2.1. Alphabetical

3.1.6.1.2.2.2. Quantity

3.1.6.1.2.2.3. Expiration Date

3.1.6.1.2.2.4. Protein per serving

3.1.6.1.2.2.5. Fat per serving

3.1.6.1.2.2.6. Carbohydrates per serving

3.1.6.1.3. A label for sorting order

3.1.6.1.4. A dropdown list connected to the sorting order label

3.1.6.1.4.1. The dropdown list will have a placeholder value of ‘low-to-high’

3.1.6.1.4.1.1. If another option was selected earlier, that option will be the placeholder

3.1.6.1.4.2. The list will contain the following options:

3.1.6.1.4.2.1. Low-to-high

3.1.6.1.4.2.2. High-to-low

3.1.6.1.5. A label for user tags to be included

3.1.6.1.6. An input box connected to the user-defined tags label

3.1.6.1.6.1. There will be no placeholder value

3.1.6.1.6.2. It will accept characters, numbers, and punctuation

3.1.6.1.6.3. Multiple tags are separated by commas

3.1.6.1.7. A label for user tags to exclude

3.1.6.1.8. An input box connected to the user-defined tags label

3.1.6.1.8.1. There will be no placeholder value

3.1.6.1.8.2. It will accept characters, numbers, and punctuation

3.1.6.1.8.3. Multiple tags are separated by commas

3.1.7. There will be a button to reset the list’s sorting

3.1.7.1. On click, the sorting will be reset to alphabetical, low-too-high

3.2. When opened via the Recipe Book to select ingredients, the Pantry to select substitutions, or the Shopping List to create a menu:

3.2.1. Every ingredient in the list (3.1) will have a toggle button connected to the ingredient name

3.2.1.1. The placeholder value will be untoggled

3.2.1.2. On click, the button will toggle

3.2.2. The page will have an ‘add’ button

3.2.2.1. On click, if the user opened the Pantry from the Recipe Book to add ingredients, selected ingredients will be added to the recipe being written and the user will return to the recipe creation modal (4.1.2.1)

3.2.2.2. On click, if the user opened the Pantry from the Pantry to add substitutions, selected ingredients will be added to the substitution list being written and the user will return to the ingredient creation modal (3.1.2.1)

3.2.2.3. On click, if the user opened the Pantry from the Shopping List to add ingredients, selected ingredients will be added to the shopping list being modified and the user will return to the Shopping List page (6)

3.2.3. The page will have a ‘close’ button

3.2.3.1. On click, if the user opened the Pantry from the Recipe Book to add ingredients, the user will return to the recipe creation modal (4.1.2.1)

3.2.3.2. On click, if the user opened the Pantry from the Pantry to add substitutions, the user will return to the ingredient creation modal (3.1.2.1)

3.2.3.3. On click, if the user opened the Pantry from the Shopping List to add ingredients, the user will return to the Shopping List page (6)

4. Recipe Book

4.1. This page will contain a listing of recipes input by the user

4.1.1. The list will initially be empty

4.1.2. There will be a button to add a new recipe

4.1.2.1. On click, a modal for defining a recipe will appear, with:

4.1.2.1.1. A label for recipe name

4.1.2.1.1.1. This will be marked as required with an *

4.1.2.1.2. An input box connected to the recipe name label

4.1.2.1.2.1. There will be no placeholder value

4.1.2.1.2.2. It will accept numbers, characters, and punctuation

4.1.2.1.3. A label for serving count

4.1.2.1.4. An input box connected to the serving count label

4.1.2.1.4.1. There will be no placeholder value

4.1.2.1.4.2. It will accept leftover serving count

4.1.2.1.5. A label for recipe cook time

4.1.2.1.6. An input box connected to the cook time label

4.1.2.1.6.1. There will be no placeholder value

4.1.2.1.6.2. It will accept numbers and punctuation

4.1.2.1.7. A label for total recipe cook time

4.1.2.1.8. An input box connected to the total recipe cook time label

4.1.2.1.8.1. There will be no placeholder value initially

4.1.2.1.8.2. After ingredients and sub-recipes are selected, a placeholder value will appear that is the sum of the input cook time (4.1.2.1.7) and all sub-recipe cook times

4.1.2.1.8.2.1. If a time has already been entered, it will not be replaced by this automatically calculated time

4.1.2.1.8.3. It will accept numbers and punctuation in the format of HH:MM, 24-hour time

4.1.2.1.8.4. If an invalid time is input, there will be a visual indicator indicating such

4.1.2.1.9. A label for calories, protein, fat, and carbohydrates per serving

4.1.2.1.10. A label connected to the calories, protein, fat, and carbohydrates label that displays said information

4.1.2.1.10.1. Before ingredients and sub-recipes are selected, there will be no placeholder value

4.1.2.1.10.2. After ingredients and sub-recipes are selected, the information to be displayed will be automatically calculated from selected ingredients and sub-recipes

4.1.2.1.10.3. If one or more ingredients is lacking nutritional information, a visual indicator will be displayed to indicate this

4.1.2.1.11. A label for cost per serving

4.1.2.1.12. A label connected to the cost per serving label

4.1.2.1.12.1. Before ingredients and sub-recipes are selected, there will be no placeholder value

4.1.2.1.12.2. After ingredients and sub-recipes are selected, the information to be displayed will be automatically calculated from selected ingredients and sub-recipes

4.1.2.1.12.3. If one or more ingredients is lacking nutritional information, a visual indicator will be displayed to indicate this

4.1.2.1.13. A label for predicted leftover expiration time

4.1.2.1.14. An input box connected to the predicted leftover expiration time label

4.1.2.1.14.1. There will be no placeholder value

4.1.2.1.14.2. It will accept numbers

4.1.2.1.15. A dropdown menu connected to the predicted leftover expiration time input box

4.1.2.1.15.1. It will have values of ‘days’ and ‘weeks’

4.1.2.1.15.2. It will have a placeholder value of days

4.1.2.1.16. A label for user tags

4.1.2.1.17. An input box connected to the user-defined tags label

4.1.2.1.17.1. The placeholder value will be a list of the unique user-defined tags assigned to its ingredients and sub-recipes

4.1.2.1.17.2. It will accept characters, numbers, and punctuation

4.1.2.1.17.3. Multiple tags are separated by commas

4.1.2.1.18. A label for ingredients and sub-recipes

4.1.2.1.18.1. This will be marked as required with an *

4.1.2.1.19. A list connected to the ingredients label

4.1.2.1.19.1. The list will initially be empty

4.1.2.1.19.2. There will be a button to add a new ingredient

4.1.2.1.19.2.1. On click, the Pantry (3) will appear

4.1.2.1.19.2.1.1. All described functionality will be maintained.

4.1.2.1.19.2.1.2. If ingredients are selected and 'add' is pressed, the ingredients will be added to the ingredients list.

4.1.2.1.19.3. There will be a button to add a new sub-recipe

4.1.2.1.19.3.1. On click, the Recipe Book (4) will appear

4.1.2.1.19.3.1.1. All described functionality will be maintained, with exception of 4.1.2

4.1.2.1.19.3.1.2. If sub-recipes are selected and 'add' is pressed, the sub-recipes will be added to the ingredients list.

4.1.2.1.19.4. Each ingredient or sub-recipe added to the list will have:

4.1.2.1.19.4.1. A label for ingredient or sub-recipe quantity

4.1.2.1.19.4.2. An input box connected to the ingredient or sub-recipe quantity label

4.1.2.1.19.4.2.1. There will be no placeholder value

4.1.2.1.19.4.2.2. It will accept numbers

4.1.2.1.19.4.3. A dropdown menu connected to the ingredient or sub-recipe quantity input box

4.1.2.1.19.4.3.1. There will be no placeholder value

4.1.2.1.19.4.3.2. It will contain imperial masses

4.1.2.1.19.4.3.2.1. oz, lb

4.1.2.1.19.4.3.3. It will contain imperial volumes

4.1.2.1.19.4.3.3.1. gal, qt, pt, cup, ½ cup, ⅓ cup, ¼ cup, fl oz, tbsp, tsp, ½ tsp, ¼ tsp, ⅛ tsp

4.1.2.1.19.4.3.4. It will contain metric mass

4.1.2.1.19.4.3.4.1. kg, g

4.1.2.1.19.4.3.5. It will contain metric volumes

4.1.2.1.19.4.3.5.1. dl, cl, ml

4.1.3. Each recipe on the list will display their name, cook time, calories per serving, grams of protein per serving, grams of fat per serving, grams of carbohydrates per serving, number of leftovers, time until leftover expiry, and time since last creation

4.1.3.1. Name

4.1.3.1.1. The recipe name can be clicked

4.1.3.1.1.1. On click, the recipe creation modal (4.1.2.1) will appear

4.1.3.1.1.1.1. Existing data will show as placeholder data

4.1.3.2. Cook time

4.1.3.3. Calories per serving

4.1.3.4. Grams of protein per serving

4.1.3.5. Grams of fat per serving

4.1.3.6. Grams of carbohydrates per serving

4.1.3.7. Cost per serving

4.1.3.8. Number of leftovers

4.1.3.9. Time until leftover expiry

4.1.3.10. Time since last creation

4.1.4. Each recipe on the list will have a “remove” button

4.1.4.1. On click, the given item will be removed from the recipe book

4.1.5. The list will be sorted alphabetically by default

4.1.6. There will be a button to change the list’s sorting

4.1.6.1. On click, a modal for changing sorting will appear, with:

4.1.6.1.1. A label for metric to sort by

4.1.6.1.2. A dropdown list connected to the sorting metric label

4.1.6.1.2.1. The dropdown list will have a placeholder value of ‘alphabetical’

4.1.6.1.2.1.1. If another option was selected earlier, that option will be the placeholder

4.1.6.1.2.2. The list will contain the following options:

4.1.6.1.2.2.1. Alphabetical

4.1.6.1.2.2.2. Calories per serving

4.1.6.1.2.2.3. Protein per serving

4.1.6.1.2.2.4. Fat per serving

4.1.6.1.2.2.5. Carbohydrates per serving

4.1.6.1.2.2.6. Cook time

4.1.6.1.2.2.7. Time since last cooking

4.1.6.1.2.2.8. Leftover freshness time

4.1.6.1.2.2.9. Number of leftovers

4.1.6.1.2.2.10. Time until leftover expiry

4.1.6.1.2.3.

4.1.6.1.3. A label for sorting order

4.1.6.1.4. A dropdown list connected to the sorting order label

4.1.6.1.4.1. The dropdown list will have a placeholder value of ‘low-to-high’

4.1.6.1.4.1.1. If another option was selected earlier, that option will be the placeholder

4.1.6.1.4.2. The list will contain the following options:

4.1.6.1.4.2.1. Low-to-high

4.1.6.1.4.2.2. High-to-low

4.1.6.1.5. A label for user tags to include

4.1.6.1.6. An input box connected to the user-defined tags label

4.1.6.1.6.1. There will be no placeholder value

4.1.6.1.6.2. It will accept characters, numbers, and punctuation

4.1.6.1.6.3. Multiple tags are separated by commas

4.1.6.1.7. A label for user tags to exclude

4.1.6.1.8. An input box connected to the user-defined tags label

4.1.6.1.8.1. There will be no placeholder value

4.1.6.1.8.2. It will accept characters, numbers, and punctuation

4.1.6.1.8.3. Multiple tags are separated by commas

4.1.6.1.9. A label for ingredients to include

4.1.6.1.10. An input box connected to the ingredients label

4.1.6.1.10.1. There will be no placeholder value

4.1.6.1.10.2. It will accept characters, numbers, and punctuation

4.1.6.1.10.3. Multiple ingredients are separated by commas

4.1.6.1.11. A label for ingredients to exclude

4.1.6.1.12. An input box connected to the ingredients label

4.1.6.1.12.1. There will be no placeholder value

4.1.6.1.12.2. It will accept characters, numbers, and punctuation

4.1.6.1.12.3. Multiple ingredients are separated by commas

4.1.7. There will be a button to reset the list's sorting

4.1.7.1. On click, the sorting will be reset to alphabetical, low-too-high

4.1.8. There will be an input box to search for recipes

4.1.8.1. There will be no placeholder value

4.1.8.2. It will accept numbers, characters, and punctuation

4.1.8.3. When a value has been input, the recipe list should limit its shown recipes to those containing the input value

4.1.9. There will be an input box to search for recipes by ingredient

4.1.9.1. There will be no placeholder value

4.1.9.2. It will accept numbers, characters, and punctuation

4.1.9.3. When a value has been input, the recipe list should limit its shown recipes to those containing the input value

4.2. When opened via the Recipe Book to select sub-recipes or the Meal Planner to select recipes:

4.2.1. Every recipe in the list (4.1) will have a toggle button connected to the recipe name label

4.2.1.1. The placeholder value will be untoggled

4.2.1.2. On click, the button will toggle

4.2.2. The page will have an 'add' button

4.2.2.1. On click, if the user opened the Recipe Book from the Recipe Book to add sub-recipes, selected recipes will be added to the recipe being written as sub-recipes and the user will return to the recipe creation modal (4.1.2.1)

4.2.2.2. On click, if the user opened the Recipe Book from the Meal Planner to add recipes, selected recipes will will be added to the specific meal list being created as recipes and the user will return to the meal planner modal

4.2.3. The page will have a ‘close’ button

4.2.3.1. On click, if the user opened the Recipe Book from the Recipe Book to add sub-recipes, the Recipe Book will close and the user will return to the recipe creation modal (4.1.2.1)

4.2.3.2. On click, if the user opened the Recipe Book from the Meal Planner to add recipes, the Recipe Book will close and the user will return to the meal planner modal

4.2.4. When opened via the Meal Planner to select recipes, recipes that have available ingredients will be visually indicated by a color. Recipes that have available ingredients within a week of expiry will be visually indicated by a different color. Recipes that lack one or more ingredients but have substitutions will be indicated by a different color. Recipes that lack one or more ingredients without available substitutions will be visually indicated by another color.

5. Meal Planner

5.1. This page will contain a calendar

5.1.1. The calendar will present all days of the current month

5.1.2. Each day of the month will be clickable

5.1.2.1. On click, a modal will appear with:

5.1.2.1.1. A list

5.1.2.1.1.1. The list will initially be empty

5.1.2.1.1.2. The list will be divided into several sections, each with:

5.1.2.1.1.2.1. A label for either breakfast, brunch, lunch, dinner, dessert, or snack

5.1.2.1.1.2.2. A label for meal time

5.1.2.1.1.2.3. An input box connected to each meal time label

5.1.2.1.1.2.3.1. There will be no placeholder value

5.1.2.1.1.2.3.2. It will accept numbers and punctuation in the format of HH:MM, 24-hour time

5.1.2.1.1.2.3.3. If an invalid time is input, there will be a visual indicator indicating such

5.1.2.1.1.2.4. A label for number of servings required

5.1.2.1.1.2.5. An input box connected to each required servings label

5.1.2.1.1.2.5.1. There will be no placeholder value

5.1.2.1.1.2.5.2. It will accept numbers

5.1.2.1.1.2.6. A button to add recipes

5.1.2.1.1.2.6.1. On click, the Recipe Book (4) will appear

5.1.2.1.1.2.6.1.1. All described functionality will be maintained

5.1.2.1.1.2.6.1.2. If recipes are selected and ‘add’ is pressed, the recipes will be added to the meal list

5.1.2.1.1.2.7. A button to remove recipes

5.1.2.1.1.2.7.1. On click, the recipe will be removed from the section

5.1.2.1.1.2.8. A label for starting cook time

5.1.2.1.1.2.9. A label connected to each section’s starting cook time label

5.1.2.1.1.2.9.1. There will be no placeholder value

5.1.2.1.1.2.9.2. After a section’s meal time is entered (5.1.2.1.1.2.3), the label will present the latest time that meal preparation can start based on the input meal time and the longest of the selected non-leftover-using recipes’ cook times, taking into account whether sub-recipes are available or need to be prepared

5.1.2.1.1.2.10. Each recipe in each section will have:

5.1.2.1.1.2.10.1. A label for recipe name

5.1.2.1.1.2.10.2. A radio button with 2 options

5.1.2.1.1.2.10.3. A label connected to the other radio button for for ‘use leftovers’

5.1.2.1.1.2.10.4. A label connected to one radio button for ‘cook from scratch’

5.1.2.1.1.2.10.5. A label connected to the ‘cook from scratch’ label for ‘number of servings to cook’

5.1.2.1.1.2.10.6. An input box connected to each ‘number of servings to cook’ label

5.1.2.1.1.2.10.6.1. The placeholder value will be the number of servings made by an instance of the recipe, as input by the user in the Recipe Book

5.1.2.1.1.2.10.6.2. It will accept numbers

5.1.2.1.1.2.10.6.3. If a user inputs a value below or above the default number of servings for a recipe, there will be a separate visual indicator of each case

5.1.2.1.1.2.10.6.4. If the radio button is toggled to ‘use leftovers’, this input box will become noninteractive

5.1.2.1.1.2.10.7. A label for predicted total number of leftovers’

5.1.2.1.1.2.10.8. An input box connected to the ‘predicted total number of leftovers’ label

5.1.2.1.1.2.10.8.1. The placeholder value will be the number of servings produced according to the following formulae: when ‘cook from scratch’ is selected: ‘number of servings to cook’ - ‘number of servings the meal requires’; when ‘use leftovers’ is selected: ‘number of current leftovers’ - ‘number of servings the meal requires’

5.1.2.1.1.2.10.9. If a recipe contains sub-recipes: a list of sub-recipes connected to that recipe, with:

5.1.2.1.1.2.10.9.1. A label for ‘number of sub-recipe servings left’

5.1.2.1.1.2.10.9.2. A label that displays the current number of sub-recipe servings left, connected to the label for ‘number of sub-recipe servings left’

5.1.2.1.1.2.10.9.3. A label that displays the number of sub-recipe servings that will be left after the meal has been cooked, connected to the label for ‘number of sub-recipe servings left’

5.1.2.1.1.2.10.10. If a recipe contains possible substitutions: a list of substitutions connected to that recipe, with:

5.1.2.1.1.2.10.10.1. A list for each ingredient, containing the ingredient that may be substituted and possible substitutes for that ingredient that does not include any ingredients already in the recipe, wherein substitutions with sufficient quantity are indicated visually

5.1.2.1.1.2.10.10.2. A radio button paired with each ingredient and possible substitute. The placeholder value will be for the base ingredient radio button to be enabled

5.1.2.1.1.2.10.11. A label for starting cook time

5.1.2.1.1.2.10.12. A label connected to each recipe's starting cook time label

5.1.2.1.1.2.10.12.1. There will be no placeholder value

5.1.2.1.1.2.10.12.2. After a section's meal time is entered (5.1.2.1.1.2.3), the label will present the latest time that that recipe's preparation can start based on the input meal time

5.1.2.1.2. A label for daily calories

5.1.2.1.3. A label connected to the daily calories label

5.1.2.1.3.1. If daily calories cannot be calculated, no value will be displayed

5.1.2.1.3.2. Otherwise, it will display the total number of calories, assuming one serving of each recipe was consumed

5.1.2.1.4. A label for daily proteins

5.1.2.1.5. A label connected to the daily protein label

5.1.2.1.5.1. If daily proteins cannot be calculated, no value will be displayed

5.1.2.1.5.2. Otherwise, it will display the total grams of protein, assuming one serving of each recipe was consumed

5.1.2.1.6. A label for daily fats

5.1.2.1.7. A label connected to the daily fats label

5.1.2.1.7.1. If daily fats cannot be calculated, no value will be displayed

5.1.2.1.7.2. Otherwise, it will display the total grams of fats, assuming one serving of each recipe was consumed

5.1.2.1.8. A label for daily carbohydrates

5.1.2.1.9. A label connected to the daily carbohydrates label

5.1.2.1.9.1. If daily carbohydrates cannot be calculated, no value will be displayed

5.1.2.1.9.2. Otherwise, it will display the total grams of carbohydrates, assuming one serving of each recipe was consumed

5.1.2.1.10. A label for daily cost

5.1.2.1.11. A label connected to the daily cost label

5.1.2.1.11.1. If daily cost cannot be calculated, no value will be displayed

5.1.2.1.11.2. Otherwise, it will display the total cost

5.1.3. Each week of the year will have a button with an internal label containing its numeric value (according to ISO 8601)

5.1.3.1. On click, a textual summary of that week's nutritional and financial information will appear in a modal

5.1.3.1.1. The modal will:

5.1.3.1.1.1. Contain the following information for the week:

5.1.3.1.1.1.1. Total calories

5.1.3.1.1.1.2. Daily calories

5.1.3.1.1.1.3. Total protein

5.1.3.1.1.1.4. Daily protein

5.1.3.1.1.1.5. Total fat

5.1.3.1.1.1.6. Daily fat

5.1.3.1.1.1.7. Total carbohydrates

5.1.3.1.1.1.8. Daily carbohydrates

5.1.3.1.1.1.9. Total cost

5.1.3.1.1.1.10. Daily cost

5.1.3.1.2. Not display a value if the value cannot be calculated due to missing information

5.1.3.1.3. Otherwise it will display a value, with nutritional information assuming one serving of each recipe in the week was consumed

5.1.4. Each month of the year will have a button with an internal label displaying its name

5.1.4.1. On click, a textual summary of that month's nutritional and financial information will appear in a modal

5.1.4.1.1. The modal will:

5.1.4.1.1.1. Contain the following information for the month:

5.1.4.1.1.1.1. Total calories

5.1.4.1.1.1.2. Weekly calories

5.1.4.1.1.1.3. Daily calories

5.1.4.1.1.1.4. Total protein

5.1.4.1.1.1.5. Weekly protein

5.1.4.1.1.1.6. Daily protein

5.1.4.1.1.1.7. Total fat

5.1.4.1.1.1.8. Weekly fat

5.1.4.1.1.1.9. Daily fat

5.1.4.1.1.1.10. Total carbohydrates

5.1.4.1.1.1.11. Weekly carbohydrates

5.1.4.1.1.1.12. Daily carbohydrates

5.1.4.1.1.1.13. Total cost

5.1.4.1.1.1.14. Weekly cost

5.1.4.1.1.1.15. Daily cost

5.1.4.1.2. Not display a value if the value cannot be calculated due to missing information

5.1.4.1.3. Otherwise it will display a value, with nutritional information assuming one serving of each recipe in the month was consumed

5.1.5. There will be a button to calculate information over a specific date range

5.1.5.1. On click, a modal will appear with:

5.1.5.1.1. A label for starting date

5.1.5.1.2. An input box connected to the label for starting date

5.1.5.1.2.1. The placeholder value will display the date format of DD-MM-YYYY

5.1.5.1.2.2. It will accept numbers and punctuation

5.1.5.1.2.3. If an invalid date is input, the user should be alerted of this by a visual indicator

5.1.5.1.3. A label for ending date

5.1.5.1.4. An input box connected to the label for ending date

5.1.5.1.4.1. The placeholder value will display the date format of DD-MM-YYYY

5.1.5.1.4.2. It will accept numbers and punctuation

5.1.5.1.4.3. If an invalid date is input, the user should be alerted of this by a visual indicator

5.1.5.1.5. A button to calculate nutritional and financial information

5.1.5.1.5.1. On click, if there in an error with the starting and end dates, a dialog will appear to inform the user of the specific error

5.1.5.1.5.2. Otherwise, a textual summary of that time period's nutritional and financial information will appear in a modal

5.1.5.1.5.2.1. The modal will:

5.1.5.1.5.2.1.1. Contain the following information for the time period, as they apply:

5.1.5.1.5.2.1.1.1. Total calories

5.1.5.1.5.2.1.1.2. Weekly calories

5.1.5.1.5.2.1.1.3. Daily calories

5.1.5.1.5.2.1.1.4. Total protein

5.1.5.1.5.2.1.1.5. Weekly protein

5.1.5.1.5.2.1.1.6. Daily protein

5.1.5.1.5.2.1.1.7. Total fat

5.1.5.1.5.2.1.1.8. Weekly fat

5.1.5.1.5.2.1.1.9. Daily fat

5.1.5.1.5.2.1.1.10. Total carbohydrates

5.1.5.1.5.2.1.1.11. Weekly carbohydrates

5.1.5.1.5.2.1.1.12. Daily carbohydrates

5.1.5.1.5.2.1.1.13. Total cost

5.1.5.1.5.2.1.1.14. Weekly cost

5.1.5.1.5.2.1.1.15. Daily cost

5.1.5.1.5.2.2. Not display a value if the value cannot be calculated due to missing information

5.1.5.1.5.2.3. Otherwise it will display a value, with nutritional information assuming one serving of each recipe in the month was consumed

6. Shopping List

6.1. On opening the page, if the user has not defined a grocery trip frequency and preferred shopping day, and has not not toggled off regular grocery trips, open a modal with:

6.1.1. A label for grocery trip frequency

6.1.2. An input box connected to the grocery trip frequency label

6.1.2.1. There will be no placeholder value

6.1.2.2. It will accept numbers

6.1.3. A dropdown menu connected to the grocery trip frequency input box

6.1.3.1. There will be a placeholder value of ‘days’

6.1.3.2. It will contain:

6.1.3.2.1. Days

6.1.3.2.2. Weeks

6.1.4. A label for preferred grocery shopping day

6.1.5. A dropdown menu connected to the preferred grocery shopping day

6.1.5.1. There will be a placeholder value of ‘Friday’

6.1.5.2. It will contain:

6.1.5.2.1. Monday

6.1.5.2.2. Tuesday

6.1.5.2.3. Wednesday

6.1.5.2.4. Thursday

6.1.5.2.5. Friday

6.1.5.2.6. Saturday

6.1.5.2.7. Sunday

6.1.6. If the modal is closed without setting a grocery trip frequency and preferred grocery shopping day, toggle off the regular grocery trips setting (7.1.1).

6.2. This page will contain:

6.2.1. A list of shopping lists

6.2.1.1. The list will be divided into shopping lists according to the span of meals any given shopping list should cover

6.2.1.1.1. If a user has defined a grocery trip frequency and preferred shopping day, these sections will be automatically sized and chronologically spaced accordingly

6.2.1.1.2. Otherwise, they will be sized according to the user’s manually specified shopping lists (6.2.2)

6.2.1.1.3. Each shopping list will attempt to minimize the number of grocery stores to be visited

6.2.1.2. Each shopping list will have:

6.2.1.2.1. A button containing a label for the date range it satisfies

6.2.1.2.1.1. On click, open a modal with:

6.2.1.2.1.1.1. A label for the first day a shopping trip should satisfy

6.2.1.2.1.1.2. An input box connected to the first-day-satisfied label

6.2.1.2.1.1.2.1. The placeholder value will be the first day of the satisfied date range

6.2.1.2.1.1.2.2. It will accept numbers and punctuation

6.2.1.2.1.1.2.3. If an invalid date is input, the user should be alerted of this by a visual indicator

6.2.1.2.1.1.3. A label for the last day a shopping trip should satisfy

6.2.1.2.1.1.4. An input box connected to the last-satisfied-day label

6.2.1.2.1.1.4.1. The placeholder value will be the last day of the satisfied date range

6.2.1.2.1.1.4.2. It will accept numbers and punctuation

6.2.1.2.1.1.4.3. If an invalid date is input, the user should be alerted of this by a visual indicator

6.2.1.2.1.1.5. If the modal is applied, adjacent shopping lists should have their date ranges updated to prevent meals from not being covered by a shopping list. All affected shopping lists should update their contents accordingly.

6.2.1.2.2. A button to add ingredients to the shopping list manually

6.2.1.2.2.1. On click, the Pantry (3) will appear

6.2.1.2.2.1.1. All described functionality will be maintained

6.2.1.2.2.1.2. If ingredients are selected and ‘add’ is pressed, the ingredients will be added to the shopping list list.

6.2.1.2.3. A button to ‘close out’ a shopping list and confirm that it has been used and its contents have been purchased

6.2.1.2.3.1. On click,

6.2.1.2.4. Any ingredients with insufficient quantity to satisfy the planned meals in the time covered by the shopping list

6.2.1.2.5. Any ingredients whose quantity would fall below its threshold quantity to satisfy the planned meals in the time covered by the shopping list

6.2.1.2.6. Any ingredients that would expire before they get used for a meal in the time covered by the shopping list

6.2.1.3. Each ingredient in a shopping list will have:

6.2.1.3.1. A label for its name

6.2.1.3.2. An input box for the quantity being purchased

6.2.1.3.2.1. The predefined value will be the quantity needed to satisfy the menu planned for the shopping list's date range

6.2.1.3.2.2. It will accept numbers and punctuation

6.2.1.3.3. A dropdown menu for the quantity unit

6.2.1.3.3.1. The placeholder value will be the predefined ingredient quantity unit

6.2.1.3.3.2. It will contain imperial masses

6.2.1.3.3.2.1. oz, lb

6.2.1.3.3.3. It will contain imperial volumes

6.2.1.3.3.3.1. gal, qt, pt, cup, ½ cup, ⅓ cup, ¼ cup, fl oz, tbsp, tsp, ½ tsp, ¼ tsp, ⅛ tsp

6.2.1.3.3.4. It will contain metric mass

6.2.1.3.3.4.1. kg, g

6.2.1.3.3.5. It will contain metric volumes

6.2.1.3.3.5.1. dl, cl, ml

6.2.1.3.4. A label for ingredient cost

6.2.1.3.5. A button to remove the ingredient from the shopping list

6.2.1.3.5.1. On click, the ingredient paired with the button should be removed from the shopping list

6.2.2. A button to define a new shopping list

6.2.2.1. On click, open a modal with:

6.2.2.1.1. A label for the first day a shopping trip should satisfy

6.2.2.1.2. An input box connected to the first-day-satisfied label

6.2.2.1.2.1. The placeholder value will be the date format of DD-MM-YYYY

6.2.2.1.2.2. It will accept numbers and punctuation

6.2.2.1.2.3. If an invalid date is input, the user should be alerted of this by a visual indicator

6.2.2.1.3. A label for the last day a shopping trip should satisfy

6.2.2.1.4. An input box connected to the last-satisfied-day label

6.2.2.1.4.1. The placeholder value will be the date format of DD-MM-YYYY

6.2.2.1.4.2. It will accept numbers and punctuation

6.2.2.1.4.3. If an invalid date is input, the user should be alerted of this by a visual indicator

6.2.2.2. If the user input data and applied it, shopping lists should be modified to account for the new shopping list's satisfied date range, and the new shopping list should be inserted in its correct chronological position in the list of shopping lists

7. Settings

7.1. This page will contain:

7.1.1. A toggle button for regular grocery trip (automatic shopping list creation)

7.1.1.1. The placeholder value will be ‘toggled’

7.1.2. A label for grocery trip frequency

7.1.3. An input box connected to the grocery trip frequency label

7.1.3.1. There will be no placeholder value

7.1.3.2. It will accept numbers

7.1.4. A dropdown menu connected to the grocery trip frequency input box

7.1.4.1. There will be a placeholder value of ‘days’

7.1.4.2. It will contain:

7.1.4.2.1. Days

7.1.4.2.2. Weeks

7.1.5. A label for preferred grocery shopping day

7.1.6. A dropdown menu connected to the preferred grocery shopping day

7.1.6.1. There will be a placeholder value of ‘Friday’

7.1.6.2. It will contain:

7.1.6.2.1. Monday

7.1.6.2.2. Tuesday

7.1.6.2.3. Wednesday

7.1.6.2.4. Thursday

7.1.6.2.5. Friday

7.1.6.2.6. Saturday

7.1.6.2.7. Sunday

7.1.7. A button to sign out

7.1.7.1. On click, the user will be signed out and presented the sign-in page

8. Misc

8.1. When a ingredient or leftover has an approaching expiration, a modal will appear informing the user of impending expiration

8.1.1. The user will be informed when 7 days away from expiration

8.1.2. The user will be informed when 3 days away from expiration

8.1.3. The user will be informed when 1 days away from expiration

8.1.4. This modal will mention the ingredient or leftover in question

8.1.5. Any ingredient or leftover that will expire within a day of the ingredient that triggers this modal will also be mentioned, but this will not recursively propagate

8.2. The color scheme should be neutral and light

9. Modal

9.1. All modals will have a button to close the modal

9.2. All modals taking user input will have a button to apply the user input and close the modal

Timeline

- September 3 - September 9
  - Group: Submit Project Proposal
  - Erik: Research PostgreSQL.
  - Jerome: Research Node.js, Supabase.
  - Maxwell: Research Node.js, Supabase.
- September 10 - September 16
  - Group: Set up an environment for project development, including Git repository. Begin code experiments. Determine rough idea of UI and UX desired.
  - Erik: Continue PostgreSQL research, begin planning database.
  - Jerome: Continue PostgreSQL and Node.js research, begin planning database.
  - Maxwell: Set up Supabase and Repl.it.
- September 17 - September 23
  - Group:
  - Erik: Establish PostgreSQL database. Implement ingredients and recipe database entries as implicitly described in the Feature List.
  - Jerome: Finalize UI and UX plan.
  - Maxwell: Finalize UI and UX plan. Establish user sign in and authentication in Supabase.
- September 24 - September 30
  - Group:
  - Erik: Implement code to create, modify, and remove ingredient database entries. Implement code to create, modify, and remove recipe database entries without sub-recipes.
  - Jerome: Implement code and implement UI/UX skeleton on the frontend.
  - Maxwell: Implement UI/UX skeleton on the frontend.
- October 1 - October 7
  - Group: Test each other's deliverables from the previous week. Integrate deliverables where possible and test the result.
  - Erik: Implement code to create recipe database entries with subrecipes. Implement code to automatically populate recipe nutritional information based on ingredients and recipes.
  - Jerome: Implement code and/or implement frontend application of creation, modification, and removal of ingredients.
  - Maxwell: Implement frontend application of creation, modification, and removal of ingredients. Implement additional features when applicable.
- October 8 - October 14
  - Group: Test each other's deliverables from the previous week. Integrate deliverables where possible and test the result.
  - Erik: Implement planned meals database. Implement code to add, modify, and remove planned meal database entries.
  - Jerome: Implement backend code and/or implement frontend application of recipe entry, display of nutritional information, etc.
  - Maxwell: Implement frontend application of recipe entry, display of nutritional information, and additional features when applicable.

- October 15 - October 21
  - Group: Test each other’s deliverables from the previous week. Integrate deliverables where possible and test the result.
  - Erik: Implement database-adjacent code enabling described meal planning functionality.
  - Jerome: Implement meal planning database-adjacent code and/or implement frontend code. Assist in Implementing calendar functionality related to meal planning.
  - Maxwell: Implement frontend application of the addition of, modification of, and removal of meals from the planned meals database. Implement calendar functionality related to meal planning. Implement additional features when applicable.
- October 22 - October 28
  - Group: Test each other’s deliverables from the previous week. Integrate deliverables where possible and test the result.
  - Erik: Implement code to create shopping lists from planned meal database entries and ingredient database entries below their threshold.
  - Jerome: Implement code to create shopping lists and/or implement further functionality from the week prior.
  - Maxwell: Implement further functionality from the week prior, and additional features when applicable.
- October 29 - November 4
  - Group: Test each other’s deliverables from the previous week. Integrate deliverables where possible and test the result.
  - Erik: Implement supplementary database-adjacent shopping list code.
  - Jerome: Implement supplementary database-adjacent shopping list code and/or implement frontend shopping list feature.
  - Maxwell: Implement frontend application of the shopping list feature, and additional features when applicable.
- November 5 - November 11
  - Group: Begin creating the poster, if possible.
  - Erik: Delegate poster creation duties. Work on assigned poster duties. Test frontend functionality.
  - Jerome: Work on assigned poster duties. Assist in testing frontend and backend functionality.
  - Maxwell: Work on assigned poster duties. Test backend functionality.
- November 12 - November 18
  - Group: Begin poster if necessary, preferably begin finalizing poster. Begin finalizing the codebase.
  - Erik: Delegate poster creation duties. Work on assigned poster duties. Check that all guaranteed features have been completely implemented or are approaching that stage.
  - Jerome: Work on assigned poster duties. Assist Erik and Maxwell as necessary.
  - Maxwell: Work on assigned poster duties. Determine what changes need to be made to the frontend, if any.
- November 19 - November 25

    - Deadline: November 22: Poster due
    - Group: Finish and turn in the poster. Finalize codebase, aside from necessary changes mandated by testing.
    - Erik: Polish assigned sections of poster. Unify poster voice and style. Make any final backend changes necessary to ensure guaranteed features are available.
    - Jerome: Polish assigned sections of poster. Make any necessary frontend changes to ensure guaranteed features are available.
    - Maxwell: Polish assigned sections of poster. Make any necessary frontend changes to ensure guaranteed features are available.
- November 26 - December 2
    - Group: Prepare for technical interviews by individually studying the codebase, and asking each other questions concerning what one has worked on. Begin preparing the presentation, demo, final report, and supporting documentation.
    - Erik: Set up the basic outline of the final report. Delegate report writing duties. Contribute to necessary sections of the final report, presentation, demo, and documentation.
    - Jerome: Set up the basic outline of the presentation. Delegate presentation writing duties. Set up the basic outline of the demo. Contribute to necessary sections of the final report, presentation, demo, and documentation.
    - Maxwell: Document what documentation is required. Delegate supporting documentation duties. Contribute to necessary sections of the final report, presentation, demo, and documentation.
- December 3 - December 9
    - Deadline: December 6: Technical interviews begin
    - Group: Continue working on final report, presentation, demo, and documentation.
    - Erik: Contribute to necessary sections of the final report, presentation, demo, and documentation.
    - Jerome: Contribute to necessary sections of the final report, presentation, demo, and documentation.
    - Maxwell: Contribute to necessary sections of the final report, presentation, demo, and documentation.
- December 10 - December 16
    - Deadline: December 14: All code complete, final presentation and demo, final report and support documentation deadline
    - Group: Present presentation and demo, turn in final report and supporting documentation.
    - Erik: Polish final report. Double check presentation and supporting documentation.
    - Jerome: Polish presentation. Double check report and supporting documentation.
    - Maxwell: Polish supporting documentation. Double check report and presentation.