Project Plan: Generative AI - CookCompass

Group: *20*
Members: *Trayan Tsonev (12127140)*, *Dominik Vallazza (11713121)*,
Elias Hirsch (12341086), *Juraj Simkovic (11845263)*

November 19, 2025

Abstract

Building a Generative AI application that lets users find and customize recipes based on available ingredients and dietary needs represents a practical solution to the everyday challenges of meal planning. This project plan outlines a comprehensive approach to developing an interactive chatbot-based recipe recommendation system using Retrieval-Augmented Generation (RAG), leveraging the Food.com dataset to provide personalized, constraint-aware recipe suggestions through natural language queries.

1 Users (5.1.1)

Target Group

While CookCompass can be used by anyone cooking at home, we have identified 2 main target groups that would especially benefit from our system.

- Students often face tight budgets, narrow kitchen space and limited time, all of which limit the ingredients and techniques they can realistically use when cooking. While "quick and easy" student-focused recipes do exist, they tend to be quite repetitive and/or take longer to find. With CookCompass we aim to make finding and customizing suitable recipes quick and seamless.
- People with dietary restrictions (e.g. those with allergies, certain medical conditions, vegans, vegetarians, those from certain religious backgrounds, etc.) often have a hard time finding recipes beyond a handful of familiar options (e.g. gluten-free banana bread). Additionally, factoring in non-dietary considerations, like available ingredients, can further narrow their choices. CookCompass will let them leverage both existing datasets and an LLM to find and/or derive the perfect recipe.

User Goals

From the user's perspective, success means quickly finding recipes that satisfy dietary constraints/preferences and/or maximize the usage of available ingredients.

Workflows

Typical workflows begin with identifying available ingredients and constraints, followed by a natural language query such as "vegetarian pasta with tomatoes and spinach", "what can I

make with chicken, rice, and broccoli?", "egg-free lava cake", etc.

Users currently rely primarily on web search, recipe sites, cooking-themed YouTube channels and increasingly general purpose LLM-chatbots.

By introducing CookCompass, we hope to eliminate:

- the need to browse multiple results until finding a suitable recipe
- the guesswork that often comes with trying to customize the recipe based on available ingredients and/or dietary needs and preferences
- the hallucinations produced by LLMs that don't leverage RAG

2 Data (5.1.2)

Sources and Formats

The system uses the Food.com Recipes and Interactions dataset, which contains over 180,000 recipes and more than 700,000 reviews spanning many years. This dataset is accessed via CSV exports suitable for programmatic parsing. Key fields include recipe IDs, titles, ingredients, instructions, cooking time, servings, nutritional data, user reviews, ratings, tags, and timestamps, enabling both retrieval and evaluation. Target dataset size for development is a stratified sample of 10K to 100K recipes to balance diversity and evaluation reliability against project time and compute limits.

Organization Dimensions

- Granularity: Each recipe is a self-contained document; ingredients are atomic items and instructions are ordered steps.
- Connections: Similarities are mostly implicit and inferred from ingredients, techniques, and tags; user tags exist but can be inconsistent or incomplete.
- Completeness: Most recipes include full ingredients and instructions, while nutritional fields and some metadata can be sparse or noisy.
- Context: Temporal metadata (submission and review times) and interactions provide popularity and trend signals over time.
- Heterogeneity: Core structure is consistent, but user reviews and tags vary in quality and vocabulary, and the ingredients require unit and synonym normalization.

Data Governance

The Food.com dataset is publicly accessible for research; no personal user PII is processed beyond pseudonymous interactions contained in the source. Any additional logs collected during testing will be anonymized and stored without direct identifiers, and only aggregate metrics will be shared for evaluation.

Data preparation includes cleaning, de-duplication, tokenization, chunking and vectorization.

3 The Problem (5.1.3)

In recent years, searching the web for recipes has become somewhat impractical and frustrating for many users, as search engines favor SEO-optimized results that often start with long, irrelevant descriptions.

Real-world example:

- Query: "no bake cheesecake"
- Top result: "These are very serious questions you might consider before making it, especially the goo one. But after making real no-bake cheesecake and tasting the final results, I'm here to declare that no-bake cheesecake should be everyone's dessert obsession. My husband, who "doesn't like cheesecake," went back for seconds. This stuff is GOOD."

In addition, this approach offers limited accommodation for ingredient availability and dietary preferences which often forces users into uncertainty and improvisation. General purpose LLM-chatbots address this to some extent but they often propose "recipes" that make no sense from a culinary standpoint or are simply impractical.

Real-world example:

- Prompt: "Suggest a recipe for butter chicken but replace the chicken with a different protein".
- Response: "Sure! Here's a simple Tuna Butter Chicken style recipe using canned tuna instead of chicken: (...)"

Those difficulties become even more pronounced for multi-contraint queries.

4 The Solution (5.1.4)

Concept

CookCompass helps users discover recipes that match their available ingredients, dietary restrictions, and time constraints. At a conceptual level, the system performs four core tasks:

- Understand: Interpret natural language queries by identifying ingredients, diets (e.g., vegan, gluten-free), time limits and preferences.
- Retrieve: Find recipes from the dataset that meet these restrictions by using both exact ingredient matches and broader semantic similarity.
- Connect: Highlight connections between recipes, such as similar ingredients, preparation methods, or suitable substitutes.
- Generate: Highlight connections between recipes, such as similar ingredients, preparation methods, or suitable substitutes.

User Interface

We chose a Streamlit web application because it is easy to use, works directly in the browser, and is suitable for both quick searches on a laptop and practical use while cooking.

This interface is well-suited to our target group because it works immediately in any browser, requires no installation, and adapts easily to both desktop and mobile use, making it ideal for students and home cooks who often search for recipes while cooking or shopping.

In this user interface, users can enter search queries in natural language, adjust restrictions such as available ingredients, dietary requirements, and preparation time, and instantly discover recommended recipes, which are displayed as clear cards with explanations of why each option suits their needs.

Technical Approach

Challenge 1: Choosing suitable models. We will attempt to find an embedding model and LLM that balance accuracy/output quality on the one hand and computational cost/latency on the other.

Challenge 2: Choosing a retrieval strategy. We will explore different retrieval strategies (mainly Sparse Retrieval, Dense Retrieval and Hybrid Retrieval) across different components of the dataset (e.g. recipe names, ingredient lists, instructions, etc.) and with a different number of items to be retrieved in order to find a good balance between the relevancy and novelty/diversity of the retrieved results. If necessary, we will also look into Multi-Query/Self-Query approaches.

Challenge 3: Constraint enforcement. Due to the high importance of query constraints for our use cases (e.g. dietary preferences, ingredient availability, allergies, etc.) we will test out different ways of enforcing them. We plan to focus primarily on pre- and post-retrieval filtering, along with suitable prompt engineering that allows the LLM to compensate for any deficiencies in the retrieved content.

5 Evaluation (5.1.7)

Success Definition

We will consider 3 main criteria:

- A: The system's ability to suggest relevant recipes that conform to the query constraints
- B: Reasonable latency
- C: The system's ability to respond to secondary queries (e.g. explanation, recipe modification, etc.)

Metrics and Protocol

Criterion A will be evaluated using Mean Average Precision (mAP). We chose this metric as it takes into account both the relevance and the rank of the suggestions. For example, output "irrelevant-irrelevant-relevant" scores lower than "relevant-irrelevant-irrelevant", but "relevant-relevant-irrelevant" is preferable to both. Other metrics such as precision/recall (which only cares about relevance) or mean reciprocal rank (which only takes the rank of the first relevant result into account) were deemed unsuitable.

We will construct a dataset of queries that capture a diverse sort of constraints (e.g. allergies, available ingredients, national cuisine, etc.) and we will calculate the map scores (potentially

across different constraint categories). Due to the stochastic nature of LLM output despite the RAG architecture, the relevance of each output item will be evaluated manually. Our target will be a score of > 0.75.

Criteria B and C will be evaluated in a small user study.