Capstone Cooking

# Team Members

* Calen Jones
* John Mitchell
* Nathaniel Maynard

# Abstract

Capstone Cooking is a web-based app that helps users track, manage, and maintain a collection of recipes and ingredients. Users will log in with their account and be greeted with an array of tools and previous recipes they’ve saved. Unit conversions, recipe lookup, ingredient management, and more all in one convenient place.

# Description

Cooking has always been an important part of home life, but people often struggle with the logistics surrounding it, including keeping up with and editing recipes, converting unit measurements and scales, and keeping up with what groceries they have and what they need. Capstone Cooking strives to make cooking smarter by managing these tasks for any home cook, from college students on a tight budget to parents of large families. This app will also help cooks reduce waste and use groceries more efficiently by tracking what ingredients they have and allowing them to plan meals ahead of time.

With an easy sign-in to the app, users are greeted with a screen that allows them to store grandma’s home-cooked recipes, meals that they’ve created or customized on their own, or as many AI-generated slop creations as their heart desires. The user can enter the ingredients and amounts used, cooking temperatures and times, equipment needed like pans and measuring cups, calorie estimate, and their own text descriptions needed to reproduce their recipe. If the user wants to write out their entire life story and the entire historical origin of the recipe, as many authors choose to do when posting online, they can do that within the application as well and even include a picture. If the user is unsure of how many calories the meal has, the app will even attempt to calculate it for them. They can also go back to these recipes later to make edits or show different scaled versions of them. Now they have easy reference to their favourites with the click of a button any time it’s needed. All recipes are saved to a library within the user’s account, and they can export individual recipes or folders to PDF format for offline use as well.

Users can organize their saved recipes with different folders and tags so that they are easier to find, and can browse through them using a filter search by recipe names, calorie amount, tags, and ingredients. If the user is unsure of what to cook that day, it can also pull a random recipe to try, especially helpful for indecisive users or cooking casuals who have other concerns. There is even a whitelist feature for when users crave a specific ingredient, a blacklist feature for days when something isn’t wanted or a guest has an allergen, a spice level indicator for when the user is feeling spicy, and a filter for specific cooking methods for when users want to cook in a way that fits their day that day. The app can also be used to scale an individual recipe up to feed a family, or to scale a large recipe down for just one serving. If the recipe is from overseas and uses different measurements or units, that isn’t an issue either. Users can simply open the recipe, specify their unit of choice, and be greeted with measurements they have the tools to work with.

The app will also allow users to plan meals ahead of time, where they can create a batch of one or multiple recipes and choose what days to eat them. From this schedule, the app can automatically create a grocery list for them, where they can select what they do and don’t need, as well as get a rough price breakdown of the ingredients desired. This list also allows the app to keep an internal log of ingredients gotten and deducts them when recipes are selected and used. If an ingredient is marked unavailable on the shopping list or a user wants to make do without, the app will offer substitutions on certain ingredients when possible, saving users the hassle if they choose. While in the kitchen, the user will be able to put the application into reading mode, which will hide most of the editing features to improve readability, and the webpage will keep their screen active to stop their device from sleeping due to inactivity.

When cooking, the app offers support along the way, too. Opening a recipe and choosing to begin cooking that meal enters instruction mode, where the app will offer the user step-by-step guidance on how to prepare the dish (assuming this is included in the recipe). In this mode, the app also prevents devices from going into sleep mode, which saves users the hassle of reopening the device every few minutes or getting their screen dirty in the process.

While there are many different recipe apps and websites available, Capstone Cooking aims to be more helpful by handling some of the more annoying logistics parts of cooking. With the app handling things like meal planning, portion scaling, and shopping lists, it allows the user to spend more time in the kitchen cooking and perfecting their own recipes stored in one place.

# Features

## Essentials

* Create/edit/delete recipes
* User accounts
* Search recipes
* Recipe organisation, tags, folders
* Scale recipes
* Add photos of recipes
* Conversion between imperial and metric, and also between different measurements
* Filtered recipe searches
  + Blacklisting of ingredients
  + Requiring ingredients
  + Cooking method
  + Spice level (Chinese takeout spicy metre)
* Export recipes to PDF

## Optionals

* Meal planning
* Ingredient amount tracking
* Shopping list generation
* Equipment needed (pans, measuring cups, tinfoil, etc.)

## Nice-To-Haves

* Random recipe selection based on the ingredients currently available or based on user favourites
* Substitution suggestions
* Offline recipe access
* Price estimation of meals/ingredients
* Calorie estimation
* Instruction mode - screen will show step-by-step instructions
* Prevent device sleep mode while instructions are opened

# Technology Stack

* Platform: Web Application
* IDE: Visual Studio Code
* Languages: TypeScript, CSS, SQL
* 3rd Party Frameworks: React
* Communication Software: primarily Discord
* AI: Gemini free, avoiding use for implementations unless we get stuck

# Server Information

* Storing accounts and recipes with Supabase and Postgres SQL

# Team Member Backgrounds

Nathan has experience building web applications and familiarity with both React and TypeScript, so he will be working on the back-end of the project, as well as helping on the frontend and database when needed. John has learned TypeScript, works in game development, and is skilled and comfortable in design, so he will be working on the frontend of the project. Calen has moderate familiarity with SQL and has worked on designing a few other applications, so she will be working on setting up and maintaining the database in Supabase and helping out on design when needed.

# Dependencies, Limitations, & Risks

The login feature, PDF export, as well as nutrition information depend on the below APIs or packages:

* Login: <https://www.npmjs.com/package/react-login-page>
* Nutrition API: <https://api-ninjas.com/api/nutrition>
* jsPDF: <https://github.com/parallax/jsPDF>

The biggest risk for us in the project is that none of us has directly worked with Supabase in setting up a database; that said, Supabase was used on the last web application project Calen had worked on with no issues, so we still have the ability to reference old code, and there are copious amounts of tutorials to be found on Supabase’s main site, Stack Overflow, YouTube, etc., so we’re confident we should be able to use it without much of a hitch. Another risk is that John has not worked on building websites/web apps before, so he has to learn CSS and use TS to build an interface; that should not present too much of an issue, though, because he is already skilled in design, knows some TS already, and Calen and Nathan are prepared to jump around and help out on the frontend if needed. Also, the project is going to be written primarily in TypeScript, and Calen is only familiar with JavaScript, so there is a need to learn that as well; this also shouldn’t come with much risk, though, since she will be primarily working in Supabase.

The only current dependency we have in the project is that Nathan will not be able to begin writing the code for the backend work until the database is established, but the current timeline involves piecemealing the schema, so he should be able to start working on his code fairly soon.

# Timeline

Week 1-2

**Nathan**: Getting familiar with Supabase and the syntax used in connecting to it and

**John**: Start drawing skeletons and/or working in Figma to develop a rough look for how each page will look.

**Calen**: Focus on developing an ERD for how the database should be structured; begin learning TypeScript to assist in other areas of the project.

**All**: Develop a UML diagram and/or discuss the flow of the

Week 3-4

**Nathan**: Begin writing backend code to query Supabase to create or validate user accounts/profiles, set up the login page component

**John**: Set and style the interface for the login screen and account profile

**Calen**: Develop the database schema in Supabase, needing at least a user and/or account table already ready; confer with Nathan on DB needs, and help with building the interface for the login screen and account profile

Week 5-6

**Nathan**: Write backend code needed to create/edit/delete/organize recipes and images, and scaling/converting recipes

**John**: Develop and design the interface for the pages displaying and/or organizing recipes and photos

**Calen**: Verify tables needed to display/update recipes and photos are set up and test for functionality; confer with Nathan on DB needs, and work on the frontend getting recipe screens set up and styled and/or backend for recipes scaling and conversion (if needed)

Week 7-8

**Nathan**: Write backend code needed to enable search of recipes with and without filters

**John**: Develop and design the interface related to the search function (TBD on how it’s implemented)

**Calen**: Confer with Nathan on any DB needs for search function or other functions; work on design on the frontend for any pages still needing styling and assist with developing any other pages needed

Week 9-10

**Nathan**: Write backend code needed to export recipes to PDF

**John**: Develop and design the interface for PDF export and saved collection within the application

**Calen**: Confer with Nathan for any DB needs for export, test out functionality on all features, and work on the frontend where needed

Week 11-12

**Nathan**: Write backend to set up meal planning function

**John**: Develop and design the interface meal planning page/section; finalize any interface and styling changes

**Calen**: Confer with Nathan for any last DB needs, assist in meal planning feature (if needed), and finalize and smooth any design needs and flaws, respectively, on the front end

**All**: Testing out the entire application (both computer and phone) to ensure visual appeal and functionality of all features