Functionality Document

# Contents:

1. Introduction
2. Instructions
3. About the project
4. Team members
5. Programming languages

## Introduction:

Grocery App was designed to take the stress out of meal planning. Who has the time to browse the Internet or cook books for recipes and make shopping lists by hand these days? Grocery App was developed with the busy, always-on-the-go user in mind to make everyday meal planning as simple as 1-2-3.

Step 1: Go to the Recipes page and use the fun filters to view random recipes from our large database until you see one that catches your eye. Click on it for details!

Step 2: If you like it, add any of the ingredients to your shopping list by simply pressing the little green shopping cart button on the right.

Step 3: Review and print your shopping list. Voila! Your plan is done and you are ready to go!

You can plan a single meal or an entire week's worth! If you add ingredients from several different recipes, the meal titles will always display on your shopping list. Don't worry, we know you may want to add your own items to your list, that's what the Add button is for!

## Instructions:

* Users can choose to log in or use our app as a guest (For the purpose of this assignment, the 'login' isn't supported by a backend, and no true accounts exist, any combination of username/password will work. This was implemented to show our use of state)
* Users can proceed to the Recipe List page in the Nav bar and have a list of 10 random recipes rendered to them via [TheMealDB API](https://www.themealdb.com/).
* Once users select a recipe they will be directed in a new tab to the dynamic page for the recipe they selected. On this page users will find all the ingredients available to them, dish origins, allergy warnings, and a YouTube tutorial on how to make the recipe if they wish to follow along.
* Clicking on the shopping cart on any ingredient will add this item to the Shopping List
* Users can proceed back to the Recipe List page and continue adding items to the Shopping List, or proceed to the Shopping List page itself via Nav Bar or Shopping List button on the individual recipe page.
* On the shopping list:
  + Users have the option to add their own ingredient to their shopping list (Should they wish to use the printed PDF feature as a "Shopping List" to take with them to the store)
  + All items that have been added are displayed in a table with their corresponding meals in brackets.
  + Users have the ability to remove specific items by clicking the trash can or clear the entire list by clicking "Clear Shopping List"
  + Users can print the list to a PDF to print off and take with them to the store as a checklist so they do not forget any items.
* About Us:
  + This page contains a small paragraph about each team member who worked on this project.

## About the project:

Grocery App was built over the course of a week as part of the TECHCareers program in Edmonton, Alberta. To satisfy the core requirements of the project, which included clearly demonstrating competency in the use of HTML5, CSS, ES6 JavaScript, git, GitHub, React, React-Redux, and React-Router - Team Sharp (as we decided to call ourselves) decided to build a recipe app with a shopping list component.

The application was built with a mobile-first approach and is fully-responsive. In addition to the home page that describes the application and the project itself, there are also pages for the recipe list (where the user can use filters to randomly pull recipes from an API), the individual recipe (where the user can see details about the meal and has the option to add individual ingredients to their shopping list), the shopping list (where the user can add ingredients to, remove from, preview, or even print their list), the About Us (where the user can "meet" the team members), and a login page (that displays either a login/sign-in form or a page that shows who is logged in and gives them the options to either browse more recipes, visit their shopping list, or sign out).

### The Recipe List

This page fetches and dynamically displays recipe data from TheMealDB API. The user can filter for recipes with certain ingredients/dietary restrictions (i.e. 'chicken' or 'vegetarian' recipes) or they can ask for '10 Random Meals'. Each list item (recipe) features a thumbnail of the meal, the title of the dish, and a details button. The user can click on any of the elements to open the individual recipe in another window.

### The Individual Recipe Page

In addition to a larger photo of the dish, the individual recipe page also displays a quick description of the meal(About), any dietary restrictions, a list of the ingredients necessary to make it, and instructions on how to prepare the meal (in both written and embedded YouTube video format). The ingredients area has a numbered list with the name and quantity of each ingredient and a green shopping cart button that turns into a checkmark when the user clicks it. This indicates that the ingredient has been added to the shopping list. These items get stored in the Redux store to allow for global access of the state. There is also a shopping list button at the bottom of the list to ensure easy movement between the two pages for the user.

### The Shopping List

The shopping list pulls from the global state and displays each ingredient previously added to the cart. Each item shows the amount and name of the ingredient, followed by the title of the meal in brackets, and the trash button that allows the user to easily remove any entry from the list. There are 3 buttons at the bottom: The '+-Add' button expands a hidden section of the page, giving the user the option to add their own items to the list (ingredient, amount, and meal title). This small form doesn't have any mandatory fields, leaving the user completely in charge of what they want to add to their own lists. The 'Reset' button clears the form fields but will not affect the shopping list itself. The second shopping list button allows the user to print their list. When clicked, a print preview dialogue box opens. The third button clears the entire list.

## Team members:

### Krisztina Pap:

Krisztina is hard-at-work turning her passion for programming into a career. She has a strong background in training, project management, and tech. When she is not writing code, she enjoys reading fantasy novels, jogging, and playing board games with her family. Krisztina was nominated for the role of Team-Lead for this assignment. She worked primarily on Redux and Bug Solving.

### Lindsey Graham:

Lindsey spent the early years of his professional life in the Oil & Gas sector, doing jobs ranging from Purchasing, to onsite work as an Electrician. In early 2019 due to being unemployed, he picked up a few programming courses on Udemy to learn Python and ended up loving it to the point that he knew he wanted to change careers and take up the path of Software Development. Lindsey assumed the role of GitHub Manager on this project, handling all pull requests, merge conflicts, and primarily worked on React & API content.

### Tosin Olaniyi:

Tosin is a Pharmacist by Profession and Programmer by Passion. Tosin has keen interest in using technology and data science to improve the processes and systems used in the health sector. Tosin took on the big job of CSS for this project. He primarily worked on ensuring the styling of every page was consistent, bug fixes, and React Content.

## Programming languages used:

* React.js
* Redux
* React-Router
* CSS
* HTML