Pantry Pal
Design Document

By: Jared Bauta, Eric DeAngelis,
Nick Hutchinson, William Jacobs,
David Liotta, David Smits, Gene Zaleski

2/8/19

Senior Project Spring

Table of Contents 1

High Level Design Description

The purpose of the project is to aid aspiring chefs by using ingredients from their kitchen to find new recipes to either develop new skills or simply help them decide what to have for dinner.

Users will come to the website and immediately be able to enter ingredients into a custom search engine that will return a number of recipes which can all be made with the given ingredients. This may result in a large number of recipes, so the user will have the ability to either sort or filter out the results based on a number of different parameters. They can sort by nutritional value, cost, difficulty, or ratings. Filters can restrict the recipes shown based allergy friendliness, vegan friendliness, or meal period(breakfast, lunch, dinner).

Additional features will be available to users that choose to create a free account. Google Oauth will be used for the user login. They will be able to save items in their pantry so they won't need to re-enter all of their ingredients every time they sign in. Other basic information such as allergies will be saved to automatically filter out recipes in violation of that allergy. They will also be able to like/dislike and leave a review on recipes they try for future users to see. One of the more compelling features for users with an account is that after they cook a few recipes they will begin to form a “taste profile” that will be able to intelligently predict a recipe that the user will likely enjoy and then recommend this recipe.

All of the search functionality will be powered by the Spoonacular API. This API has a significant number of endpoints available for use. We will make use of the ability to search recipes, search recipes by ingredient, search recipes by nutritional value, and many others. There is an endpoint for every one of our specific features.

The more social aspects of the website such as the like and comment system will be stored on our own database. The Spoonacular API provides an ID for every recipe, so our database will use that to identify which recipe gets paired with which comments and reviews.

Problem Solving Approaches

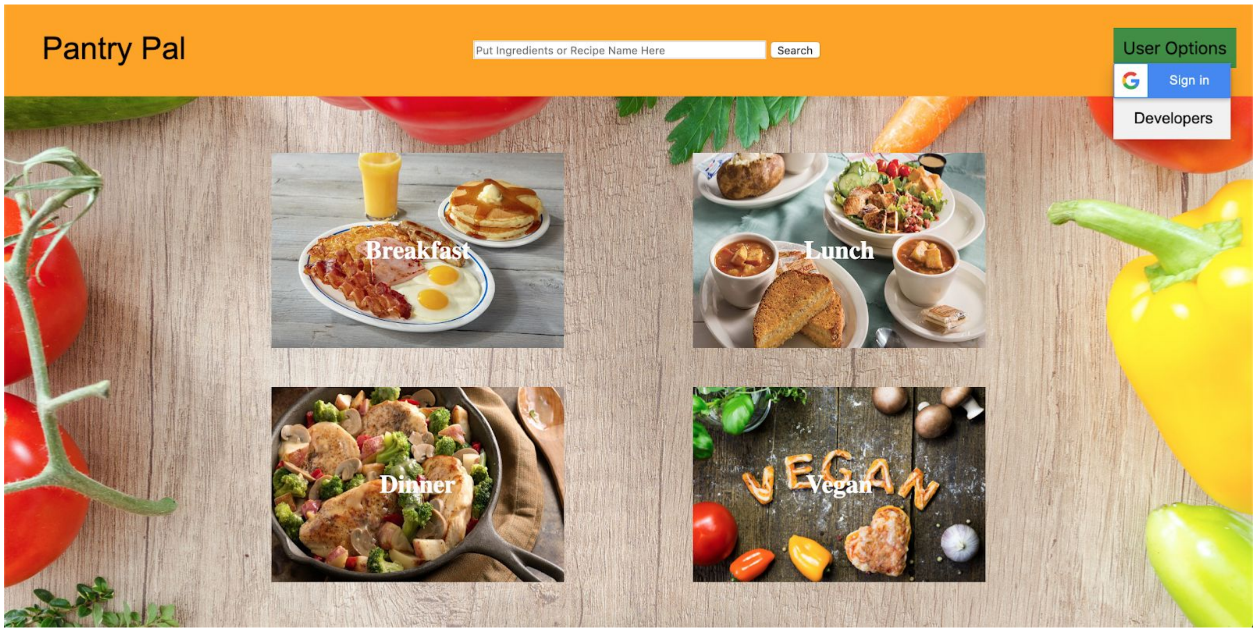
One of the biggest problems facing this project was what the user would be able to do within the website itself. It started off rather simple with the idea to save recipes and store food in a “virtual pantry” but the scope was too limited for the overall project. This was followed with many new ideas involving a comment system, a mile radius check to see what stores had the cheapest good for a recipe, review system, and filters for such things as allergies, vegan, etc. After this now our scope of our project became too large and we ultimately had to trim some of the more unrealistic goals off. We decided against the compare prices check for food around you system as it over complicated the project. However, the idea of filters,comments under the recipes,and reviews by other users was kept in as it creates a much more user interactive website then before.

Another issue faced was that of picking the correct API that would benefit us the most during this project. Overall we looked at around 15 different API’s all with their unique strength and weaknesses. Many API’s lack the total package we were looking for but Spoonacular API was the closest we found to what we envisioned what our website could look like and had many useful endpoints that we could use in our project.

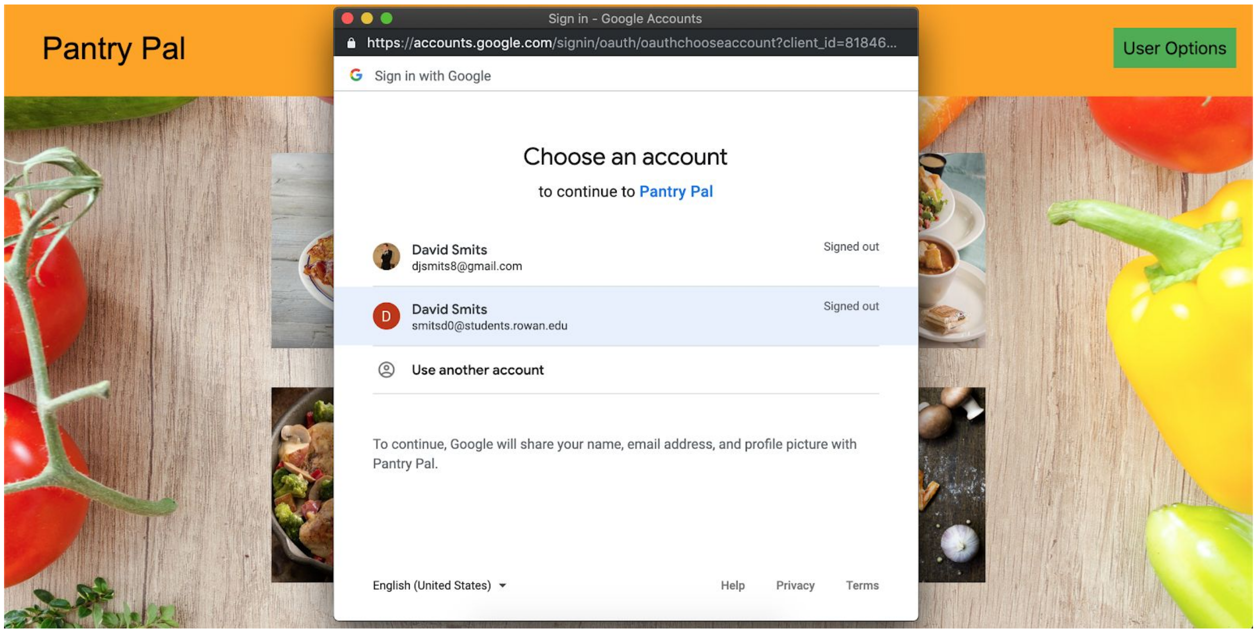
Also, the use of PHP is a spike because some of our members had no experience in using it as opposed to a few members had used it before in building a website. We decided on using it because it would be easy to integrate into our project and produced visually appealing websites. Members unfamiliar with PHP did their own research into how it worked and in understanding that how it wou0ld be integrated into the project.

Screens

Home Page: Guests can access the home page and the recipe results page. Signing in grants the user access to a profile page, inventory page, and allergy specification.



Login Page: Sign into your Google account using Google Oauth.


Pantry Pal
User Options
Sign in - Google Accounts
https://accounts.google.com/signin/oauth/oauthchooseaccount?client_id=81846...
Sign in with Google

Choose an account

to continue to Pantry Pal

David Smits
djsmits8@gmail.com
Signed out

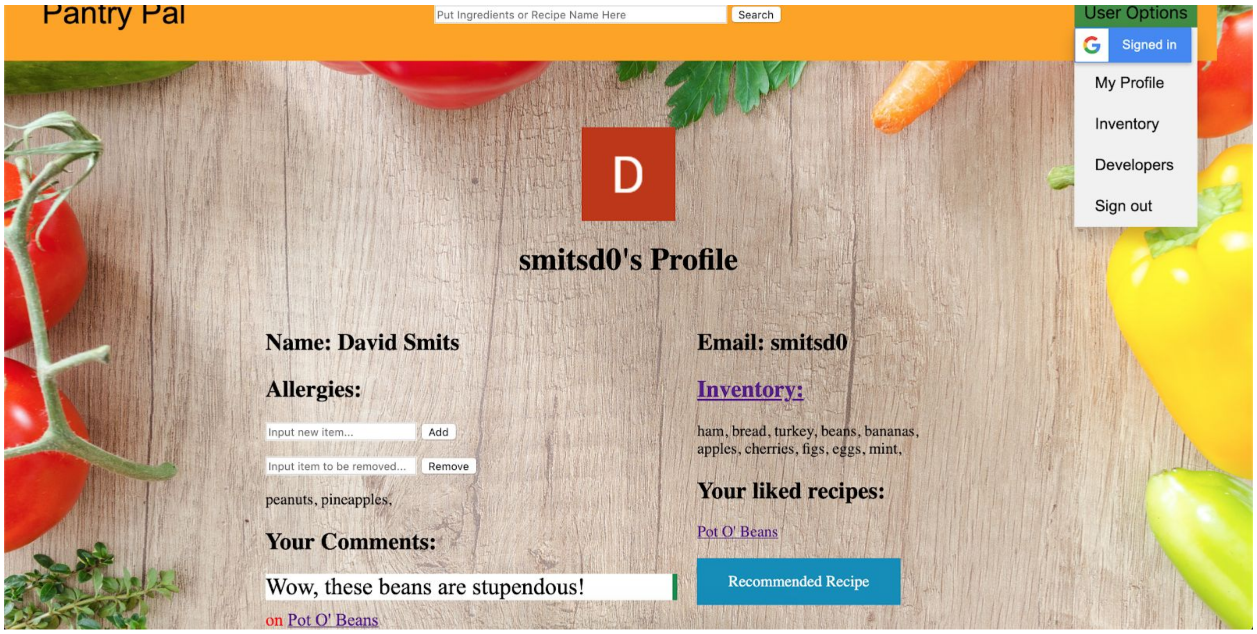
David Smits
smitsd0@students.rowan.edu
Signed out

Use another account

To continue, Google will share your name, email address, and profile picture with Pantry Pal.

English (United States) Help Privacy Terms

__Profile Page:__ Created when signed in. Includes name, email, allergies, your inventory, your comments, and liked recipes.



smitsd0's Profile

Name: David Smits

Email: smitsd0

Allergies:

Input new item...

Input item to be removed...

peanuts, pineapples,

Inventory:

ham, bread, turkey, beans, bananas, apples, cherries, figs, eggs, mint,

Your liked recipes:

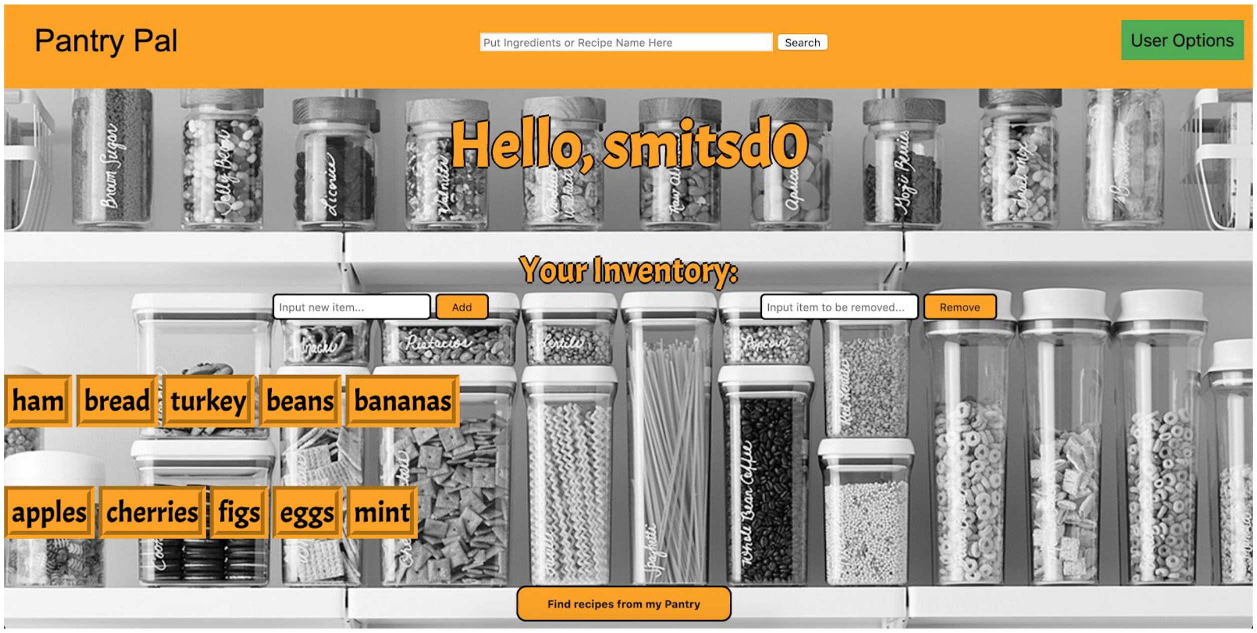
Pot O' Beans

Your Comments:

Wow, these beans are stupendous!

on Pot O' Beans

__Inventory Page:__ Gave the user the ability to add and remove ingredients to their inventory and search for recipes by what they have in their pantry.


Pantry Pal
Put Ingredients or Recipe Name Here
Search
User Options
Input new item...
Add
Input item to be removed...
Remove
Find recipes from my Pantry

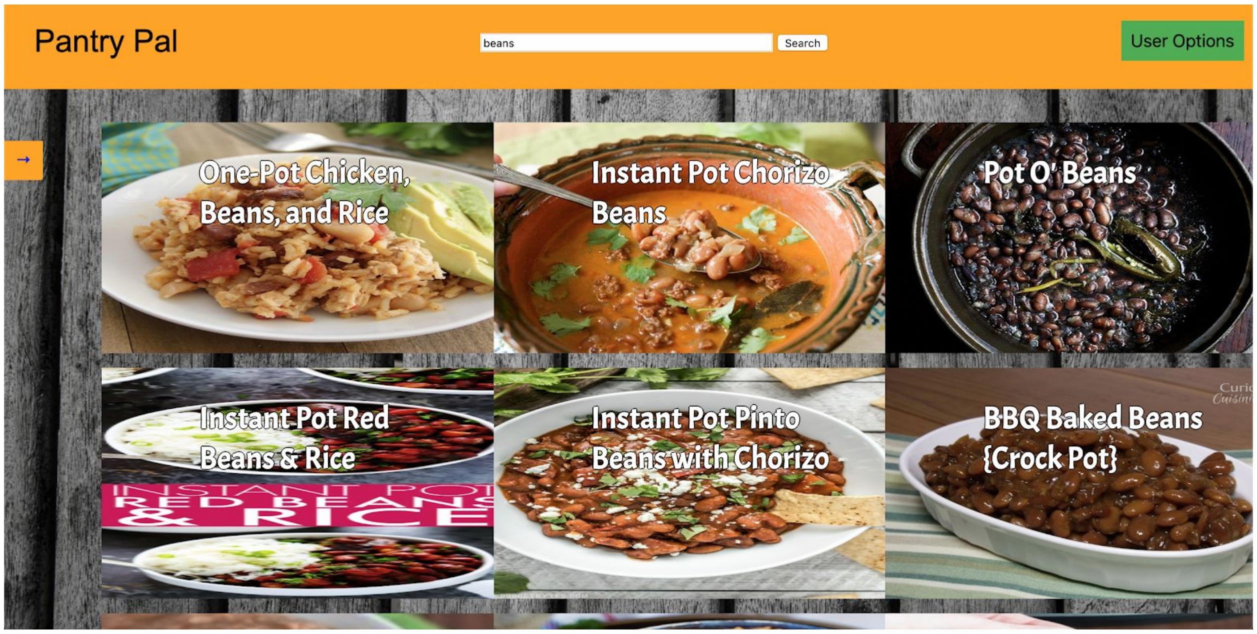
Hello, smitsd0

Your Inventory:

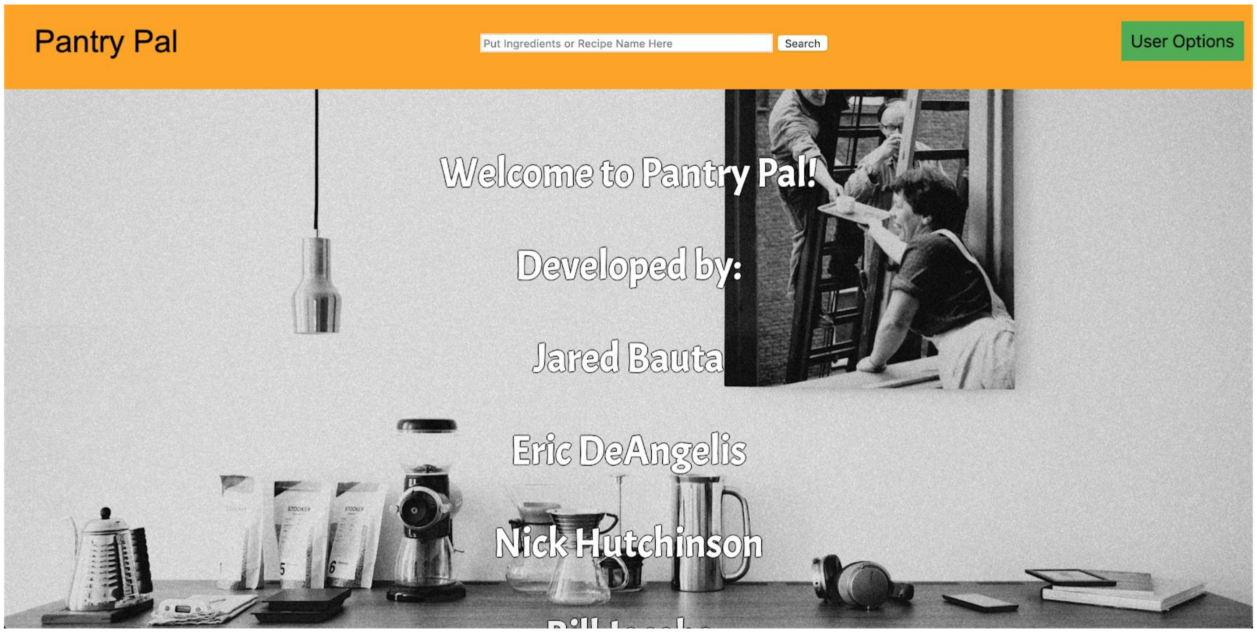
ham bread turkey beans bananas

apples cherries figs eggs mint

Results Page: Search randomly by not entering an ingredient, search by one ingredient, or search by multiple ingredients with comma separation. This results page outputs 60 recipes based on the inputted ingredients.


Pantry Pal
beans
Search
User Options
→
One-Pot Chicken, Beans, and Rice
Instant Pot Chorizo Beans
Pot O' Beans
Instant Pot Red Beans & Rice
Instant Pot Pinto Beans with Chorizo
BBQ Baked Beans {Crock Pot}

Developers Page: All members of the project group.



Welcome to Pantry Pal!

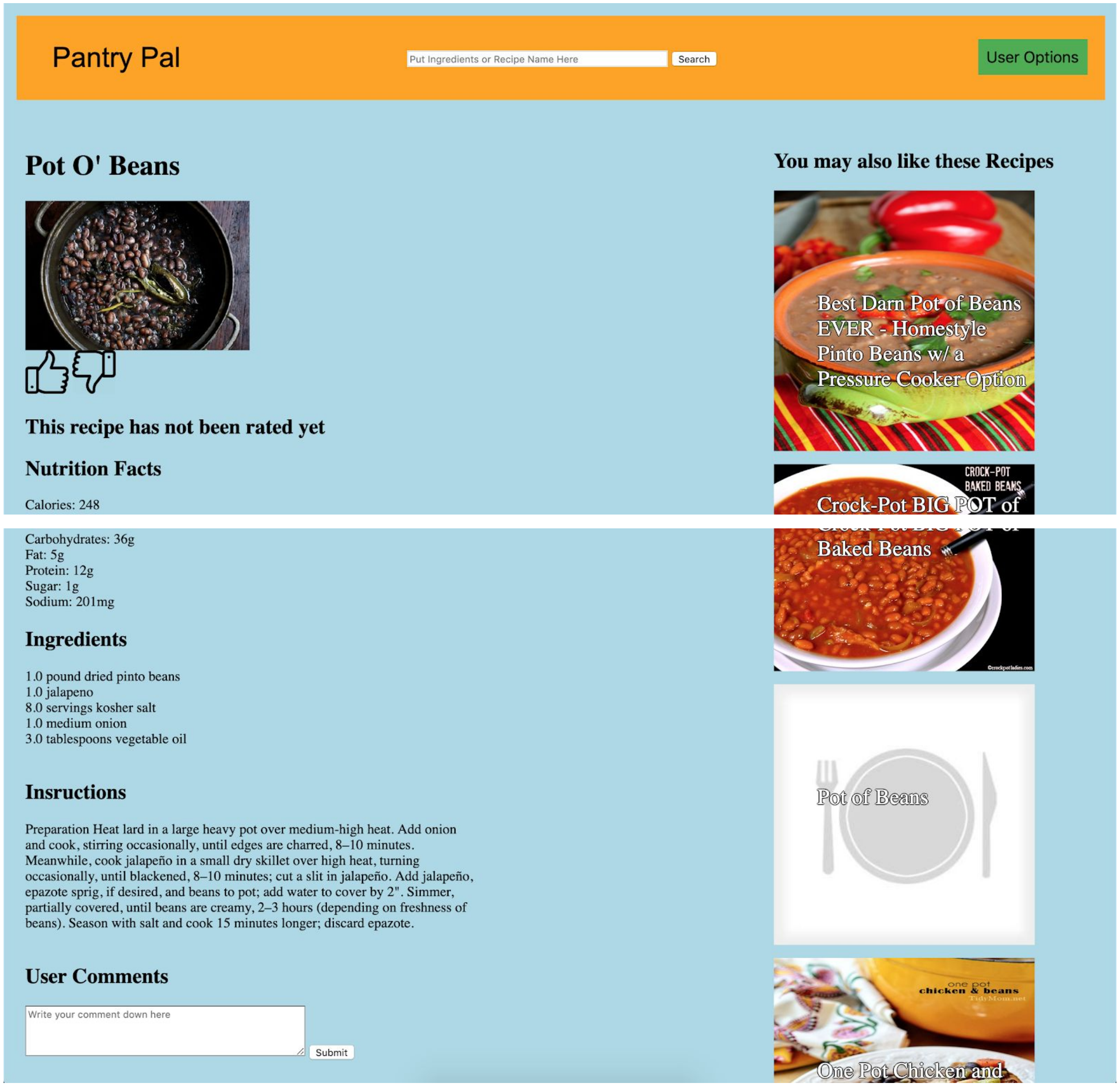
Developed by:

Jared Bauta

Eric DeAngelis

Nick Hutchinson

<u>Recipe Page:</u> When a recipe is clicked it redirects you to this page which includes the recipe rating, nutrition facts, ingredients, instructions, comments, and recommended recipes.



Pot O' Beans

This recipe has not been rated yet

Nutrition Facts

Calories: 248

Carbohydrates: 36g
Fat: 5g
Protein: 12g
Sugar: 1g
Sodium: 201mg

Ingredients

1.0 pound dried pinto beans
1.0 jalapeno
8.0 servings kosher salt
1.0 medium onion
3.0 tablespoons vegetable oil

Insructions

Preparation Heat lard in a large heavy pot over medium-high heat. Add onion and cook, stirring occasionally, until edges are charred, 8–10 minutes. Meanwhile, cook jalapeño in a small dry skillet over high heat, turning occasionally, until blackened, 8–10 minutes; cut a slit in jalapeño. Add jalapeño, epazote sprig, if desired, and beans to pot; add water to cover by 2". Simmer, partially covered, until beans are creamy, 2–3 hours (depending on freshness of beans). Season with salt and cook 15 minutes longer; discard epazote.

User Comments

Write your comment down here

Submit

You may also like these Recipes

Best Darn Pot of Beans EVER - Homestyle Pinto Beans w/ a Pressure Cooker Option

Crock-Pot BIG POT of Baked Beans

Pot of Beans

One Pot Chicken and

Screen Navigation

When a new user visits the website for the first time, they're greeted with the sites' Home page. Pantry Pal will be able to be used as both a registered user, as well as an anonymous visitor. The user will be presented with a message telling them about the advantages of logging into the site, however will note that the site is usable without an account. The user can select the Login option from the top right of the screen in the"navbar", which will be present on each screen of the site. While not logged in, the navbar will only contain a home button and a login option. While logged in, it will contain a home button and drop down menu in place of the login button with links to their profile, inventory/pantry items, and finally the option to logout. The Home Page of the website will consist of the navbar, dual purpose search bar(search by either ingredients or recipe name), and clickable tabs which will present the user with a more guided browsing approach focused on a food subcategory, recommendations (logged in only), popular items, or a certain meal.

If the user does choose to login, they will be prompted to login with their Google account information, developed using Google's OAuth system. If this is the user's first time on the site, they will be brought to the profile customization page where they can create a username, upload a profile picture, list their allergies, and input their pantry items. Otherwise, they will be redirected to the logged in version of the website with all of the drop down options present in the navbar.

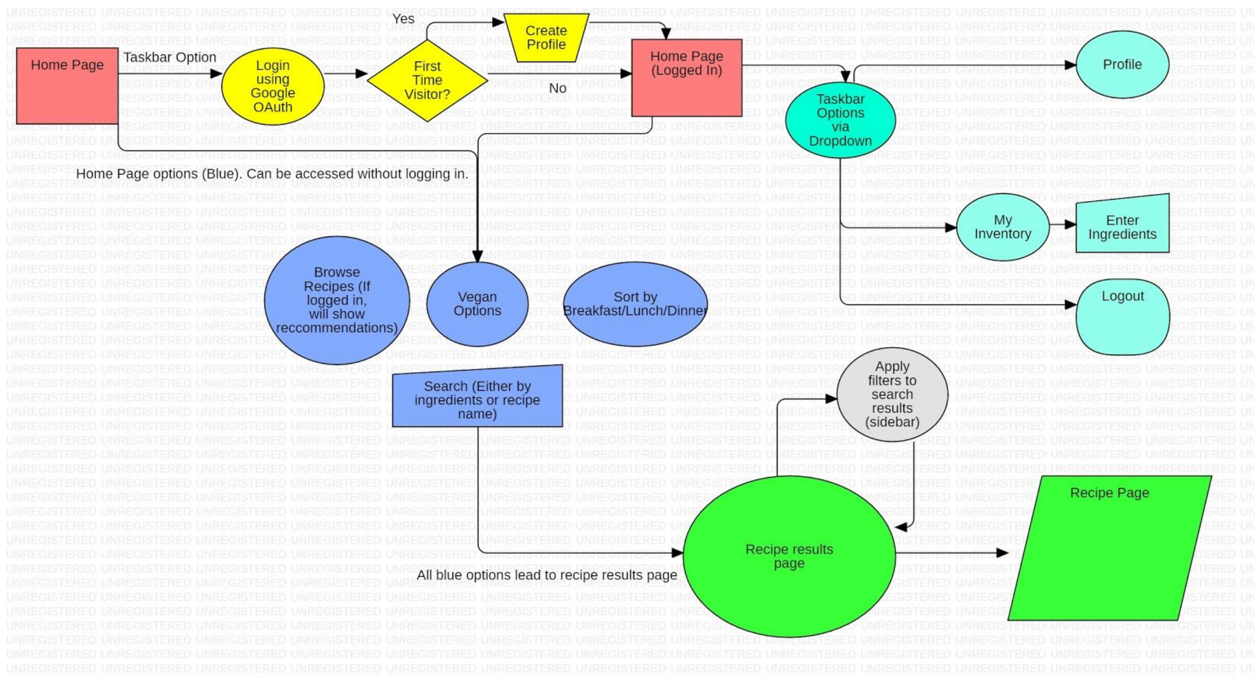
As stated above, the drop down menu will have redirects to the user's profile, Inventory, and a logout option. The user profile page is basically the same as when a user first signs up; It contains a profile picture, username, allergies, and inventory, all of which are editable. The allergies and inventory tab will show the user all of their items in list format if clicked.

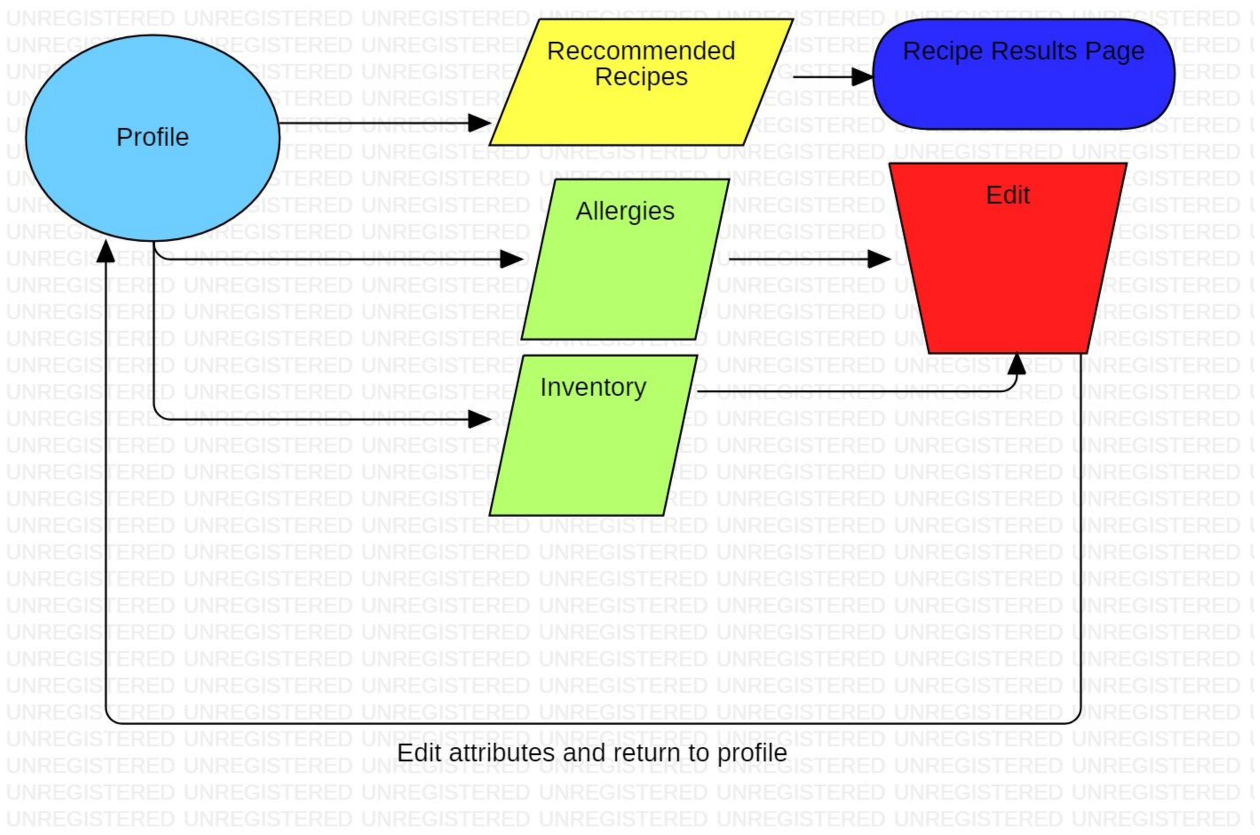
From the Home page, the user can access the websites search function to look up recipes. If the user wants to search for recipes by ingredients, they can enter various ingredients separated by commas and search. If the user wants to search by recipe name, they can do so from the same search bar. Both options will lead to the recipe results page. This page will display the search results in list fashion, with each recipe having a link to its own page. On the left side of the screen, there will be a column with checkbox controlled filters for your search results which can narrow down a large number of search results. Finally, the user can start a new search from the search bar that has carried over from the Home page.

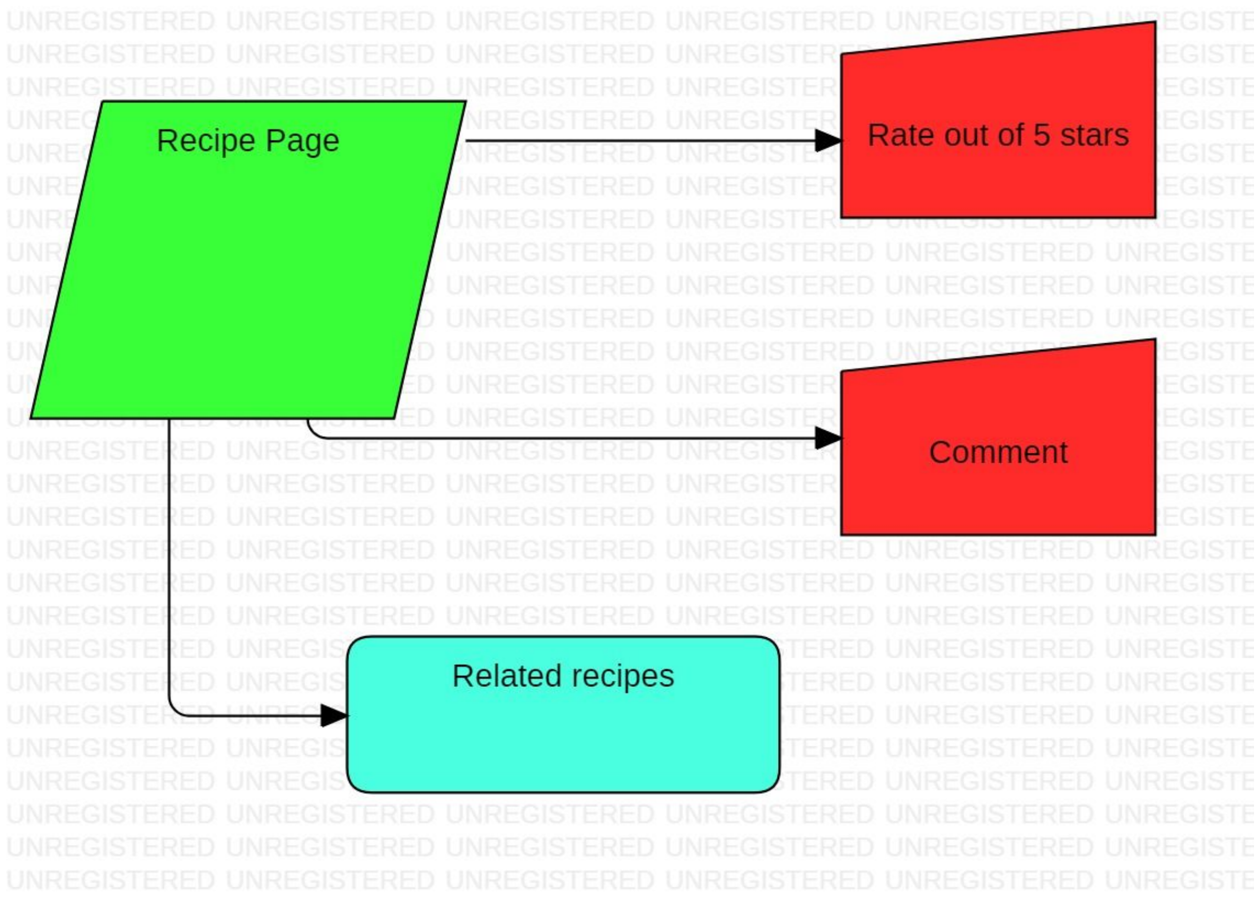
Once the user has found a recipe they are interested in, they can click the link to said recipe from the recipe results page. This page will display a picture of the food, recipe, community rating, and comments.

If the user decides to use the Home page tabs to aid their browsing, they will be redirected to specific recipe results page to give them a wide selection from foods to choose from. The “Vegan” tab will direct to a recipe results page showing popular vegan options. Finally, the “Breakfast/Lunch/Dinner” tab will direct the user to a meal specific results page in which they can choose the meal to browse from in the side filters tab.

Flow Diagrams


Home Page
Taskbar Option
Login using Google OAuth
First Time Visitor?
Yes
Create Profile
No
Home Page (Logged In)
Taskbar Options via Dropdown
Profile
My Inventory
Enter Ingredients
Logout
Home Page options (Blue). Can be accessed without logging in.
Browse Recipes (If logged in, will show reccommendations)
Vegan Options
Sort by Breakfast/Lunch/Dinner
Search (Either by ingredients or recipe name)
Apply filters to search results (sidebar)
Recipe results page
All blue options lead to recipe results page
Recipe Page
UNREGISTERED




Recipe Page
Rate out of 5 stars
Comment
Related recipes

Technology Stack

<u>Backend</u>

An AWS linux machine running Ubuntu will be used as our host server thanks to the Computer Science Department of Rowan University. We're using a MySQL database to store basic attributes about our users for example their username and liked recipes. MySQL was chosen due to its simplicity and popularity. We're also using Apache to deploy our website. PHP is going to be our programming language for transferring data between our database and the client side display.

<u>Frontend</u>

HTML and CSS will be necessary for designing and displaying the website on any modern web browser. To help with certain interactive features, we will use JavaScript. We will be using web server solution stack services such as XAMP, WAMP server, MAMP to test and develop our programs locally before we deploy them on the AWS server.

<u>API</u>

To have access to a large database of recipes to display to our user's we'll be using Spoonacular's Rapid API REST service. With Spoonacular's database at our disposal we'll be able to allow users to search for recipes, get information on ingredients, make ingredient substitutions, and even filter out certain ingredients from a search altogether. The reason we chose Spoonacular over other recipe databases is because of their robust endpoints that allow us to query data for recipes and accommodate our user's filters with very little backend processing on our part.

Authentication and Security

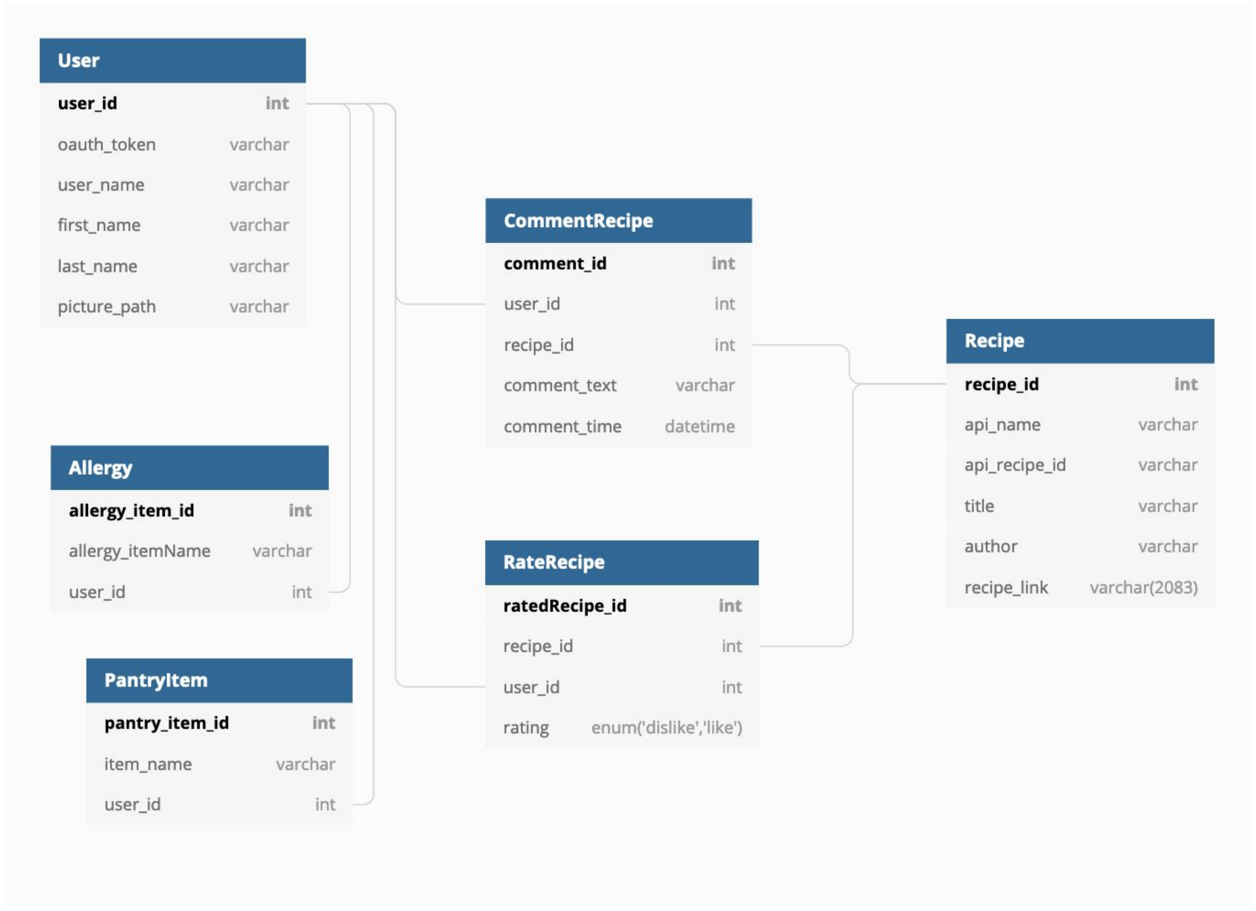
The website’s user authentication will be performed using Google’s OAuth to simplify our login system. The main reason for implementing OAuth is to avoid developing account management services for our web application. A good example of this complication would be handling if a user forgets their password.

Input and Output

Input: The user will be primarily be using a mouse and keyboard as an input device throughout the website interface. Users will input ingredients and recipe names into the main search bar to find food to make.

Output: The output will be a list of recipes the user could make based on their inputted ingredients.

Database Schema



The User table is used to hold information retrieved from google oAuth to be accessed later through an endpoint. The table has a unique primary key that is automatically incremented named user_id .

The primary key from the User table is used in other tables, Allergy and PantryItem, to establish a one to many relationship to link a user to the information in the table. Without a user, information would not be able to be stored in either table. The primary key of the User table is also used in CommentRecipe and RateRecipe to form a bridge between a user and a recipe. The dependency for the tables are different for these tables because these tables also have the primary key of another table, Recipe. This means for information to be stored in either table there needs to be a user and a recipe already in the database.

The Allergy table and PantryItem are similar in the information stored but different in how the information is used. The Allergy table is used to hold items that a user is allergic to, to filter recipes/results with these items. The PantryItem table is used to store items that a user has on hand and wishes to search for recipes with.

The CommentRecipe and RateRecipe tables are both tables that require a user_id and a recipe_id. Both tables are used to represent an action that a user can perform in regards to a recipe. The CommentRecipe table is used to store a comment that a user leaves on a specific recipe so that it can be later retrieved when the recipe is visited. The RateRecipe table is used to store the rating of a recipe, like or dislike, linked with a particular user. This table is used to count the total number of likes for a particular recipe, gather insights about a user's preferences and also to delete/add/change likes on a recipe.

The last table is the Recipe table and is similar to the User table in that the information is brought in from another api. In our case most, if not all, of the information brought in was from the spoonacular api containing recipes organized by ingredient. Since we wanted to utilize spoonaculars more optimized searching we did not store *normal* information associated with a recipe in this table such as ingredients, instructions, etc. Our table instead consisted of the api_name and api_recipe_id fields associated with the api where we retrieved the recipe from and the id used by that api to refer to the recipe. This was used to quickly search (spoonacular) the api for the exact recipe we wanted providing as little information as possible. The title, author, and recipe_link fields are used to decouple the database ever so slightly from an api because, if necessary, all information about a recipe stored in the database could be retrieved from the recipe_link.

Restful Endpoints

The endpoints are labeled in the form of REQUEST TYPE /endpoint/url/. Inputs to an endpoint are denoted by brackets {}. Additional parameters will be handled in through query strings and/or form data (for post requests).

USER
GET api/User/read_one.php?user_name={}
Returns user table corresponding to the user_name.

GET api/User/read_name.php?user_id={}
Returns user table corresponding to the user_id.

POST api/User/create.php
Creates a new user from the specified JSON, returns a verification message.
Example JSON
{"oauth_token":"google",
"user_name":"Johnsmith69",
"first_name":"John",
"last_name":"Smith",
"picture_path":"http://google.com/udsjnlsjdkrwo98erij"}

DELETE api/User/delete.php?user_id={}
Deletes a user within the database corresponding to the user_id.

RECIPE
GET api/Recipe/read_one.php?api_name={}&api_recipe_id={}
Returns the attributes corresponding to a recipe in JSON.
POST api/Recipe/create.php
Creates a new recipe from the specified JSON within the database, returns a verification message. Example JSON
{"api_name": "spoon",
"api_recipe_id": "93475920384",
"title": "5 Minute EGG Salad",
"author": "Food Network",
"recipe_link": "foodnetwork.com/eggsalad"}

RATE RECIPE
GET api/RateRecipe/liked.php?user_id={}
Returns attributes from the RateRecipe table corresponding to a user_id as a JSON.

GET api/RateRecipe/likes.php?recipe_id={}
Returns attributes from the RateRecipe table that correspond to a recipe_id as a JSON.

GET api/RateRecipe/read_one.php?recipe_id={}&user_id={}
Returns RateRecipe instance for one corresponding recipe_id and user_id as a JSON.

UPDATE api/RateRecipe/update.php?user_id={}&recipe_id={}&rating={}

Updates the rating corresponding to a recipe_id and a user_id where the rating is one of the enum types ‘like’ or ‘dislike’.

POST api/RateRecipe/create.php
Adds a like or dislike to the corresponding to the data from a passed JSON.
JSON example
{"recipe_id" : "8",
"user_id" : "1",
"rating" : "like"}

DELETE api/RateRecipe/delete.php?recipe_id={}&user_id={}
Deletes a RateRecipe column corresponding to the recipe_id and user_id, returns a JSON encoded confirmation message.

COMMENT RECIPE
GET api/CommentRecipe/recipe_page.php?recipe_id={}
Returns data within the commentRecipe table corresponding to a recipe_id as a JSON.

GET api/CommentRecipe/profile_page.php?user_id={}
Returns data within the commentRecipe table corresponding to a user_id as a JSON.

POST api/CommentRecipe/create.php
Adds a comment to the corresponding data within the JSON example.

PANTRY ITEM
GET api/PantryItem/read_one.php?user_id={}
Returns data from the pantryItem corresponding to a user_id as a JSON.

POST api/pantryItem/create.php
Adds an ingredient or item_name for corresponding JSON data, example
{"item_name": "veal",
"user_id": "4"}

DELETE api/PantryItem/delete?item_name={}&user_id={}

Deletes a pantry item corresponding to a item_name for a user_id, returns a confirmation message.

ALLERGY
GET api/Allergy/userAllergy.php?user_id={}
Returns data within the allergy table corresponding to a user_id as a JSON.

POST api/Allergy/create.php
Creates an allergy for a corresponding to JSON data, example
{"allergy_itemName" : "veal",
"user_id" : "3"}

DELETE api/Allergy/delete.php?allergy_itemName={}&user_id={}
Deletes an allergy corresponding to the allergy item and the user_id, returns a confirmation message in JSON format.

Backend TBD

- UPDATE requests must be made for all tables including "User, Recipe, Allergy, PantryItem, CommentRecipe, & RateRecipe ".
- Google Oauth needs to be integrated with the pantrypal database and Restful endpoints need to be planned. This means possible column additions and editing of current endpoints.
- Comment and Rating endpoints need separate queries to sort by username and sort by recipe to allow for the users to easily find recipes they comment on or like as well as update recipe pages with current conversations and ratings.
- Comments are also not stored in Eastern Standard Time and need to be adjusted.

Frontend TBD

- Add section tabs to homepage (breakfast, lunch, dinner, vegan, etc.)
- Move search bar to nav bar so it is accessible on all pages of the site
- Checking search filters redirects to a new page (results page)
- Get likes/dislikes percentage working (recipe page)
- Get user comments set up (recipe page)
- Nutrition facts on recipe page
- Major redesign of Profile page
  - Retrieve user data (user name, email, inventory, allergies, liked recipes, recommendations based on likes?)
- Clean up style on all pages to look as good as possible