**1. What tech stack will you use for your final project? We recommend that you use React and Node for this project, however if you are extremely interested in becoming a Python developer you are welcome to use Python/Flask for this project.**

For this project the tech stack that we will use is a combination of React/Javascript for frontend, and Node/Express to facilitate the backend server.

Postgresql will facilitate the database.

**2. Is the front-end UI or the back-end going to be the focus of your project? Or are you going to make an evenly focused full-stack application?**

I believe this will be a project that evenly focuses on both front and back-end, therefore an evenly focused full stack application.

**3. Will this be a website? A mobile app? Something else?**

Primarily this will be a mobile focussed application, but also using media queries will be available as a website.

**4. What goal will your project be designed to achieve?**

The initial goal is to create a dynamic full-stack project, with a clean React-based front-end and node express fully robust server side application.

The site is aimed at anyone who likes to cook food, so the goal is to achieve a light-weight, but powerful api-driven community based application that is a one stop shop for users and their curiosities for cooking with the ingredients that they have.

**5. What kind of users will visit your app? In other words, what is the demographic of your users?**

The demographic of users will be quite wide, anyone who has curiosity or fun with cooking, to people who will use it more pragmatically to find out what recipes are achievable with the ingredients they have at home. Other uses would be if cooking with guests that have allergies, we can deselect items to avoid and see what dishes can be made *without dairy* for example.

**6. What data do you plan on using? How are you planning on collecting your data? You may have not picked your actual API yet, which is fine, just outline what kind of data you would like it to contain. You are welcome to create your own API and populate it with data. If you are using a Python/Flask stack are required to create your own API.**

<https://spoonacular.com/food-api/pricing> // free with 150 calls per day

<https://www.themealdb.com/api.php> with paypal support we can filter recipes by multi-ingredient, which would be very useful.

**7. In brief, outline your approach to creating your project (knowing that you may not know everything in advance and that these details might change later). Answer questions like the ones below, but feel free to add more information:**

**a. What does your database schema look like?**

Users

* Users have recipes
* Friends
* comments
* favorited comments
* ingredients
* favorites

Recipes

* Recipes have ingredients
* Ingredients
* comments

Ingredients

* ingredients are in recipes
* ingredients are owned by users

Comments

* posted by users
* favorited by users

**b. What kinds of issues might you run into with your API? This is especially important if you are creating your own API, web scraping produces notoriously messy data.**

The initial potential issue I can identify is to have an effective free of charge API that can support filtered recipe search by multi-ingredients which will be the backbone of our “What's in the fridge” recipe recommendation system.

**c. Is there any sensitive information you need to secure?**

So far just the usual password encryption, using our Bcrypt library.

**d. What functionality will your app include?**

Users can add to their list of available ingredients, with nice interfacing.

Users can create shopping lists, and when shopping is complete, automatically add all items from the list into their items.

With the items in their available ingredients the user can search available recipes, with added specified parameters (e.g vegetarian, non vegetarian).

Users can pin recipes to other user’s profiles.

Users can favorite recipes, and comment on them for other users to see how their experience was, with possible tips etc.

**e. What will the user flow look like? f. What features make your site more than a CRUD app? What are your stretch goals?**

The homepage will offer a little demo without login,

To select a list for ingredients and then search available recipes.

This will serve as a taste, for when you register and login then you can save your fridge items and update regularly, for up to date and easy recipe searching.

Navbar: Home, My Fridge, Recipes, Community?, Login/Logout

My Fridge/Profile route is protected by middleware where you need to have an account, otherwise it will return a login/register component,

This will list the items in your fridge and available recipes.

This is also where users will find the shoplist feature.

The stretch goals here is to build a smooth running single site application with React and Node Express, with specific attention to approachable style and comfortable UX:

**Extra notes: Original Brainstorm**

Food and ingredients/Recipes app

using

<https://spoonacular.com/application/frontend/downloads/spoonacular-api-slides.pdf>

<https://github.com/forbesg/bbc-good-food-recipe-scraper>

Great for displaying React and components

Users can add to their profile, ingredients they like, ingredients they would like to avoid, ingredients they currently own (+ find a way to conveniently update this when purchased) and search potential recipes.

-shopping list idea

saving favorite recipes, and allowing recommendations / communication between users.

Users can create their own recipes, which will be available/populated into other users recommendations if their preferences align with the newly created recipe.

This allows for further integration of users, based on similar taste.

User accounts can also be private where their created recipes will not be shared,