Artifact for Sprint 4
Team: Calculatrons

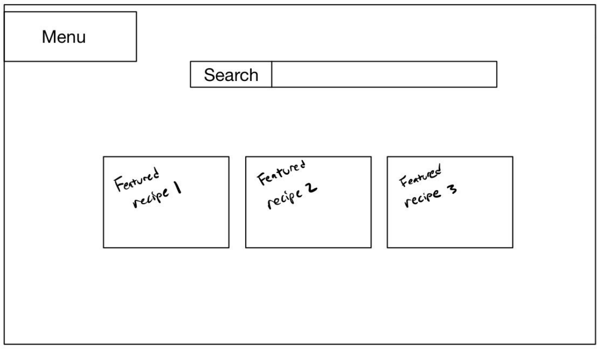
Features (user stories) to Implement in Next Sprint:

- Food Data:
(PP_recipes.csv, PP_users.csv, RAW_interactions.csv, RAW_recipes.csv, ingr_map.pkl, interactions_test.csv, interactions_train.csv, interactions_validation.csv)

- **Feature 1.** As a user I want to find a specific recipe that I want, using a search function.
- **Feature 2.** As a user I want recipes that are popular, or highest rated shown to me as I enter, so as to hopefully find one of interest.
- **Feature 3.** As a user, I want the information for the recipe given, such as calories, etc.
- **Feature 4.** As a user, I want a rating system that takes into account what people have thought of the recipes that they have tried.
- **Feature 5.** As a user, I want the ability to read comments on how the recipe came out, or how it might be made better by other users who would like to comment on it.
- **Feature 6.** As a user, I want to be able to save certain recipes for later.
- **Feature 7.** As a user, I want to see the recipes that I saved.
- **Feature 8.** As a user, I want to see the ingredients used in a recipe.

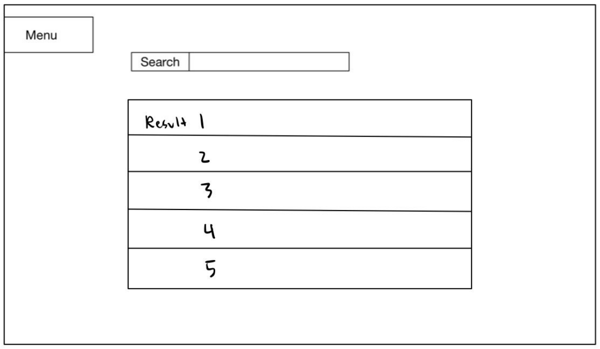
GUI Design for Insert/Update/Delete:

- Home Page: This is the page the user sees first



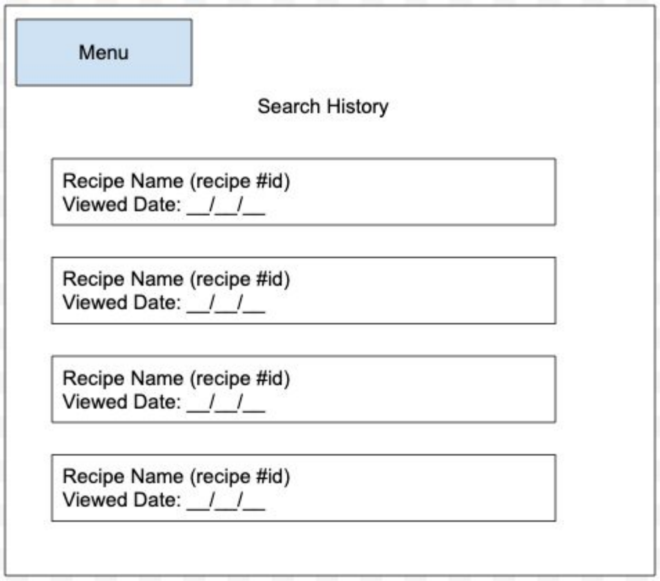
Home Page

- Results Page: When the user enters a search to find a particular set of items, those searched results will appear in this format.



Results

- Search History: All recipes that the particular user has searched for in the past will appear in this format.



Search History

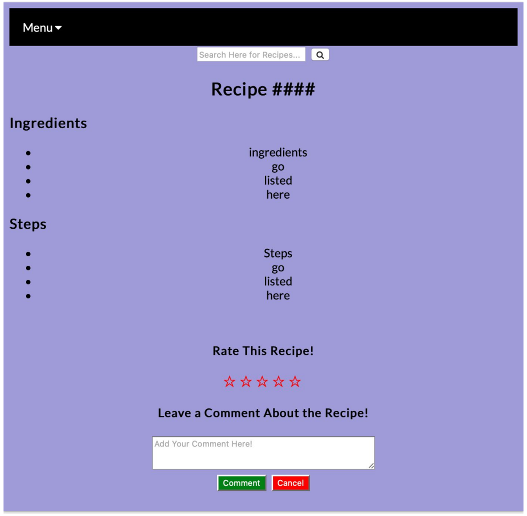
Recipe Name (recipe #id)
Viewed Date: __/__/__

Recipe Name (recipe #id)
Viewed Date: __/__/__

Recipe Name (recipe #id)
Viewed Date: __/__/__

Recipe Name (recipe #id)
Viewed Date: __/__/__

- Recipe Page: Each recipe has its own page and that is where specific information about that particular recipe can be viewed by the user.



Recipe ####

Ingredients

- ingredients
- go
- listed
- here

Steps

- Steps
- go
- listed
- here

Rate This Recipe!

☆☆☆☆☆

Leave a Comment About the Recipe!

Add Your Comment Here!

TEST CASES

- **Feature 1 Test Cases:** As a user I want to find a specific recipe that I want, using a search function.
    - **Test case 1**: as a user, I should be able to easily locate a button that indicates searching and have the ability to type in an input.
        - Correct Output: The website displays a search button with either the label, or a symbol that can be used to type in and search a keyword or words.
    - **Test case 2**: as a user, I enter the information and search to find important information relating to the words typed.
        - Correct Output: The website accepts the input entered by the user, and then displays the proper recipes that are in the .csv file used.
- **Feature 2 Test Cases:** As a user I want recipes that are popular, or highest rated shown to me as I enter, so as to hopefully find one of interest.
    - **Test Case 1:** As a user, I want my default search results to show the highest rated recipes first.
        - Correct Output: The website displays recipes with ratings in descending order by default.

- **Feature 3 Test Cases:** As a user, I want the information for the recipe given, such as calories, etc.
    - **Test Case 1:** As a user, I should be able to easily see all the nutritional information listed in each recipe with the corresponding values per serving
        - Correct Output: I should see an organized table with one column that lists each category of nutritional information (protein, carbohydrates, calories, etc.) and the second column lists the corresponding amount for each category.
- **Feature 4 Test Cases:** As a user, I want a rating system that takes into account what people have thought of the recipes that they have tried.
    - **Test Case 1**: As a user, I want to be able to see a number/rating/star highlight based on what people have thought of the recipe.
        - Correct Output: In each recipe image, the colored star ratings should be seen every time each recipe is seen in the search results and after clicking on the selected recipe when the user wants to get more information.
- **Feature 5 Test Cases:** As a user, I want the ability to read comments on how the recipe came out, or how it might be made better by other users who would like to comment on it.
    - **Test Case 1:** As a user, I should be able to find a list of comments below the description of each recipe.
        - Correct Output: A list of comments for each recipe should be visible when I scroll down past the listed recipe. The list contains the commenters' names, their message, and any other identifying factors.
    - **Test Case 2:** As a user, I should be able to add a comment of my own for a recipe that I feel a certain way about.
        - Correct Output: After pressing the "comment" button adjacent to the title and image of the selected recipe, a message box opens up. I type in my thoughts on the recipe and press the "post" button adjacent to it. The message is now inputted to this recipe's list of comments and appears alongside the rest.
- **Feature 6 Test Cases:** As a user, I want to be able to save certain recipes for later.
    - **Test Case 1:** As a user, I press the "save" button next to the image or title of the selected recipe, which should save the recipe.
        - Correct Output: The recipe's id is downloaded and available for use locally.

- **Feature 7 Test Cases:** As a user, I want to see the recipes that I saved.
    - **Test Case 1:** As a user, I should be able to find a list of recipes that were saved by me.
        - Correct Output: A list of recipes should be displayed that were saved by user.
    - **Test Case 2:** As a user, I should be able to add/remove recipes from the saved list.
        - Correct Output: New recipes that were added should be visible and deleted recipes should no longer be visible.
- **Feature 8 Test Cases:** As a user, I want to see the ingredients used in a recipe.
    - **Test Case 1:** As a user, I should be able to see every ingredient and nutrition information listed for each recipe.
        - Correct Output: A table of ingredients is listed in an organized matter.

COMPLETED:

- Created a local client/server application using NodeJS and JavaScript that sends requests to the server and returns a message. (finished by Abel Theodros)
    - Basic dummy front-end using HTML and CSS (Abel Theodros)
- Created a basic README.md file on Github. (finished by Michelle Dozal)
- Created an overall project proposal that lists our potential features and objectives of our project/application (finished by Abel Theodros, Michelle Dozal, Kinjal Mugatwala, Terry Jung, and Steven Joseph)
- Front-end search bar & navigation layout created using HTML and CSS (Michelle Dozal)
    - Search button that connects search bar to back-end (Michelle Dozal, Abel Theodros)
    - Back-end request for csv files, returns dummy message (Abel Theodros)
- Added .csv files to google drive as a backup (Kinjal Mugatwala)
- Server.js → organized csv files into an array and this array
- Parsed a particular csv file and formatted it into an Array object to make it easier to access certain data values (Kinjal Mugatwala, Terry Jung, Steven Joseph)
- Created recipeviewer.html frontend as the page where a specific recipe is viewed (Michelle Dozal)
- Frontend Star Rating System so the user can rate the recipe (Michelle Dozal)
- Frontend Comment Section/Submit & Cancel button to Post comments (Michelle Dozal)
- Created the savedrecipes.html,searchhistory.html,top_recipes.html,under100cal.html (Kinjal Mugatwala)
- Created a drop-down menu that has the list of other pages the user can visit (Kinjal Mugatwala)
- (October 28) - Using Python and Flask for the backend instead of NodeJS as NodeJS was becoming more difficult to incorporate
    - Files created:
        - Server-side = server.py (utilizing libraries from Flask)
        - Client-side = index.html (contains html, css, and js implementation), results.html (used to display the search results for now)
- Created a parser() function using the csv library that reads in each value using the reader() function and tries to separate the values based on the delimiter (Terry Jung, Steven Joseph)
- Passing in values that user inputted in the search bar to server.py to create some output, searched input value can be seen in the url (Kinjal Mugatwala)

TO DO LIST FOR THE NEXT SPRINT:

- Search operation for 6 different categories in the data (late Sprint-2, due 10/30)
    - Recipe id, ingredient id, cooking time, number of steps, recipe name, rating value
- Input and update selected recipes into savedrecipes.html (late Sprint-3, due TBD)
- Remove selected recipes from savedreciepes.html (late Sprint-3, due TBD)
- Back-end program to overwrite on the existing CSV
- Acceptance criteria: The method successfully writes on an existing csv with the new values.
- Figure out how to organize and display the search results (figure out which GUI elements will be involved)
- Streamline when necessary
- Add more to the front-end to make it nicer/organize the search results
- Figure out how to save searched results and access search history (backup)
    - Add recipes that were searched in past to the 'Search History' page
- Ability to see comments that where posted on the page
- Have the rating stay with the recipe and create an "Avg" rating.
- Figure out which analytics we want to display and explore - need a total of 6