## **I. Overview**

This web app helps users manage their ingredients and intuitively discover recipes they can cook with what they have on hand. Core features include: Google authentication, creation and management of user-specific ingredient lists, ingredient search via TheMealDB API, recipe search based on ingredient lists, saving favorite recipes, intuitive UI for managing lists and viewing recipe results

Live App: <https://meal-match-csce-606-78c84fa1ddb6.herokuapp.com>

Demonstration video on Youtube: <https://www.youtube.com/watch?v=DXb_NLKoTNQ>

## **II. Google Sign In**

* Click “Log in with Google.”
* Choose your Google account.
* If the app is in testing mode, ensure your Google account has been added as a test user (you will be informed should access be restricted).
* After signing in, you will be redirected to your Dashboard

## **III. Dashboard**

The Dashboard is your main control center. From here you can:

* Manage Ingredient Lists to create or edit/view your ingredient lists
* Manage Saved Recipes to view and remove your saved recipes
* Search Recipes to select an ingredient list and find recipes that can be cooked using those ingredients.

## **IV. Managing Ingredient Lists**

### **1. Create a New List**

* Click “Manage Ingredient Lists.”
* Click “Create new list.”
* A new list will appear with the default title “Untitled list.”
* Click “View/Edit” to open the list editor

### **2. Add Ingredients**

Inside a list:

* Click “Add Ingredients +” to open the ingredient search modal.
* Type an ingredient name (for example, “chicken” or onion”).
* Select items from the search results - they will appear as “chips” below the search box.
* Click “Save” when done.

*Note: you can also rename your list using the text box at the top.*

### **3. Edit or Delete a List**

* To rename a list, edit the title and click “Save.”
* To delete a list, click “Delete” next to it on the Manage Lists page.

## **V. Search for Recipes**

### **1. Choose Your Ingredient List**

* From the Dashboard, find the “Search Recipe” section.
* Select a list from the dropdown (e.g., “Taco Tuesday”).
* Click “Search Recipe.”

### **2. View Recipe Results**

You will see all matching recipes retrieved from TheMealDB API. Each recipe card shows:

* Recipe title and image
* Number and details of ingredients you are missing
* Details of ingredients you have
* “Like” button to open a description modal with full recipe details
* “Hate” button to intuitively skip to the next recipe waiting in line
* “Save Recipe” button to add current viewed recipe to your saved recipes list

### **3. Filter Recipes**

At the top of the recipe results page:

* Use the Filter by Missing Ingredients” option to limit recipes based on how many ingredients you do not have.
* Example: set “2” and click “Apply filter” button to only show recipes missing two or fewer ingredients.

## **VI. Saving Recipes**

### **1. Save a Recipe**

* On any recipe result or modal within the recipe search result page, click “Save Recipe.”
* The recipe will be stored under your account.

### **2. View Saved Recipes**

* Go to “Manage Saved Recipes” from the Dashboard.
* See all your saved recipes with thumbnails and details.
* Click “View Description” to open a modal with the full recipe.
* Click “Remove” to delete it from your saved list.

## **VII. Logging Out**

* Click “Log out” button at the top of Dashboard
* You will be signed out and returned to the login page

## **VIII. Testing and Deployment**

Automated Testing

* Unit tests (RSpec) cover:

- Core models (Ingredient, IngredientList, SavedRecipe, etc.) for validations, associations, and custom logic.

- Service objects like MealDbClient for API integration, caching, and error handling.

* Controller and Request specs validate full CRUD flows, redirects, flash messages, and access control.
* System (feature) specs using Capybara simulate real user interactions for:

- Dashboard UI navigation

- Ingredient list management

- Recipe search and display

- OAuth login and logout behaviors

Integration & Acceptance

* Cucumber-style acceptance tests mimicking end to end flows , ensuring:

- Authentication (Google OAuth2) works with registered test users.

- Ingredient selection and recipe search behave correctly across HTML and JSON formats.

- Frontend modals and JS fallbacks (no-JS forms) function as expected.

Deployment

* Ensure all migrations are included and reviewed.
* Include a data backfill plan for new production schema changes.
* Verify environment variables (THEMEALDB\_API\_KEY, etc. see the environment variables section of the project readme at <https://github.com/CSCE-606-Project-1/foo1/blob/main/README.md#environment-variables>) are set in staging and production.
* Deployment proceeds only after QA sign-off and all tests pass in CI.

## **IX. API Quick Reference**

1. Authentication

* GET /login → Render login page.
* DELETE /logout → Log out current user.
* GET /oauth/google\_oauth2/callback → Handle Google OAuth2 callback after successful login.

2. Dashboard

* GET /dashboard → Show user dashboard (after login).

3. Ingredient Lists

* GET /ingredient\_lists → View all ingredient lists.
* POST /ingredient\_lists → Create a new ingredient list.
* GET /ingredient\_lists/:id → View a specific ingredient list (and its ingredients).
* PUT /ingredient\_lists/:id or PATCH /ingredient\_lists/:id → Update an existing ingredient list.
* DELETE /ingredient\_lists/:id → Delete an ingredient list.
* GET /ingredient\_search?query=<query> → Search for ingredients by name (used in the “Add Ingredients” UI).
* GET /add-ingredients → Display add-ingredients interface (legacy route).

4. Recipes

* GET /recipes/ingredient\_lists/intermediate?ingredient\_list\_id=<id> → Redirect intermediary for generating recipe search URLs.
* GET /recipes/ingredient\_lists/:ingredient\_list\_id → Search for recipes using ingredients from the specified ingredient list.

5. Saved Recipes

* GET /saved\_recipes → Show all saved recipes for the current user.
* POST /saved\_recipes → Save a new recipe.
* DELETE /saved\_recipes/:id → Delete a saved recipe.

6. System & Health

* GET /up → Health check endpoint (returns 200 if app is running).
* GET / → Redirects to /dashboard.